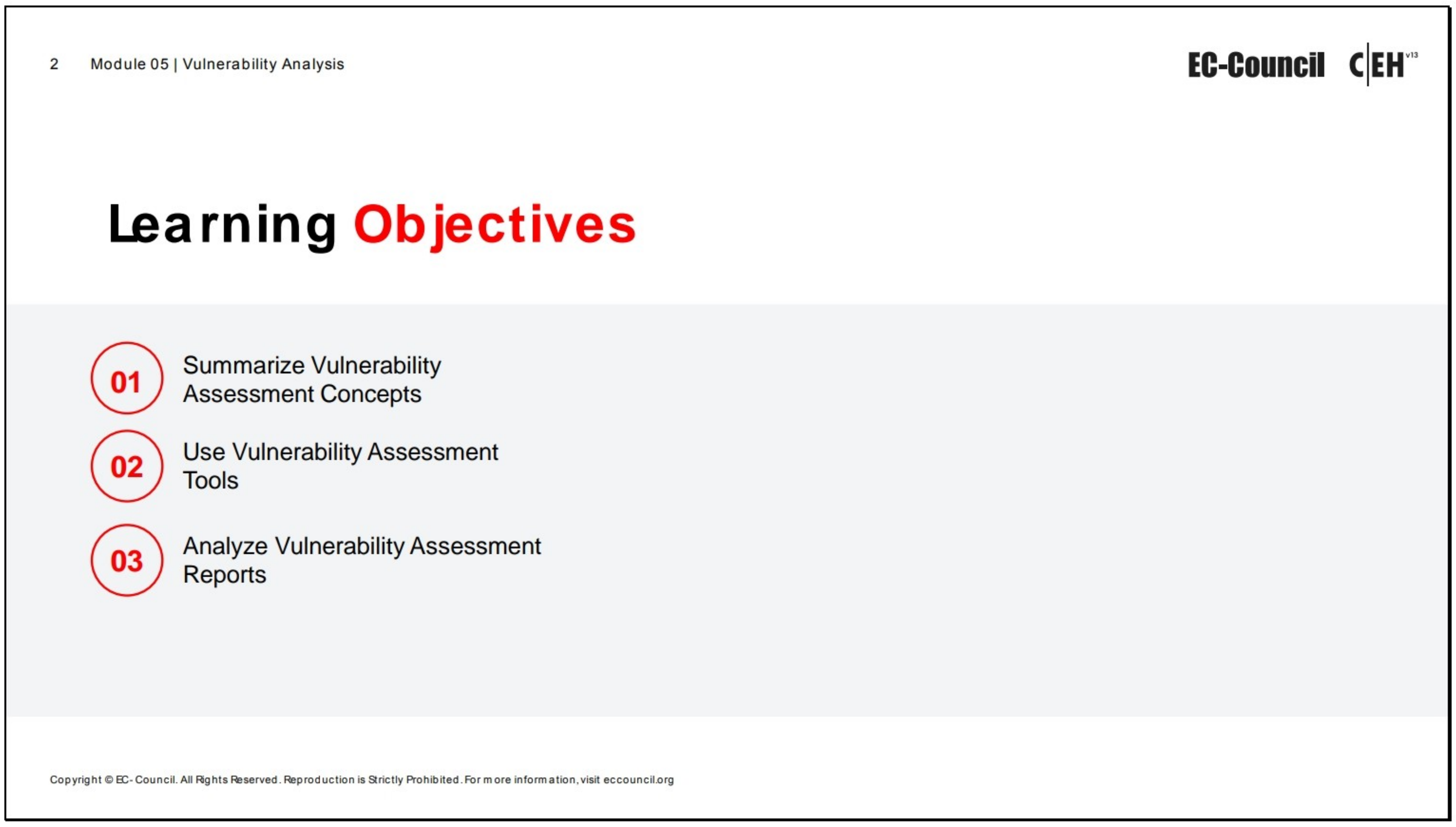
Module 05
Vulnerability Analysis

EC-Council
Official Curricula

EC-Council C|EH v13

Certified Ethical Hacker



EC-Council C|EH v13

Learning Objectives

01 Summarize Vulnerability Assessment Concepts

02 Use Vulnerability Assessment Tools

03 Analyze Vulnerability Assessment Reports

Learning Objectives

In today's world, organizations depend heavily on information technology for protecting vital information. This information is associated with areas of finance, research and development, personnel, legality, and security. Vulnerability assessments scan networks for known security weaknesses.

Attackers perform vulnerability analysis to identify security loopholes in the target organization's network, communication infrastructure, and end systems. The identified vulnerabilities are used by attackers to further exploit that target network.

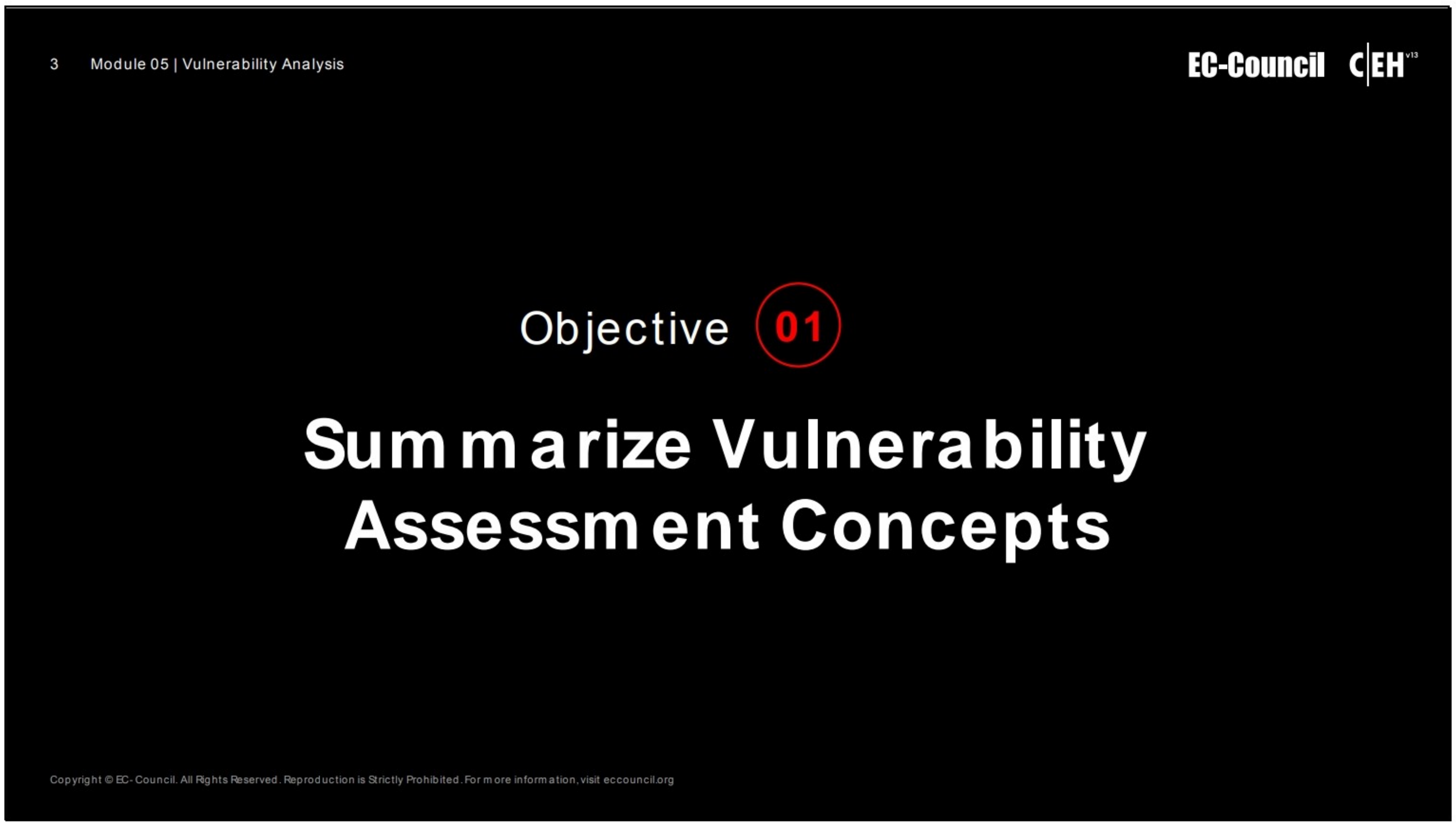
Vulnerability assessment plays a major role in providing security to any organization's resources and infrastructure from various internal and external threats. To secure a network, an administrator needs to perform patch management, install proper antivirus software, check configurations, solve known issues in third-party applications, and troubleshoot hardware with default configurations. All these activities together constitute a vulnerability assessment.

This module starts with an introduction to vulnerability assessment concepts. It also discusses the various vulnerability scoring systems, vulnerability databases, vulnerability management life cycle, and various approaches and tools used to perform vulnerability assessments. This module will provide knowledge about the tools and techniques used by attackers to perform a quality vulnerability analysis. It concludes with an analysis of the vulnerability assessment reports that help an ethical hacker to fix the identified vulnerabilities.

At the end of this module, you will be able to:

- Understand vulnerability classification and vulnerability scoring systems
- Describe the vulnerability-management lifecycle

- Understand vulnerability research, vulnerability scanning, and vulnerability analysis
- Understand various types of vulnerability scanning techniques
- Use various vulnerability assessment tools
- Generate and analyze vulnerability assessment reports


3 Module 05 | Vulnerability Analysis

EC-Council C|EH v13

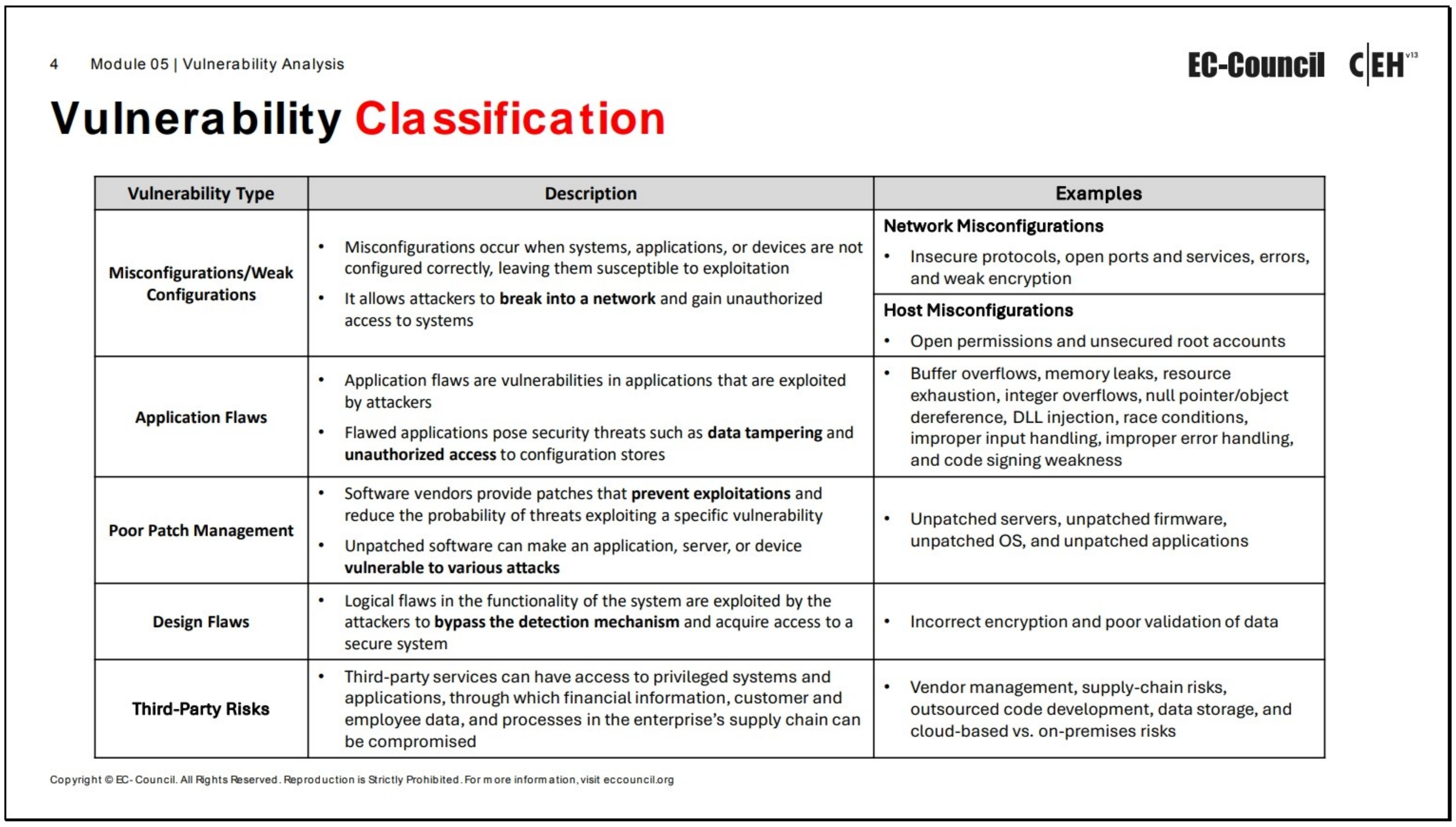
Objective 01

Summarize Vulnerability Assessment Concepts

Copyright © EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited. For more information, visit eccouncil.org

Vulnerability Assessment Concepts

Any vulnerability that is present in a system can be hazardous and can cause severe damage to the organization. It is important for ethical hackers to have knowledge about various types of vulnerabilities that they can employ, along with various vulnerability scanning techniques. This section provides an overview of vulnerability classification, vulnerability scoring systems, vulnerability databases, the vulnerability-management lifecycle, and types of vulnerability scanning.



4 Module 05 | Vulnerability Analysis

EC-Council C|EH v13

Vulnerability Classification

Vulnerability Type	Description	Examples
Misconfigurations/Weak Configurations	• Misconfigurations occur when systems, applications, or devices are not configured correctly, leaving them susceptible to exploitation • It allows attackers to **break into a network** and gain unauthorized access to systems	**Network Misconfigurations** • Insecure protocols, open ports and services, errors, and weak encryption **Host Misconfigurations** • Open permissions and unsecured root accounts
Application Flaws	• Application flaws are vulnerabilities in applications that are exploited by attackers • Flawed applications pose security threats such as **data tampering** and **unauthorized access** to configuration stores	• Buffer overflows, memory leaks, resource exhaustion, integer overflows, null pointer/object dereference, DLL injection, race conditions, improper input handling, improper error handling, and code signing weakness
Poor Patch Management	• Software vendors provide patches that **prevent exploitations** and reduce the probability of threats exploiting a specific vulnerability • Unpatched software can make an application, server, or device **vulnerable to various attacks**	• Unpatched servers, unpatched firmware, unpatched OS, and unpatched applications
Design Flaws	• Logical flaws in the functionality of the system are exploited by the attackers to **bypass the detection mechanism** and acquire access to a secure system	• Incorrect encryption and poor validation of data
Third-Party Risks	• Third-party services can have access to privileged systems and applications, through which financial information, customer and employee data, and processes in the enterprise's supply chain can be compromised	• Vendor management, supply-chain risks, outsourced code development, data storage, and cloud-based vs. on-premises risks

Vulnerability Classification

Vulnerabilities present in a system or network are classified into the following categories:

Misconfigurations/Weak Configurations

Misconfigurations or weak configurations are a common type of vulnerability that can lead to significant security risks. These occur when systems, applications, or devices are not configured correctly, leaving them susceptible to exploitation. It allows attackers to break into a network and gain unauthorized access to systems. Misconfigurations may occur both intentionally and unintentionally, and they affect web servers, application platforms, databases, and networks. Attackers can detect misconfigurations through various scanning techniques and then exploit backend systems. Therefore, administrators must change the default configuration of devices and optimize device security.

- **Network Misconfigurations**

  Frequent changes to network and security devices are inevitable and essential for business improvement. However, administrators should ensure that all network components are configured appropriately because any loops in the implemented changes can cause adverse effects on the network such as performance degradation, service outage, and network intrusions. The following are some examples of weak network configurations.

  - **Insecure Protocols**

    Insecure protocols transmit information or data in plaintext without implementing any encryption techniques to secure the data. The use of vulnerable protocols causes authentication and integrity issues because attackers can leverage the unencrypted files or data transmission and tamper with the data in transit. Attackers can also gain remote access to the vulnerable system once they capture the credentials being shared in plaintext. This vulnerability can be avoided by removing devices operating on insecure protocols and deploying a centralized master node to update protocols.

- **Open Ports and Services**

  User communications with an application or service can be achieved through TCP or UDP port numbers, which accept and transmit the information in the form of packets. The source and destination addresses can be identified through the unique IP addresses assigned to them. In addition to these, many ports operate in a network for specific services. Servers often operate with some open ports, but all open ports are not dangerous, unless they are misconfigured, unpatched, or implemented with poor security rules. However, the open ports must be limited and used only for important services. Leaving ports open for unnecessary services can invite new threats to the network. Open ports and services may lead to the loss of data or Denial-of-Service (DoS) attacks and allow attackers to perform further attacks on other connected devices. Administrators must continuously check for unnecessary or insecure ports and services to reduce the risk to the network.

- **Errors**

  Improper configuration of applications or services can generate error reports while loading pages. Such error reports can provide detailed information to attackers searching for security flaws, application vulnerabilities, programming faults, or other exploits. Using outdated software can also generate security errors, which can be susceptible to remote attacks using techniques such as code injection to manipulate the application. To prevent this vulnerability, skilled programming practices need to be adopted in such a manner that the application does not disclose critical information that could help attackers exploit the application server.

- **Weak Encryption**

  Implementing proper encryption methods can secure the data being transmitted across a network and the data saved on storage devices. The encrypted files can be accessed only with the corresponding decrypted key held by the client or application. Weak encryption can allow attackers to perform man-in-the-middle attacks, sniff the traffic to modify data, and then masquerade as the legitimate service to communicate with the end users with false information. The following are some causes of weak encryption:

  - Using a weak encryption algorithm
  - Key generation with guessable credentials
  - Insecure key distribution

- **Host Misconfigurations**

  Attackers can exploit configuration flaws in the host server to manipulate the resources and gain remote administrator access. The debugging functions could be activated, and unknown users may gain administrative permissions. These vulnerabilities may allow attackers to evade authentication mechanisms and access critical information, possibly with elevated privileges. The following are some examples of weak host configuration.

  - **Open Permissions**

    Granting unnecessary permissions to a user or group of users to access applications or files can lead to security issues such as data leakage or corruption of system functionality. Managing permissions is a complicated task, where administrators or users can potentially make mistakes such as allowing unknown guests to read and write critical files. An attacker can also perform privilege escalation by using unnecessarily created accounts to access unprotected files or to run commands on the operating system (OS).

  - **Unsecured Root Accounts**

    Using manufacturer-allotted default administrative account credentials for the database or applications can lead to system security issues. Failing to implement a secure password privacy policy can allow attackers to guess the credentials using different brute-force techniques.

Application Flaws

Application flaws are vulnerabilities in applications that are exploited by attackers. Applications should be secured using the validation and authorization of the user. Flawed applications pose security threats such as data tampering and unauthorized access to configuration stores. If applications are not secured, sensitive information may be lost or corrupted. Hence, developers must understand the anatomy of common security vulnerabilities and develop highly secure applications by providing proper user validation and authorization.

The following are some of the application flaws that can be exploited by attackers.

- **Buffer Overflows**

  Buffer overflows are common software vulnerabilities resulting from coding errors that allow attackers to gain access to the target system. In a buffer overflow attack, the attacker undermines the functioning of programs and attempts to take control of the system by writing content beyond the allocated size of the buffer. Insufficient bounds checking in the program is the root cause of this vulnerability. The buffer cannot handle data beyond its limit, causing the flow of data to adjacent memory locations and overwriting their data values. When a buffer overflow occurs, systems often crash, become unstable, or show erratic program behavior.

- **Memory Leaks**

  A memory leak or resource leak is an unintended class of memory consumption that occurs when a programmer fails to erase an assigned block of memory when no longer required. It is caused by exceptional circumstances, flaw conditions, and uncertainty over which portion of code is responsible for freeing memory. These conditions depend on application consequences in cases such as such as short-lived user-land applications, long-lived user-land applications, and kernel-land processes. A memory leak results in software reliability–related concerns and encourages a malicious actor to take control over the compromised system to perform attacks such as DoS to crash the system, inject malicious code to change application behavior, and hijack the program's control flow. Tools such as Valgrind, which is compatible with the Unix/Linux environment, track memory leaks and display the status of the software environment.

- **Resource Exhaustion**

  A resource exhaustion attack damages the server by sending multiple resource requests from different locations to exploit software bugs or errors, thereby hanging the system and server or causing a system crash. In software applications, memory management has an error of memory leaks that can be exploited easily by remote attackers. It is similar to a DoS attack in that it can compromise or exhaust the resources available for a system in the network. Owing to design or code errors, any interaction or connection established between the client and server can waste resources or consume more resources than required.

- **Integer Overflows**

  An integer overflow occurs when an arithmetic function generates and attempts to store an integer value larger than the maximum value that the allocated memory space can store. These overflow conditions may lead to undesirable behavior of the software. Failure to discover an overflow condition beforehand can cause security and reliability issues in the program. Alongside yielding inaccurate results and causing software instability, integer overflows can also lead to buffer overflows and open doors for attackers to manipulate values, eventually leading to random or malicious code execution.

- **Null Pointer/Object Dereference**

  Also known as a null reference, a null pointer is a value stored to represent that the pointer is not designated to any valid object; it also indicates invalid memory location. The majority of null-pointer issues lead to common software reliability issues, but once an attacker deliberately triggers a null-pointer dereference, they might be able to use the resulting exception to evade the security logic and make the application disclose debugging details that can help in devising strategies for subsequent attacks. Programs generally utilize these null pointers to indicate a condition such as the last point of unspecified length and incompetence to perform some operations; this type of null-pointer usage is comparable to the nullable types and no value in the option type. A null-pointer dereference can prevent a program from execution or crash the program and cause it to exit.

- **DLL Injection**

  When an application runs third-party code or untrusted code that loads an assembly or DLL file, an attacker may exploit this vulnerability to inject a malicious DLL into the current running process and execute malicious code. Furthermore, loading DLL files without specifying the complete path of the file location may allow attackers to create a malicious DLL and place it in a location that precedes the path of the legitimate DLL file. Consequently, the application executes the malicious DLL. To prevent such vulnerabilities, programmers must never load untrusted DLLs from user input and must always invoke DLLs by specifying the full path of the file location.

- **Race Conditions**

  A race condition is an undesirable incident that occurs when a software or system program depends on the execution of processes in a sequence and on the timing of the programs. This condition occurs when a system that handles events in a sequential format is coerced to perform multiple operations simultaneously. The condition results in the improper execution of a program or software bugs. A typical race condition occurs when multiple threads depend on a shared resource. Most race conditions impact the security associated with the system. An attacker can perform DoS or privilege escalation attacks by accessing the shared resource of a trusted process.

  - **Time of Check/Time of Use**

    The time of check or time of use (TOC/TOU) is a software error that occurs because of the race condition that occurs after checking the state of particular segment of the system at a specific time and before the time of using the checking results. In simple terms, it is defined as the change in system state from the time of checking for a prediction to the time of acting on the prediction. It is a timing vulnerability that occurs when the system grants access permission to a resource request. For example, when a user wishes to transfer an amount from one account to another, a risk of an attack exists in the middle of the transaction between the TOC and TOU, i.e., from the time of checking whether the required amount is available to the time of transferring that amount.

- **Improper Input Handling**

  Input handling is defined as the verification of application functionalities such as validation, filtering, sanitizing, encryption, and decryption of input data. Failure in verifying the input data results in vulnerabilities. Input validation is mandatory to ensure the integrity of incoming data by checking and comparing the data with the type of expected data. Data originating from both trusted and untrusted sources have the risk of being corrupted by attackers using techniques such as SQL injection, cross-site scripting, and buffer overflow. Implementing both client-side and server-side validation ensures effective data authentication.

- **Improper Error Handling**

  Improper error handling occurs when an attacker exploits the security system by utilizing error information. Most web applications or servers disclose detailed information about errors such as database dumps and stack traces. They can also generate detailed errors that include information about the system condition such as system call failure, timeouts, exceptions, and data availability, which can help an attacker analyze and attack the system. Fail-open is one of the security issues caused by improper error handling. Fail-open is defined as the granting of access after a system has failed or denied access.

- **Code Signing Weakness**

  This type of vulnerability occurs when digitally signed software or code has a certificate that lacks validity, has expired, or originated from an unauthorized authority. Inadequate management of the private keys generated during the code signing process may leave some security flaws within the software set for deployment. Attackers leverage these security flaws to steal private keys and embed malicious code within the software. Using the stolen keys, attackers can falsely authenticate the software as legitimate, leading users to install it. Implementing secure storage for private keys in Hardware Security Modules (HSMs) helps defend against the misuse of such private keys.

Poor Patch Management

A patch is a small piece of software designed to fix problems, security vulnerabilities, and bugs as well as improve the usability or performance of a computer program or its supporting data. Software vendors provide patches that prevent exploitations and reduce the probability of threats exploiting a specific vulnerability. Unpatched software can make an application, server, or device vulnerable to various attacks. The following are some examples of poor patch management.

- **Unpatched Servers**

  Servers are an essential component of the infrastructure of any organization. There have been several cases where organizations ran unpatched and misconfigured servers that compromised the security and integrity of the data in their system. Hackers search for these vulnerabilities in servers and exploit them. These unpatched servers serve as a hub for attackers or an entry point into the network. This can lead to the exposure of private data, financial loss, and discontinuation of operations. Updating software regularly and maintaining systems properly by patching and fixing bugs can help in mitigating the vulnerabilities caused by unpatched servers.

- **Unpatched Firmware**

  Unpatched firmware may lead to vulnerabilities through which an attacker can easily enter a corporate network and steal critical information or damage critical resources. Firmware vulnerabilities allow attackers to inject malicious code, infect legitimate updates, delete data stored on the hard drive, or even control the system hardware from a remote location in some cases. To mitigate such vulnerabilities, security professionals must regularly check and update the firmware.

- **Unpatched OS**

  Attackers use systems having unpatched OSes as the origin of an infection vector to infect other systems or devices connected to the same network. Attackers scan for systems having unpatched OSes and use those systems for spreading malware to other systems connected to the network. If an attacker identifies a vulnerability in an OS kernel file or shared library, they can exploit this vulnerability in an attempt to perform privilege escalation using malware that gains system- or root-level access. Security professionals must enable the auto-update feature to update OSes automatically and regularly.

- **Unpatched Applications**

  Unpatched application vulnerabilities allow attackers to inject and run malicious code by exploiting a known software bug. Generally, no software or applications are flawless. Software vendors frequently release patches to resolve identified vulnerabilities. Unpatched applications pave the way for attackers to exploit and compromise the security of systems and software. Therefore, it is important for organizations to apply vulnerability patches and upgrade applications on a regular basis.

Design Flaws

Vulnerabilities due to design flaws are universal to all operating devices and systems. Design vulnerabilities such as incorrect encryption or the poor validation of data refer to logical flaws in the functionality of the system that attackers exploit to bypass the detection mechanism and acquire access to a secure system.

Third-Party Risks

A third party can become another potential threat to enterprises. Third-party services or products can have access to privileged systems and applications, through which financial information, customer and employee data, and processes in the enterprise's supply chain can be compromised. The third party may be trustworthy, but enterprises usually do not check if they maintain appropriate standards and security measures; eventually, they can become a threat for the enterprise network. Major third-party risks include identity theft, intellectual property theft, data breaches, implantation of file-less malware, and network intrusions. An organization should be aware of third-party risks and run real-time, continuous risk management processes within the environment. The following are different types of risks associated with third-party dependency.

- **Vendor management:** It is the activity of selecting suppliers and assessing the risks of third-party services and products. It includes all the essential programs and processes required for an organization to handle and manage operations and communications with its third-party vendors. Organizations often depend on third-party vendors to save expenses, fend off market rivalry, increase productivity, and gain higher profits with lower effort. However, if the third-party vendor is not trusted or fails to follow the required standards, it can pose risks to the organization's data or information. The organization may need to face all the consequences in case of a breach. The best approach to discover risks associated with the third party include employing best vendor management practices alongside enforcing third-party vendor risk management systems.

- **System integration**: It is a process of employing third-party services or hiring third-party vendors to run business operations. When a third party hosts the services or performs software development for the company, the system integrators need full access to the systems/application. As the integrators work from inside the company, they can easily evade firewalls and security solutions and install malware or spyware in the network. The integrators can also employ port scanning techniques to obtain data packets directly from the network. Organizations need to oversee the operations of third-party vendors and the progress of projects.

- **Lack of vendor support**: Organizations often depend on third-party vendors to manage the security of systems inside a network. In such cases, the vendors are entrusted with discovering and fixing issues before they get exploited, and they become members within the working environment of the organization. As they deal with complex network infrastructure, insufficient knowledge in handling security systems or identifying risks can open avenues for new cyber-attacks. Vendors should be adept in finding issues and should be encouraged to maintain a high quality of work and keep systems secure and updated.

- **Supply-chain risks**: The majority of network devices and systems in an organization are often purchased from a third party. The use of such equipment in each segment along the supply chain can potentially pose security risks due to improper maintenance or configuration. Proper security controls must be implemented for the equipment/devices or software that organizations purchase or borrow from a third party. For instance, the software or hardware purchased from a third party may not be properly sanitized. In such cases, malware concealed inside the previously provisioned equipment can infect the new systems deployed in the organization and spread to all other devices connected to the network.

- **Outsourced code development**: In some cases, enterprises do not have all the resources required for developing products inside their environment. In such cases, organizations hire contract-based third parties to develop products or software. In such cases, organizations should establish a secure environment for the third-party designers to develop and assess the code being built. Organizations should also determine where the code needs to be stored and place appropriate security controls to the storage space because the code can be stolen to develop similar projects. After the coding process is completed, the product requires thorough testing, and developers should ensure that unauthorized access to the application resources is prevented. It is also important to ensure that resources being accessed by the application are stored in a protected environment and that data are encrypted before being transmitted over the network.

- **Data storage**: With the emergence of cloud technology, organizations are storing large amounts of data in third-party storage spaces, where vendors may also have access to organizations' data. Therefore, the data should be frequently inspected for security concerns to protect sensitive information related to customers, employees, or users. Organizations should insist that appropriate security controls be implemented and integrity be maintained for the data stored in the third-party storage. Data transmission should be performed with encryption and through a secure channel.
- **Cloud-based vs. on-premises risks:** As organizations are migrating their business infrastructure to cloud environments, storage and data exposure issues often arise in third-party storage locations. On the other hand, businesses running in an on-premises environment may also have issues such as weak security configurations, application or software vulnerabilities, and vendor issues that can emerge from network devices such as firewalls, switches, and routers, which are placed within the organization's infrastructure. Proper configuration and encryption are the main solutions for both environments. In the context of cloud security, the cloud provider has the sole responsibility of securing the cloud; however, the client should also be aware of the best practices to use cloud services in a secure manner.

Default Installations/Default Configurations

Default installations are usually user-friendly — especially when the device is being used for the first time when the primary concern is the usability of the device rather than the device's security. In some cases, infected devices may not contain any valuable information, but are connected to networks or systems that have confidential information that would result in a data breach. Failing to change the default settings while deploying the software or hardware allows the attacker to guess the settings to break into the system.

Systems or devices with default configurations, if connected to the production or corporate network, enable attackers to perform advanced persistent attacks. These systems allow attackers to gain information about the target OS and other vulnerabilities existing in the target network. Based on the identified vulnerabilities, attackers may perform further attacks. When connecting a system or device to a network, it is important to disable unnecessary components and services associated with the default configuration.

Operating System Flaws

Due to vulnerabilities in the operating systems, applications such as Trojans, worms, and viruses pose threats. These attacks use malicious code, script, or unwanted software, which results in the loss of sensitive information and control of computer operations. Timely patching of the OS, installing minimal software applications, and using applications with firewall capabilities are essential steps that an administrator must take to protect the OS from attacks.

Default Passwords

Manufacturers provide users with default passwords to access the device during its initial set-up, which users must change for future use. When users forget to update the passwords and continue using the default passwords, they make devices and systems vulnerable to various attacks, such as brute force and dictionary attacks. Attackers exploit this vulnerability to obtain access to the system. Passwords should be kept confidential; failing to protect the confidentiality of a password allows the system to be easily compromised.

Zero-Day Vulnerabilities

Zero-day vulnerabilities are unknown vulnerabilities in software/hardware that are exposed but not yet patched. These Vulnerabilities are exploited by the attackers before being acknowledged and patched by the software developers or security analysts. Zero-day vulnerabilities are one of the major cyber-threats that continuously expose the vulnerable systems until they get patched.

Legacy Platform Vulnerabilities

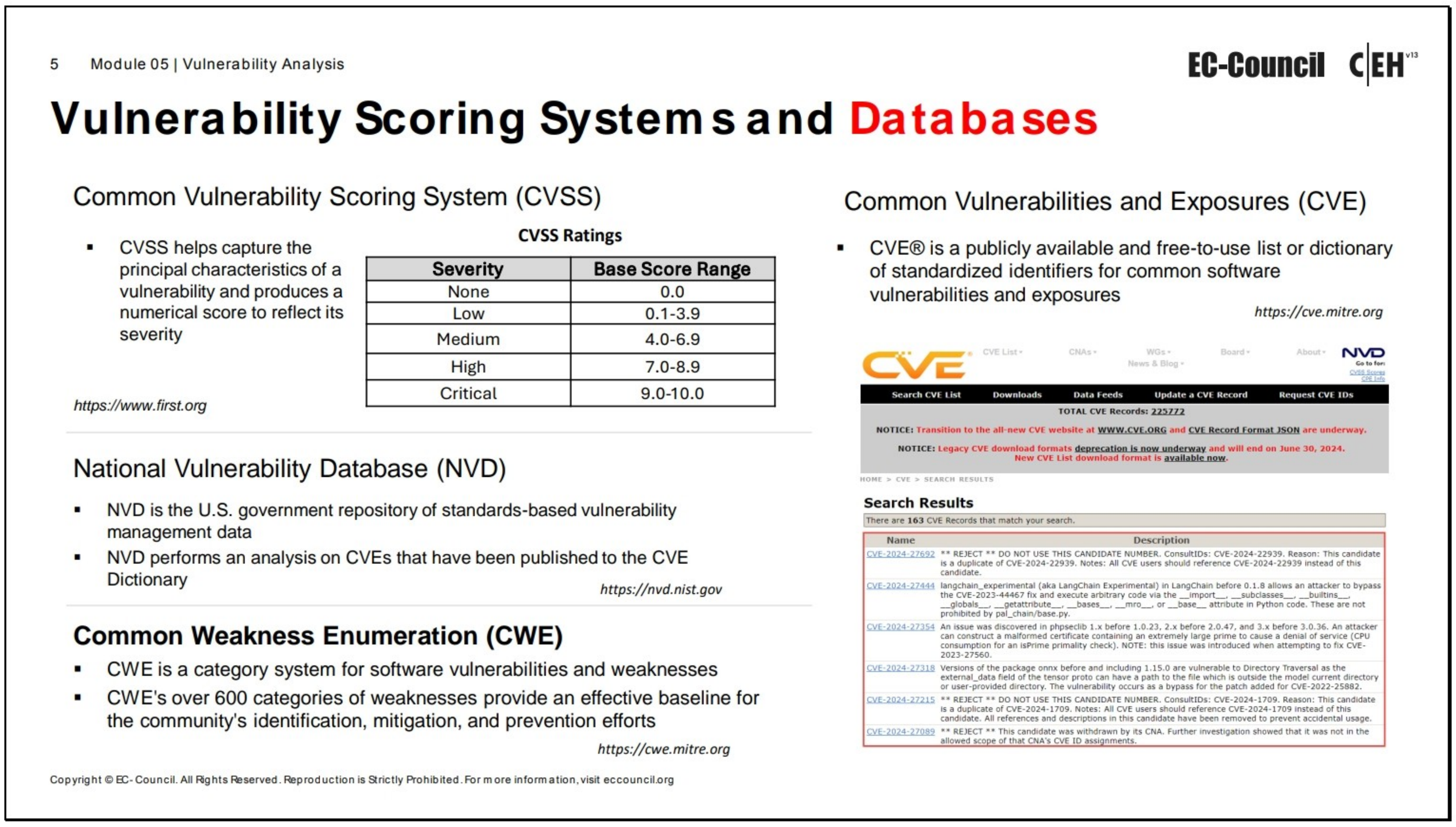
Legacy platform vulnerabilities are caused by obsolete or familiar codes. Legacy platforms are usually not supported when patching technical assets such as smartphones, computers, IoT devices, OSes, applications, databases, firewalls, intrusion detection systems (IDSs), or other network components. This type of vulnerabilities could cause costly data breaches for organizations. Legacy systems can be secured using other security controls, rather than by fixing them. Another possible solution is to segregate these systems from the network so that attackers cannot gain physical access to them.

System Sprawl/Undocumented Assets

The system sprawl vulnerability arises within an organization network because of an increased number of system or server connections without proper documentation or the understanding of their maintenance. These assets are often neglected over time, making them susceptible to attacks. It could also lead to expensive maintenance because each vulnerable asset will be included in the maintenance cost each time effective maintenance is required or the latest hardware or software upgrades need to be scheduled. Additionally, undocumented assets do not support multiplexed database backups or quick multi-streaming, thereby forcing IT teams to choose between fast backups and capacity optimization.

Improper Certificate and Key Management

Improper certificate and key management may lead to many vulnerabilities that allow attackers to perform password cracking and data exfiltration attacks. Keys stored on servers are vulnerable to attacks. Security professionals need to ensure that keys are stored in an encrypted format and are decrypted only in a protected secure environment. Storing or retaining legacy or outdated keys also poses major threats to organizations. Private keys used with certificates must be stored in a highly secured environment; otherwise, an unauthorized individual can intercept the keys and gain access to confidential data or critical systems.



5 Module 05 | Vulnerability Analysis

EC-Council C|EH v13

Vulnerability Scoring Systems and Databases

Common Vulnerability Scoring System (CVSS)

- CVSS helps capture the principal characteristics of a vulnerability and produces a numerical score to reflect its severity

https://www.first.org

CVSS Ratings

Severity	Base Score Range
None	0.0
Low	0.1-3.9
Medium	4.0-6.9
High	7.0-8.9
Critical	9.0-10.0

National Vulnerability Database (NVD)

- NVD is the U.S. government repository of standards-based vulnerability management data
- NVD performs an analysis on CVEs that have been published to the CVE Dictionary

https://nvd.nist.gov

Common Weakness Enumeration (CWE)

- CWE is a category system for software vulnerabilities and weaknesses
- CWE's over 600 categories of weaknesses provide an effective baseline for the community's identification, mitigation, and prevention efforts

https://cwe.mitre.org

Common Vulnerabilities and Exposures (CVE)

- CVE® is a publicly available and free-to-use list or dictionary of standardized identifiers for common software vulnerabilities and exposures

https://cve.mitre.org

CVE List • CNAs • WGs • News & Blog • Board • About • NVD Go to for: CVSS Scores CPE Info

Search CVE List | Downloads | Data Feeds | Update a CVE Record | Request CVE IDs

TOTAL CVE Records: 225772

NOTICE: Transition to the all-new CVE website at WWW.CVE.ORG and CVE Record Format JSON are underway.

NOTICE: Legacy CVE download formats deprecation is now underway and will end on June 30, 2024. New CVE List download format is available now.

HOME > CVE > SEARCH RESULTS

Search Results

There are **163** CVE Records that match your search.

Name	Description
CVE-2024-27692	** REJECT ** DO NOT USE THIS CANDIDATE NUMBER. ConsultIDs: CVE-2024-22939. Reason: This candidate is a duplicate of CVE-2024-22939. Notes: All CVE users should reference CVE-2024-22939 instead of this candidate.
CVE-2024-27444	langchain_experimental (aka LangChain Experimental) in LangChain before 0.1.8 allows an attacker to bypass the CVE-2023-44467 fix and execute arbitrary code via the __import__, __subclasses__, __builtins__, __globals__, __getattribute__, __bases__, __mro__, or __base__ attribute in Python code. These are not prohibited by pal_chain/base.py.
CVE-2024-27354	An issue was discovered in phpseclib 1.x before 1.0.23, 2.x before 2.0.47, and 3.x before 3.0.36. An attacker can construct a malformed certificate containing an extremely large prime to cause a denial of service (CPU consumption for an isPrime primality check). NOTE: this issue was introduced when attempting to fix CVE-2023-27560.
CVE-2024-27318	Versions of the package onnx before and including 1.15.0 are vulnerable to Directory Traversal as the external_data field of the tensor proto can have a path to the file which is outside the model current directory or user-provided directory. The vulnerability occurs as a bypass for the patch added for CVE-2022-25882.
CVE-2024-27215	** REJECT ** DO NOT USE THIS CANDIDATE NUMBER. ConsultIDs: CVE-2024-1709. Reason: This candidate is a duplicate of CVE-2024-1709. Notes: All CVE users should reference CVE-2024-1709 instead of this candidate. All references and descriptions in this candidate have been removed to prevent accidental usage.
CVE-2024-27089	** REJECT ** This candidate was withdrawn by its CNA. Further investigation showed that it was not in the allowed scope of that CNA's CVE ID assignments.

Vulnerability Scoring Systems and Databases

Due to the growing severity of cyber-attacks, vulnerability research has become critical as it helps to mitigate the chance of attacks. Vulnerability research provides awareness of advanced techniques to identify flaws or loopholes in the software that can be exploited by attackers. Vulnerability scoring systems and vulnerability databases are used by security analysts to rank information system vulnerabilities and to provide a composite score of the overall severity and risk associated with identified vulnerabilities. Vulnerability databases collect and maintain information about various vulnerabilities present in information systems.

Following are some of the vulnerability scoring systems and databases:

- Common Vulnerability Scoring System (CVSS)
- Common Vulnerabilities and Exposures (CVE)
- National Vulnerability Database (NVD)
- Common Weakness Enumeration (CWE)

Common Vulnerability Scoring System (CVSS)

Source: *https://www.first.org*

The CVSS is a published standard that provides an open framework for communicating the characteristics and impacts of IT vulnerabilities. The quantitative model of the system ensures repeatable and accurate measurement while enabling users to view the underlying vulnerability characteristics that were used to generate the scores. Thus, the CVSS is well-suited as a standard measurement system for industries, organizations, and governments that need accurate and consistent vulnerability impact scores. Two common uses of CVSS are the prioritization of vulnerability remediation activities and calculation of the severity of vulnerabilities discovered in a system. The FIRST.Org, Inc provides CVSS scores for almost all known vulnerabilities.

The CVSS helps capture the principal characteristics of a vulnerability and produces a numerical score to reflect its severity. This numerical score can thereafter be translated into a qualitative representation (such as low, medium, high, or critical) to help organizations properly assess and prioritize their vulnerability management processes.

CVSS is composed of four metric groups: Base, Threat, Environmental, and Supplemental, each consisting of a set of metrics, as shown in the following table:

- **Base metric**: Represents the intrinsic characteristics of a vulnerability that are constant over time and across user environments.
  - **Exploitability metrics**: Reflects the ease and technical means by which the vulnerability can be exploited.
  - **Impact metrics:** Reflects the direct consequence of a successful exploit.
- **Threat metric**: Reflects the characteristics of a vulnerability related to threat that may change over time but not necessarily across user environments.
- **Environmental metric**: Represents the characteristics of a vulnerability that are relevant and unique to a particular consumer's environment.
- **Supplemental metric**: Includes metrics that provide context as well as describe and measure additional extrinsic attributes of a vulnerability.

Group 1: Base Metric	
Exploitability metrics	**Impact metrics**
Attack Vector	Vulnerable System Confidentiality
Attack Complexity	Vulnerable System Integrity
Attack Requirements	Vulnerable System Availability
Privileges Required	Subsequent System Confidentiality
User Interaction	Subsequent System Integrity
	Subsequent System Availability
Group 2: Threat Metric	
Exploit Maturity	
Group 3: Environmental Metric	
It includes a modified version of exploitability and impact metrics of **Group 1** (base metrics) along with confidentiality, integrity, and availability requirements.	

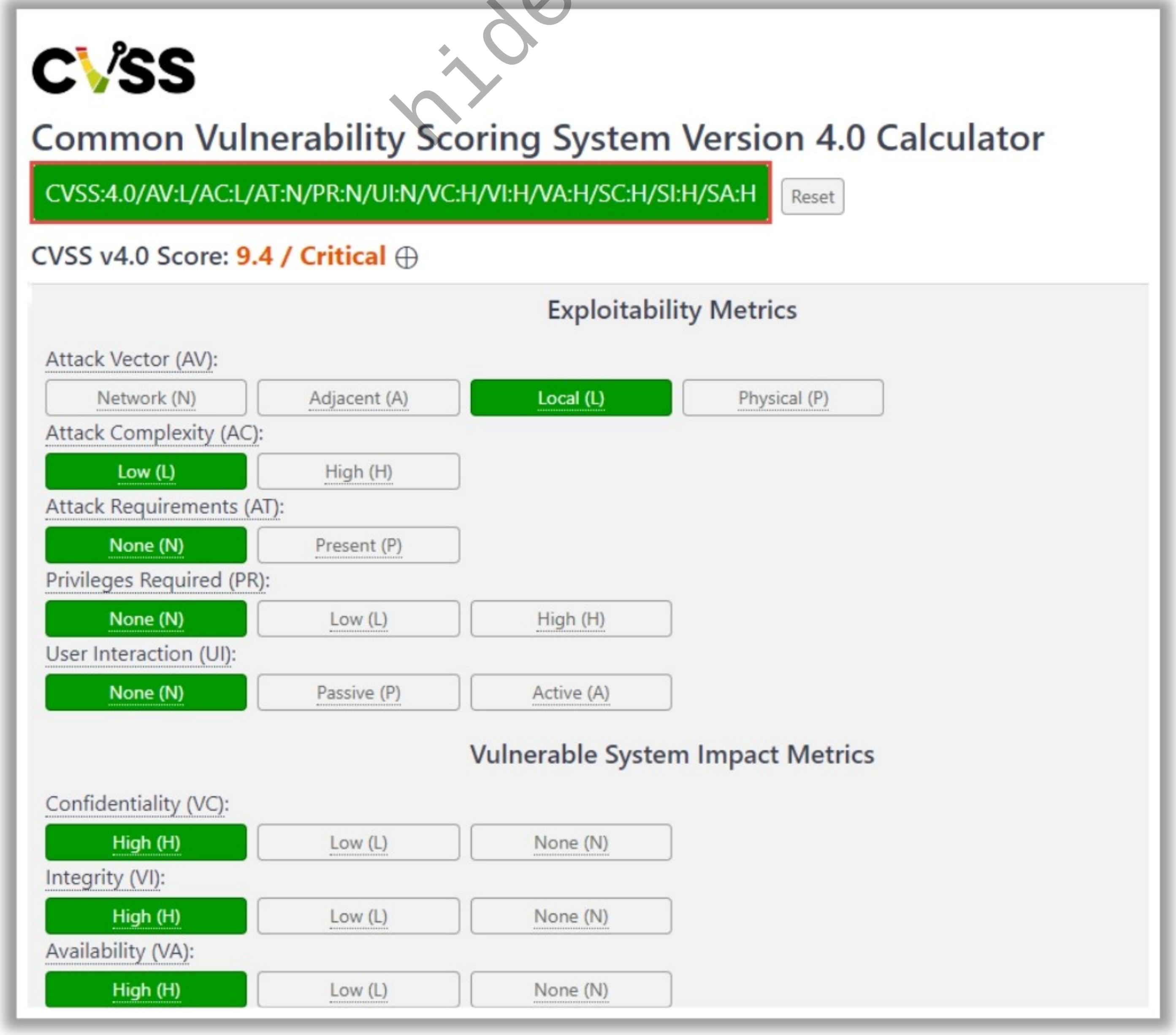
Group 4: Supplemental Metric	
Automatable	Value Density
Recovery	Vulnerability Response Effort
Safety	Provider Urgency

Table 5.1: CVSS 4.0 metric groups

The metric ranges from 1 to 10, with 10 being the most severe. The CVSS score is calculated and generated by a vector string that represents the numerical score for each group in the form of a block of text. The CVSS calculator ranks security vulnerabilities and provides the user with information on the overall severity and risks related to the vulnerability.

Severity	Base Score Range
None	0.0
Low	0.1-3.9
Medium	4.0-6.9
High	7.0-8.9
Critical	9.0-10.0

Table 5.2: CVSS ratings



Common Vulnerability Scoring System Version 4.0 Calculator

CVSS:4.0/AV:L/AC:L/AT:N/PR:N/UI:N/VC:H/VI:H/VA:H/SC:H/SI:H/SA:H

CVSS v4.0 Score: 9.4 / Critical

Exploitability Metrics

Attack Vector (AV): Network (N) Adjacent (A) [✓ Local (L)] Physical (P)

Attack Complexity (AC): [✓ Low (L)] High (H)

Attack Requirements (AT): [✓ None (N)] Present (P)

Privileges Required (PR): [✓ None (N)] Low (L) High (H)

User Interaction (UI): [✓ None (N)] Passive (P) Active (A)

Vulnerable System Impact Metrics

Confidentiality (VC): [✓ High (H)] Low (L) None (N)

Integrity (VI): [✓ High (H)] Low (L) None (N)

Availability (VA): [✓ High (H)] Low (L) None (N)

Figure 5.1: CVSS Calculator

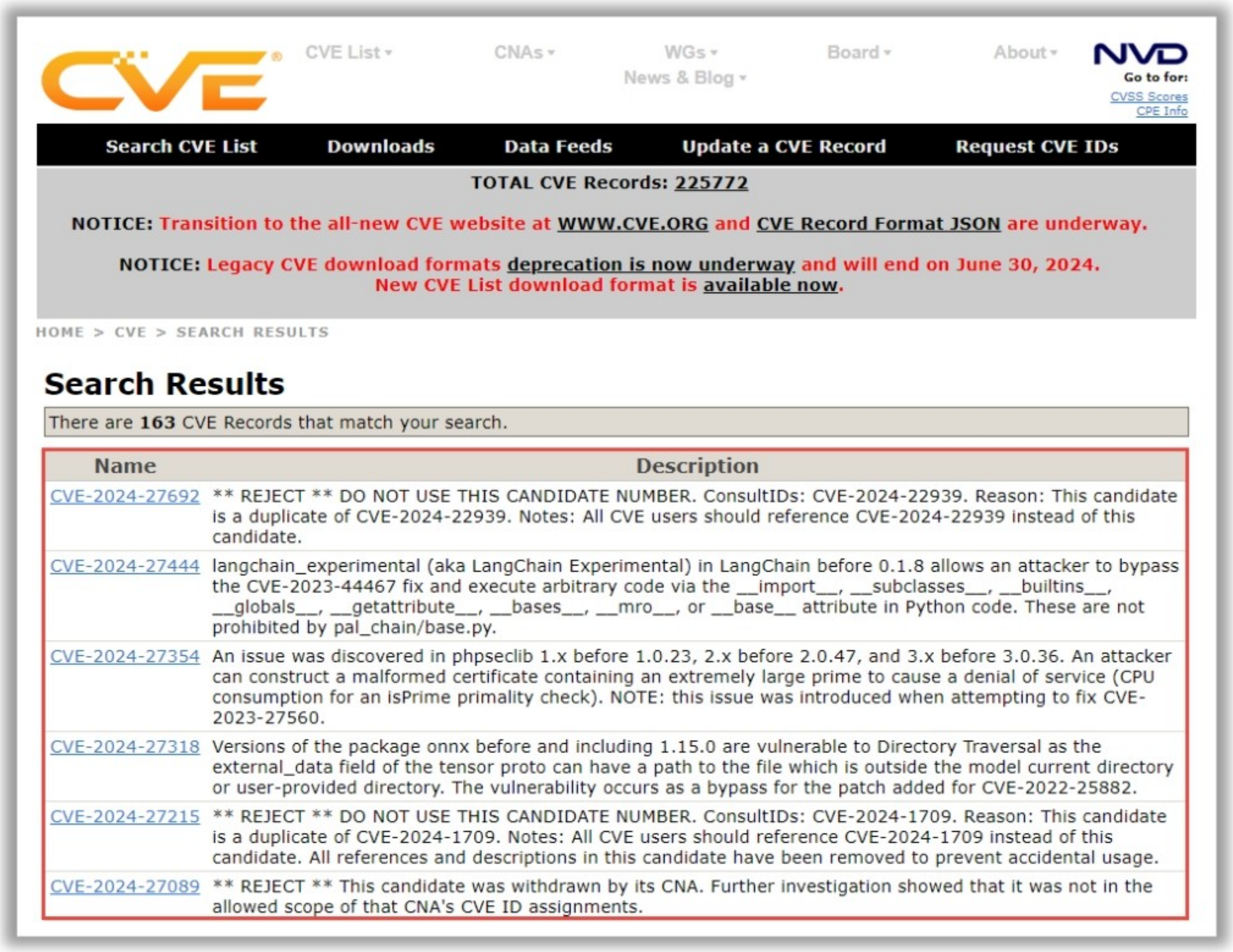
Common Vulnerabilities and Exposures (CVE)

Source: *https://cve.mitre.org*

CVE® is a publicly available and free-to-use list or dictionary of standardized identifiers for common software vulnerabilities and exposures. The use of CVE Identifiers, or "CVE IDs," which are assigned by CVE Numbering Authorities (CNAs) from around the world, ensures confidence among parties when discussing or sharing information about a unique software or firmware vulnerability. CVE provides a baseline for tool evaluation and enables data exchange for cybersecurity automation. CVE IDs provide a baseline for evaluating the coverage of tools and services so that users can determine which tools are most effective and appropriate for their organization's needs. In short, products and services compatible with CVE provide better coverage, easier interoperability, and enhanced security.

What CVE is:

- One identifier for one vulnerability or exposure
- One standardized description for each vulnerability or exposure
- A dictionary rather than a database
- A method for disparate databases and tools to "speak" the same language
- The way to interoperability and better security coverage
- A basis for evaluation among services, tools, and databases
- Free for the public to download and use
- Industry-endorsed via the CVE Numbering Authorities, CVE Board, and the numerous products and services that include CVE



TOTAL CVE Records: 225772

NOTICE: Transition to the all-new CVE website at WWW.CVE.ORG and CVE Record Format JSON are underway.

NOTICE: Legacy CVE download formats deprecation is now underway and will end on June 30, 2024.
New CVE List download format is available now.

HOME > CVE > SEARCH RESULTS

Search Results

There are **163** CVE Records that match your search.

Name	Description
CVE-2024-27692	** REJECT ** DO NOT USE THIS CANDIDATE NUMBER. ConsultIDs: CVE-2024-22939. Reason: This candidate is a duplicate of CVE-2024-22939. Notes: All CVE users should reference CVE-2024-22939 instead of this candidate.
CVE-2024-27444	langchain_experimental (aka LangChain Experimental) in LangChain before 0.1.8 allows an attacker to bypass the CVE-2023-44467 fix and execute arbitrary code via the __import__, __subclasses__, __builtins__, __globals__, __getattribute__, __bases__, __mro__, or __base__ attribute in Python code. These are not prohibited by pal_chain/base.py.
CVE-2024-27354	An issue was discovered in phpseclib 1.x before 1.0.23, 2.x before 2.0.47, and 3.x before 3.0.36. An attacker can construct a malformed certificate containing an extremely large prime to cause a denial of service (CPU consumption for an isPrime primality check). NOTE: this issue was introduced when attempting to fix CVE-2023-27560.
CVE-2024-27318	Versions of the package onnx before and including 1.15.0 are vulnerable to Directory Traversal as the external_data field of the tensor proto can have a path to the file which is outside the model current directory or user-provided directory. The vulnerability occurs as a bypass for the patch added for CVE-2022-25882.
CVE-2024-27215	** REJECT ** DO NOT USE THIS CANDIDATE NUMBER. ConsultIDs: CVE-2024-1709. Reason: This candidate is a duplicate of CVE-2024-1709. Notes: All CVE users should reference CVE-2024-1709 instead of this candidate. All references and descriptions in this candidate have been removed to prevent accidental usage.
CVE-2024-27089	** REJECT ** This candidate was withdrawn by its CNA. Further investigation showed that it was not in the allowed scope of that CNA's CVE ID assignments.

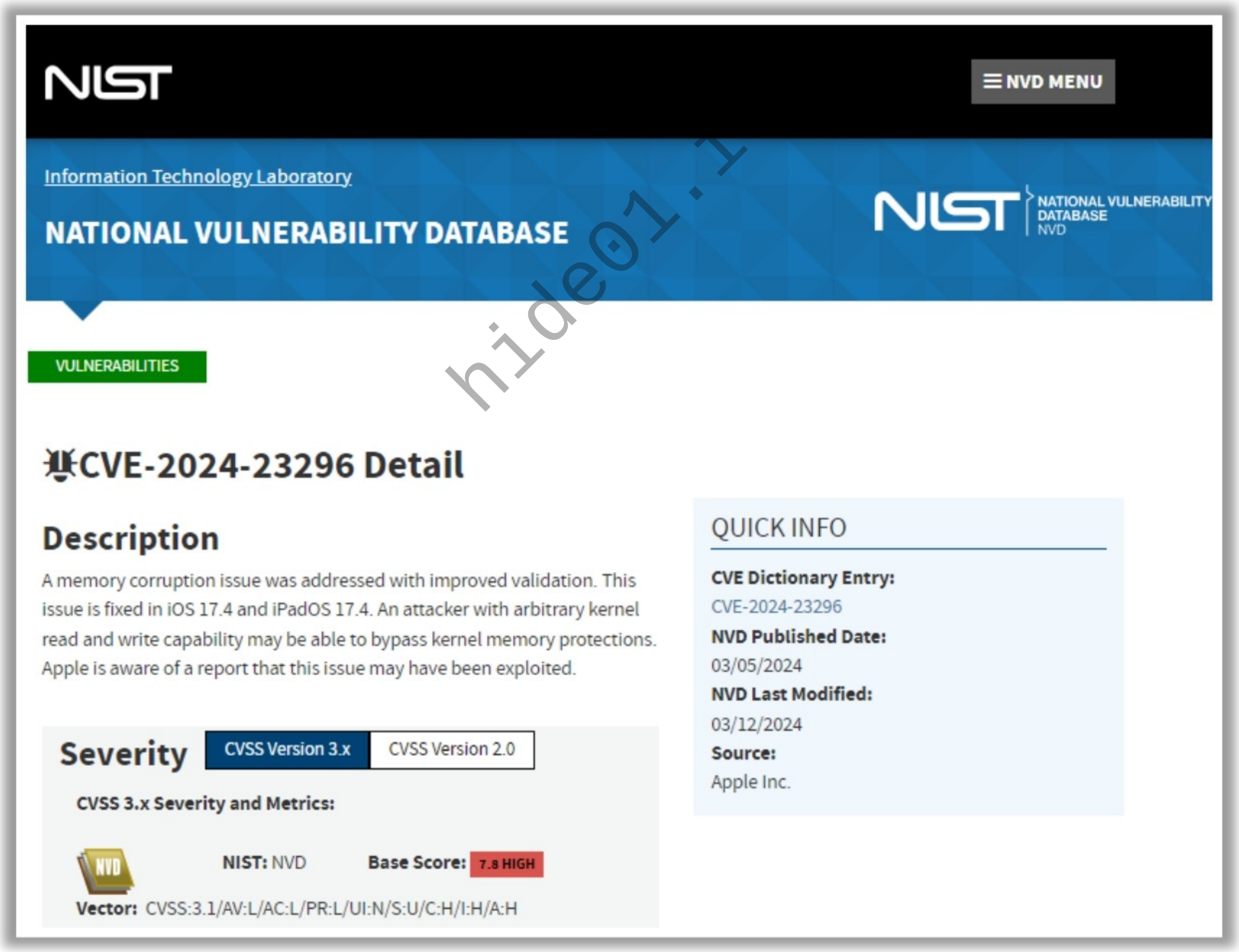
Figure 5.2: Common Vulnerabilities and Exposures (CVE)

National Vulnerability Database (NVD)

Source: *https://nvd.nist.gov*

The NVD is the U.S. government repository of standards-based vulnerability management data. It uses the Security Content Automation Protocol (SCAP). Such data enable the automation of vulnerability management, security measurement, and compliance. The NVD includes databases of security checklist references, security-related software flaws, misconfigurations, product names, and impact metrics.

The NVD performs an analysis on CVEs that have been published to the CVE Dictionary. NVD staff are tasked with the analysis of CVEs by aggregating data points from the description, references supplied, and any supplemental data that are publicly available. This analysis results in association impact metrics (Common Vulnerability Scoring System – CVSS), vulnerability types (Common Weakness Enumeration — CWE), and applicability statements (Common Platform Enumeration — CPE), as well as other pertinent metadata. The NVD does not actively perform vulnerability testing; it relies on vendors, third party security researchers, and vulnerability coordinators to provide information that is used to assign these attributes.


NIST
NVD MENU
Information Technology Laboratory
NATIONAL VULNERABILITY DATABASE
NIST NATIONAL VULNERABILITY DATABASE NVD
VULNERABILITIES

CVE-2024-23296 Detail

Description

A memory corruption issue was addressed with improved validation. This issue is fixed in iOS 17.4 and iPadOS 17.4. An attacker with arbitrary kernel read and write capability may be able to bypass kernel memory protections. Apple is aware of a report that this issue may have been exploited.

Severity CVSS Version 3.x CVSS Version 2.0

CVSS 3.x Severity and Metrics:

NIST: NVD Base Score: 7.8 HIGH

Vector: CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:H

QUICK INFO

CVE Dictionary Entry: CVE-2024-23296

NVD Published Date: 03/05/2024

NVD Last Modified: 03/12/2024

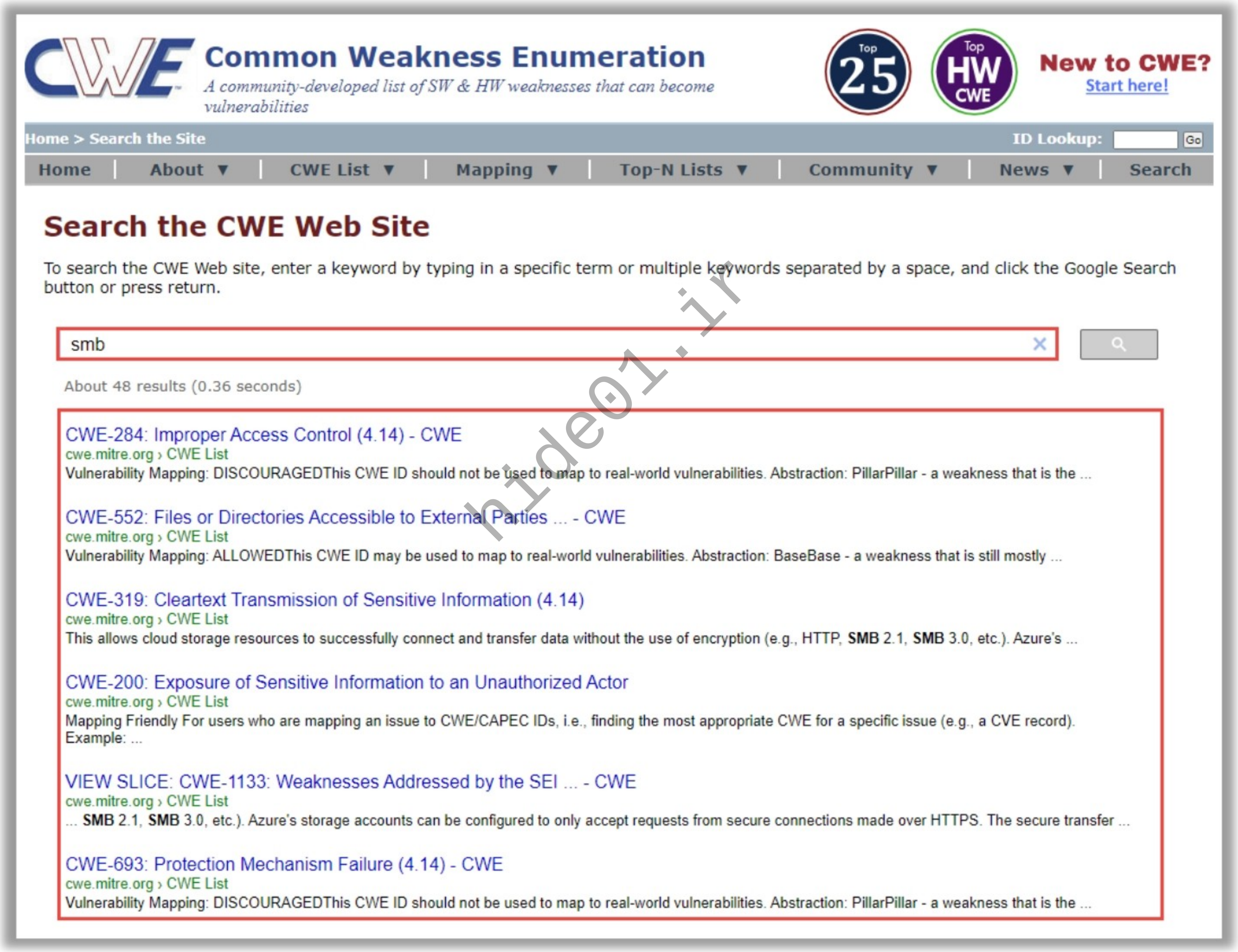
Source: Apple Inc.

Figure 5.3: Screenshot showing CVE details in the National Vulnerability Database (NVD)

Common Weakness Enumeration (CWE)

Source: *https://cwe.mitre.org*

Common Weakness Enumeration (CWE) is a category system for software vulnerabilities and weaknesses. It is sponsored by the National Cybersecurity FFRDC, which is owned by The MITRE Corporation, with support from US-CERT and the National Cyber Security Division of the U.S. Department of Homeland Security. The latest version 3.2 of the CWE standard was released in January 2019. It has over 600 categories of weaknesses, which gives CWE the ability to be effectively employed by the community as a baseline for weakness identification, mitigation, and prevention efforts. It also has an advanced search technique where attackers can search and view weaknesses based on research concepts, development concepts, and architectural concepts.



Search the CWE Web Site

To search the CWE Web site, enter a keyword by typing in a specific term or multiple keywords separated by a space, and click the Google Search button or press return.

smb

About 48 results (0.36 seconds)

CWE-284: Improper Access Control (4.14) - CWE
cwe.mitre.org › CWE List
Vulnerability Mapping: DISCOURAGEDThis CWE ID should not be used to map to real-world vulnerabilities. Abstraction: PillarPillar - a weakness that is the ...

CWE-552: Files or Directories Accessible to External Parties ... - CWE
cwe.mitre.org › CWE List
Vulnerability Mapping: ALLOWEDThis CWE ID may be used to map to real-world vulnerabilities. Abstraction: BaseBase - a weakness that is still mostly ...

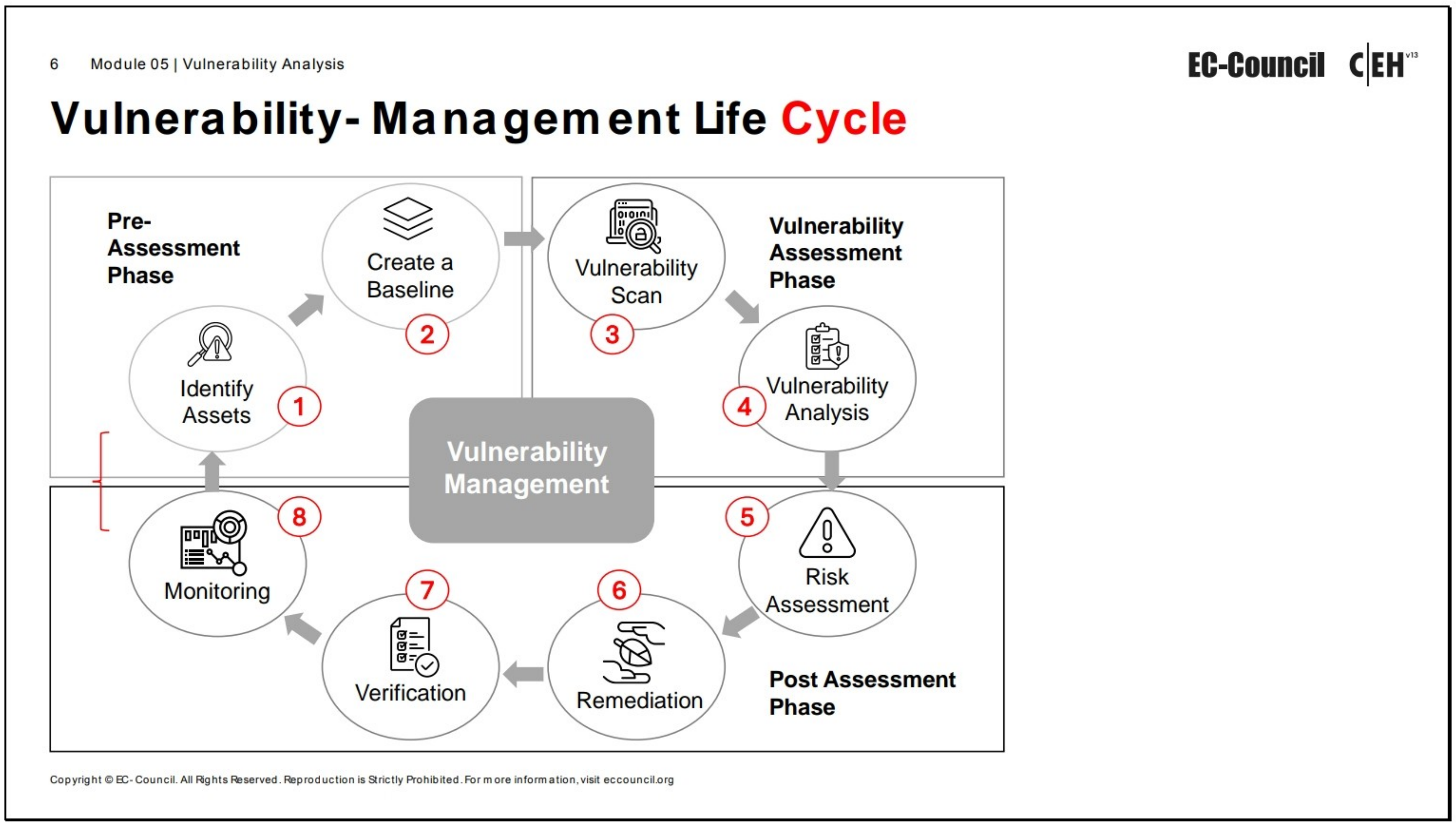
CWE-319: Cleartext Transmission of Sensitive Information (4.14)
cwe.mitre.org › CWE List
This allows cloud storage resources to successfully connect and transfer data without the use of encryption (e.g., HTTP, **SMB** 2.1, **SMB** 3.0, etc.). Azure's ...

CWE-200: Exposure of Sensitive Information to an Unauthorized Actor
cwe.mitre.org › CWE List
Mapping Friendly For users who are mapping an issue to CWE/CAPEC IDs, i.e., finding the most appropriate CWE for a specific issue (e.g., a CVE record). Example: ...

VIEW SLICE: CWE-1133: Weaknesses Addressed by the SEI ... - CWE
cwe.mitre.org › CWE List
... **SMB** 2.1, **SMB** 3.0, etc.). Azure's storage accounts can be configured to only accept requests from secure connections made over HTTPS. The secure transfer ...

CWE-693: Protection Mechanism Failure (4.14) - CWE
cwe.mitre.org › CWE List
Vulnerability Mapping: DISCOURAGEDThis CWE ID should not be used to map to real-world vulnerabilities. Abstraction: PillarPillar - a weakness that is the ...

Figure 5.4: Screenshot showing CWE results for SMB query


6 Module 05 | Vulnerability Analysis
EC-Council C|EH v13
Vulnerability- Management Life Cycle
Pre-Assessment Phase
Vulnerability Assessment Phase
Post Assessment Phase
Vulnerability Management
1 Identify Assets
2 Create a Baseline
3 Vulnerability Scan
4 Vulnerability Analysis
5 Risk Assessment
6 Remediation
7 Verification
8 Monitoring
Copyright © EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited. For more information, visit eccouncil.org

Vulnerability-Management Life Cycle

The vulnerability management life cycle is an important process that helps identify and remediate security weaknesses before they can be exploited. This includes defining the risk posture and policies for an organization, creating a complete asset list of systems, scanning and assessing the environment for vulnerabilities and exposures, and taking action to mitigate the vulnerabilities that are identified. The implementation of a vulnerability management lifecycle helps gain a strategic perspective regarding possible cybersecurity threats and renders insecure computing environments more resilient to attacks.

Vulnerability management should be implemented in every organization as it evaluates and controls the risks and vulnerabilities in the system. The management process continuously examines the IT environments for vulnerabilities and risks associated with the system.

Organizations should maintain a proper vulnerability management program to ensure overall information security. Vulnerability management provides the best results when it is implemented in a sequence of well-organized phases.

The phases involved in vulnerability management are:

- **Pre-Assessment Phase**
  - Identify Assets and Create a Baseline
- **Vulnerability Assessment Phase**
  - Vulnerability Scan
  - Vulnerability Analysis

- **Post Assessment Phase**
  - Risk Assessment
  - Remediation
  - Verification
  - Monitoring

Pre-Assessment Phase

Identify Assets and Create a Baseline

The pre-assessment phase is a preparatory phase, which involves defining policies and standards, clarifying the scope of the assessment, designing appropriate information protection procedures, and identifying and prioritizing critical assets to create a good baseline for vulnerability management and to define the risk based on the criticality and value of each system. This phase involves the gathering of information about the identified systems to understand the approved ports, software, drivers, and basic configuration of each system in order to develop and maintain a system baseline.

The following are the steps involved in creating a baseline:

1. Identify and understand business processes
2. Identify the applications, data, and services that support the business processes and perform code reviews
3. Identify the approved software, drivers, and basic configuration of each system
4. Create an inventory of all assets, and prioritize or rank the critical assets
5. Understand the network architecture and map the network infrastructure
6. Identify the controls already in place
7. Understand policy implementation and practice standard compliance with business processes
8. Define the scope of the assessment
9. Create information protection procedures to support effective planning, scheduling, coordination, and logistics

Classify the identified assets according to the business needs. Classification helps to identify the high business risks in an organization. Prioritize the rated assets based on the impact of their failure and their reliability in the business.

Prioritization helps:

- Evaluate and decide a solution for the consequence of the assets failing
- Examine the risk tolerance level
- Organize methods for prioritizing the assets

Vulnerability Assessment Phase

This phase is very crucial in vulnerability management. The vulnerability assessment phase refers to identifying vulnerabilities in the organization's infrastructure, including the operating system, web applications, and web server. It helps identify the category and criticality of the vulnerability in an organization and minimizes the level of risk. The ultimate goal of vulnerability scanning is to scan, examine, evaluate, and report the vulnerabilities in the organization's information system. Vulnerability scans can also be performed on applicable compliance templates to assess the organization's Infrastructure weaknesses against the respective compliance guidelines.

The assessment phase involves examining the architecture of the network, evaluating threats to the environment, performing penetration testing, examining and evaluating physical security, analyzing physical assets, assessing operational security, observing policies and procedures, and assessing the infrastructure's interdependencies.

Steps involved in the assessment phase:

1. Examine and evaluate the physical security
2. Check for misconfigurations and human errors
3. Run vulnerability scans using tools
4. Select the type of scan based on the organization or compliance requirements
5. Identify and prioritize vulnerabilities
6. Identify false positives and false negatives
7. Apply the business and technology context to scanner results
8. Perform OSINT information gathering to validate the vulnerabilities
9. Create a vulnerability scan report

Post Assessment Phase

The post-assessment phase, also known as the recommendation phase, is performed after and based on risk assessment. Risk characterization is categorized by key criteria, which helps prioritize the list of recommendations.

The tasks performed in the post-assessment phase include:

- Creating a priority list for assessment recommendations based on the impact analysis
- Developing an action plan to implement the proposed remediation
- Capturing lessons learned to improve the complete process in the future
- Conducting training for employees

Post assessment includes risk assessment, remediation, verification, and monitoring.

- **Risk Assessment**

  In the risk assessment phase, risks are identified, characterized, and classified along with the techniques used to control or reduce their impact. It is an important step toward identifying the security weaknesses in the IT architecture of an organization.

  In this phase, all serious uncertainties that are associated with the system are assessed and prioritized, and remediation is planned to permanently eliminate system flaws. The risk assessment summarizes the vulnerability and risk level identified for each of the selected assets. It determines whether the risk level for a particular asset is high, moderate, or low. Remediation is planned based on the determined risk level. For example, vulnerabilities ranked high-risk are targeted first to decrease the chances of exploitation that would adversely impact the organization.

  The tasks performed in the risk assessment phase include:

  - Perform risk categorization based on risk ranking (for example, critical, high, medium, and low)
  - Assess the level of impact
  - Determine the threat and risk levels

- **Remediation**

  Remediation is the process of applying fixes on vulnerable systems in order to mitigate or reduce the impact and severity of vulnerabilities. These include steps like evaluating vulnerabilities, locating risks, and designing responses for vulnerabilities. It is important for the remediation process to be specific, measurable, attainable, relevant, and time-bound.

  This phase is initiated after the successful implementation of the baseline and assessment steps.

  The tasks performed in the remediation phase include:

  - Prioritize remediation based on the risk ranking
  - Develop an action plan to implement the recommendation or remediation
  - Perform a root-cause analysis
  - Apply patches and fixes
  - Capture lessons learned
  - Conduct awareness training
  - Perform exception handling and risk acceptance for the vulnerabilities that cannot be remediated

- **Verification**

  In this phase, the security team performs a re-scan of systems to assess if the required remediation is complete and whether the individual fixes have been applied to the impacted assets. This phase includes the verification of the remedies used to mitigate risks. It provides clear visibility into the firm and allows the security team to check whether all the previous phases have been perfectly employed or not. Verification can be performed by using various means such as ticketing systems, scanners, and reports.

  The tasks performed in the verification phase include:

  - Rescanning the systems to identify if an applied fix is effective in remediating the vulnerability
  - Performing dynamic analysis
  - Reviewing the attack surface

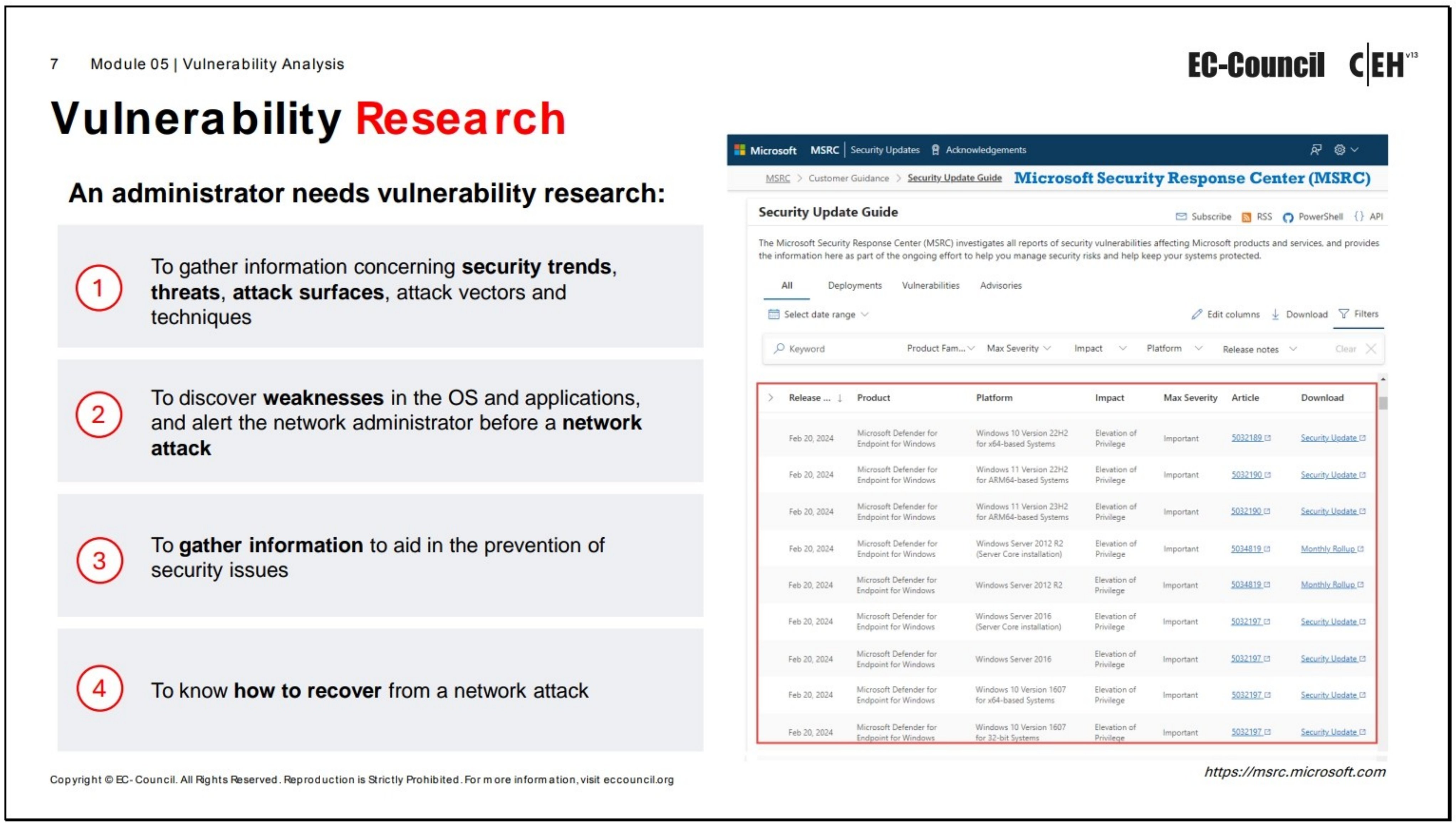
- **Monitoring**

  Organizations need to perform regular monitoring to maintain system security. Continuous monitoring identifies potential threats and any new vulnerabilities that have evolved. As per security best practices, all phases of vulnerability management must be performed regularly.

  This phase performs incident monitoring using tools such as IDS/IPS, SIEM, and firewalls. It implements continuous security monitoring to thwart ever-evolving threats.

  The tasks performed in the monitoring phase include:

  - Periodic vulnerability scan and assessment
  - Timely remediation of identified vulnerabilities
  - Monitoring intrusion detection and intrusion prevention logs
  - Implementing policies, procedures, and controls



7 Module 05 | Vulnerability Analysis

Vulnerability Research

An administrator needs vulnerability research:

1. To gather information concerning **security trends**, **threats**, **attack surfaces**, attack vectors and techniques
2. To discover **weaknesses** in the OS and applications, and alert the network administrator before a **network attack**
3. To **gather information** to aid in the prevention of security issues
4. To know **how to recover** from a network attack

Microsoft Security Response Center (MSRC)

Microsoft MSRC | Security Updates Acknowledgements

MSRC > Customer Guidance > Security Update Guide

Security Update Guide

Subscribe RSS PowerShell {} API

The Microsoft Security Response Center (MSRC) investigates all reports of security vulnerabilities affecting Microsoft products and services, and provides the information here as part of the ongoing effort to help you manage security risks and help keep your systems protected.

All Deployments Vulnerabilities Advisories

Select date range | Edit columns | Download | Filters

Keyword | Product Fam... | Max Severity | Impact | Platform | Release notes | Clear

Release ...	Product	Platform	Impact	Max Severity	Article	Download
Feb 20, 2024	Microsoft Defender for Endpoint for Windows	Windows 10 Version 22H2 for x64-based Systems	Elevation of Privilege	Important	5032189	Security Update
Feb 20, 2024	Microsoft Defender for Endpoint for Windows	Windows 11 Version 22H2 for ARM64-based Systems	Elevation of Privilege	Important	5032190	Security Update
Feb 20, 2024	Microsoft Defender for Endpoint for Windows	Windows 11 Version 23H2 for ARM64-based Systems	Elevation of Privilege	Important	5032190	Security Update
Feb 20, 2024	Microsoft Defender for Endpoint for Windows	Windows Server 2012 R2 (Server Core installation)	Elevation of Privilege	Important	5034819	Monthly Rollup
Feb 20, 2024	Microsoft Defender for Endpoint for Windows	Windows Server 2012 R2	Elevation of Privilege	Important	5034819	Monthly Rollup
Feb 20, 2024	Microsoft Defender for Endpoint for Windows	Windows Server 2016 (Server Core installation)	Elevation of Privilege	Important	5032197	Security Update
Feb 20, 2024	Microsoft Defender for Endpoint for Windows	Windows Server 2016	Elevation of Privilege	Important	5032197	Security Update
Feb 20, 2024	Microsoft Defender for Endpoint for Windows	Windows 10 Version 1607 for x64-based Systems	Elevation of Privilege	Important	5032197	Security Update
Feb 20, 2024	Microsoft Defender for Endpoint for Windows	Windows 10 Version 1607 for 32-bit Systems	Elevation of Privilege	Important	5032197	Security Update

https://msrc.microsoft.com

Vulnerability Research

Vulnerability research involves utilizing various online resources, tools, and platforms to identify, analyze, and share information about security vulnerabilities.

An administrator needs vulnerability research:

- To gather information about security trends, newly discovered threats, attack surfaces, attack vectors and techniques
- To find weaknesses in the OS and applications and alert the network administrator before a network attack
- To understand information that helps prevent security problems
- To know how to recover from a network attack
- To prioritize and apply security patches and updates effectively, mitigating risks before they can be exploited
- To adhere to industry best practices for security, ensuring systems are not just compliant, but also secured according to the highest standards
- To perform accurate risk assessments, identifying and prioritizing the most critical threats to address

An ethical hacker needs to keep up with the most recently discovered vulnerabilities and exploits to stay one step ahead of attackers through vulnerability research, which includes:

- Discovering the system design faults and weaknesses that might allow attackers to compromise a system

- Staying updated about new products and technologies and reading news related to current exploits
- Checking underground hacking web sites (Deep and Dark websites) for newly discovered vulnerabilities and exploits
- Checking newly released alerts regarding relevant innovations and product improvements for security systems
- Anticipating how a system might be attacked and take steps to mitigate those risks
- Helping organizations develop robust defensive strategies that protect against specific threats
- Tailoring security solutions to the unique needs and risk profiles of the organizations
- Conducting thorough audits that identify compliance issues and security gaps

Security experts and vulnerability scanners classify vulnerabilities by:

- Severity level (low, medium, or high)
- Exploit range (local or remote)

Ethical hackers need to conduct intense research with the help of information acquired in the footprinting and scanning phases to find vulnerabilities.

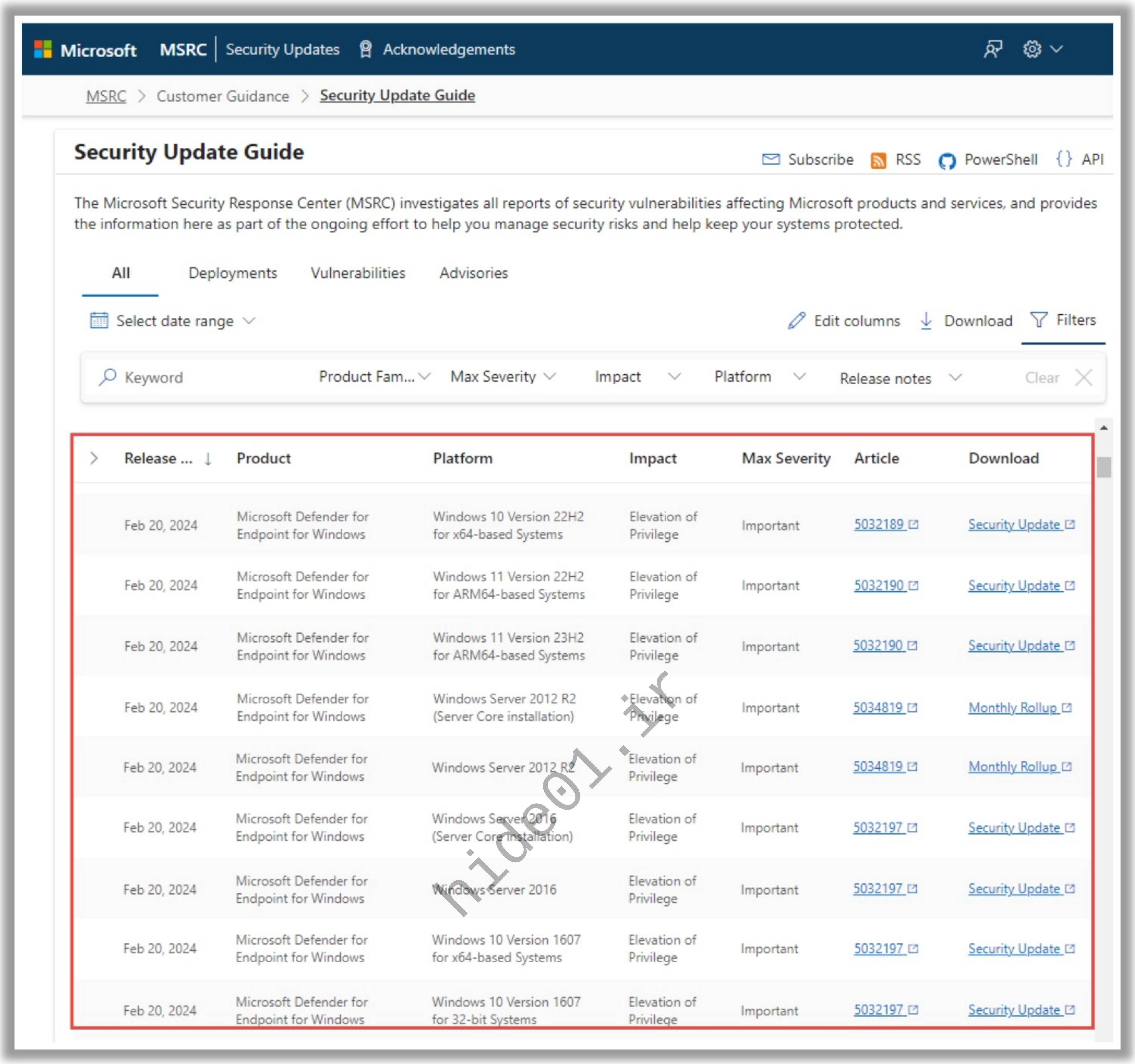
Resources for Vulnerability Research

The following are some of the websites used to perform vulnerability research.

- **Microsoft Security Response Center (MSRC)**

  Source: *https://msrc.microsoft.com*

  The Microsoft Security Response Center (MSRC) investigates all reports of security vulnerabilities affecting Microsoft products and services, and it provides information as part of an ongoing effort to help security professionals manage security risks and keep organizational systems protected.


Microsoft MSRC | Security Updates Acknowledgements
MSRC > Customer Guidance > Security Update Guide
Subscribe RSS PowerShell API
All Deployments Vulnerabilities Advisories
Select date range
Edit columns Download Filters
Keyword Product Fam... Max Severity Impact Platform Release notes Clear

Security Update Guide

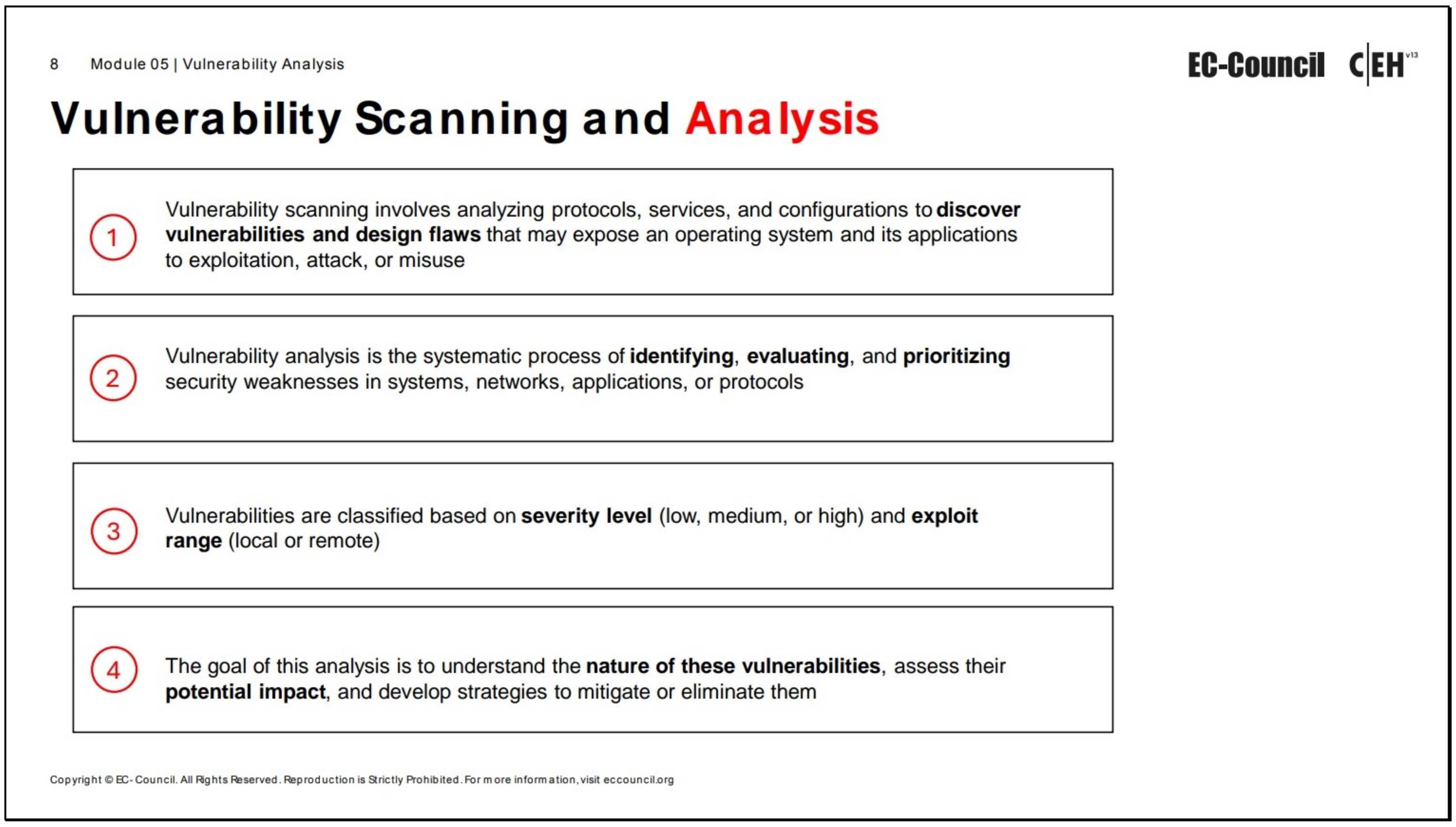
The Microsoft Security Response Center (MSRC) investigates all reports of security vulnerabilities affecting Microsoft products and services, and provides the information here as part of the ongoing effort to help you manage security risks and help keep your systems protected.

Release ...	Product	Platform	Impact	Max Severity	Article	Download
Feb 20, 2024	Microsoft Defender for Endpoint for Windows	Windows 10 Version 22H2 for x64-based Systems	Elevation of Privilege	Important	5032189	Security Update
Feb 20, 2024	Microsoft Defender for Endpoint for Windows	Windows 11 Version 22H2 for ARM64-based Systems	Elevation of Privilege	Important	5032190	Security Update
Feb 20, 2024	Microsoft Defender for Endpoint for Windows	Windows 11 Version 23H2 for ARM64-based Systems	Elevation of Privilege	Important	5032190	Security Update
Feb 20, 2024	Microsoft Defender for Endpoint for Windows	Windows Server 2012 R2 (Server Core installation)	Elevation of Privilege	Important	5034819	Monthly Rollup
Feb 20, 2024	Microsoft Defender for Endpoint for Windows	Windows Server 2012 R2	Elevation of Privilege	Important	5034819	Monthly Rollup
Feb 20, 2024	Microsoft Defender for Endpoint for Windows	Windows Server 2016 (Server Core installation)	Elevation of Privilege	Important	5032197	Security Update
Feb 20, 2024	Microsoft Defender for Endpoint for Windows	Windows Server 2016	Elevation of Privilege	Important	5032197	Security Update
Feb 20, 2024	Microsoft Defender for Endpoint for Windows	Windows 10 Version 1607 for x64-based Systems	Elevation of Privilege	Important	5032197	Security Update
Feb 20, 2024	Microsoft Defender for Endpoint for Windows	Windows 10 Version 1607 for 32-bit Systems	Elevation of Privilege	Important	5032197	Security Update

Figure 5.5: Screenshot of Microsoft Security Response Center (MSRC)

- Packet Storm (*https://packetstormsecurity.com*)
- Dark Reading (*https://www.darkreading.com*)
- Trend Micro (*https://www.trendmicro.com*)
- Security Magazine (*https://www.securitymagazine.com*)
- PenTest Magazine (*https://pentestmag.com*)
- SC Magazine (*https://www.scmagazine.com*)
- Exploit Database (*https://www.exploit-db.com*)
- Help Net Security (*https://www.helpnetsecurity.com*)
- HackerStorm (*https://www.hackerstorm.co.uk*)

- Computerworld (*https://www.computerworld.com*)
- D’Crypt (*https://www.d-crypt.com*)


8 Module 05 | Vulnerability Analysis

EC-Council C|EH v13

Vulnerability Scanning and Analysis

1. Vulnerability scanning involves analyzing protocols, services, and configurations to **discover vulnerabilities and design flaws** that may expose an operating system and its applications to exploitation, attack, or misuse
2. Vulnerability analysis is the systematic process of **identifying**, **evaluating**, and **prioritizing** security weaknesses in systems, networks, applications, or protocols
3. Vulnerabilities are classified based on **severity level** (low, medium, or high) and **exploit range** (local or remote)
4. The goal of this analysis is to understand the **nature of these vulnerabilities**, assess their **potential impact**, and develop strategies to mitigate or eliminate them

Copyright © EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited. For more information, visit eccouncil.org

Vulnerability Scanning and Analysis

Vulnerability scanning involves analyzing protocols, services, and configurations to discover vulnerabilities and design flaws that may expose an operating system and its applications to exploitation, attack, or misuse.

Vulnerability analysis is the systematic process of identifying, evaluating, and prioritizing security weaknesses in systems, networks, applications, or protocols. Vulnerabilities are classified based on severity level (low, medium, or high) and exploit range (local or remote). The goal of this analysis is to understand the nature of these vulnerabilities, assess their potential impact, and develop strategies to mitigate or eliminate them.

Additionally, vulnerability scanning and analysis assist security professionals in securing the network by identifying security loopholes or vulnerabilities in the current security mechanisms before attackers can exploit them.

Typically, vulnerability-scanning tools search network segments for IP-enabled devices and enumerate systems, operating systems, and applications to identify vulnerabilities arising from vendor negligence, system or network misconfigurations, or daily operations. Vulnerability-scanning software compares the scanned systems against the Common Vulnerabilities and Exposures (CVE) index and security bulletins provided by software vendors.

Vulnerability scanners are capable of identifying the following information:

- The OS version running on computers or devices
- IP and Transmission Control Protocol/User Datagram Protocol (TCP/UDP) ports that are listening
- Applications installed on computers

- Accounts with weak passwords
- Files and folders with weak permissions
- Default services and applications that may need to be uninstalled
- Errors in the security configuration of common applications
- Computers exposed to known or publicly reported vulnerabilities
- EOL/EOS software information
- Missing patches and hotfixes
- Weak network configurations and misconfigured or risky ports
- Verification of inventory of all devices on the network

There are two approaches to network vulnerability scanning:

- **Active scanning**: The attacker interacts directly with the target network to find vulnerabilities. Active scanning helps in simulating an attack on the target network to uncover vulnerabilities that can be exploited by the attacker.

  Example: An attacker sends probes and specially crafted requests to the target host in the network to identify vulnerabilities.

- **Passive scanning**: The attacker tries to find vulnerabilities without directly interacting with the target network. The attacker identifies vulnerabilities via information exposed by systems during normal communications. Passive scanning identifies the active operating systems, applications, and ports throughout the target network, monitoring activity to determine its vulnerabilities. This approach provides information about weaknesses but does not provide a path for directly combating attacks.

  Example: An attacker guesses the operating system information, applications, and application/service versions by observing the TCP connection setup and teardown.

Attackers scan for vulnerabilities using tools such as Nessus, Qualys, GFI LanGuard, and OpenVAS.

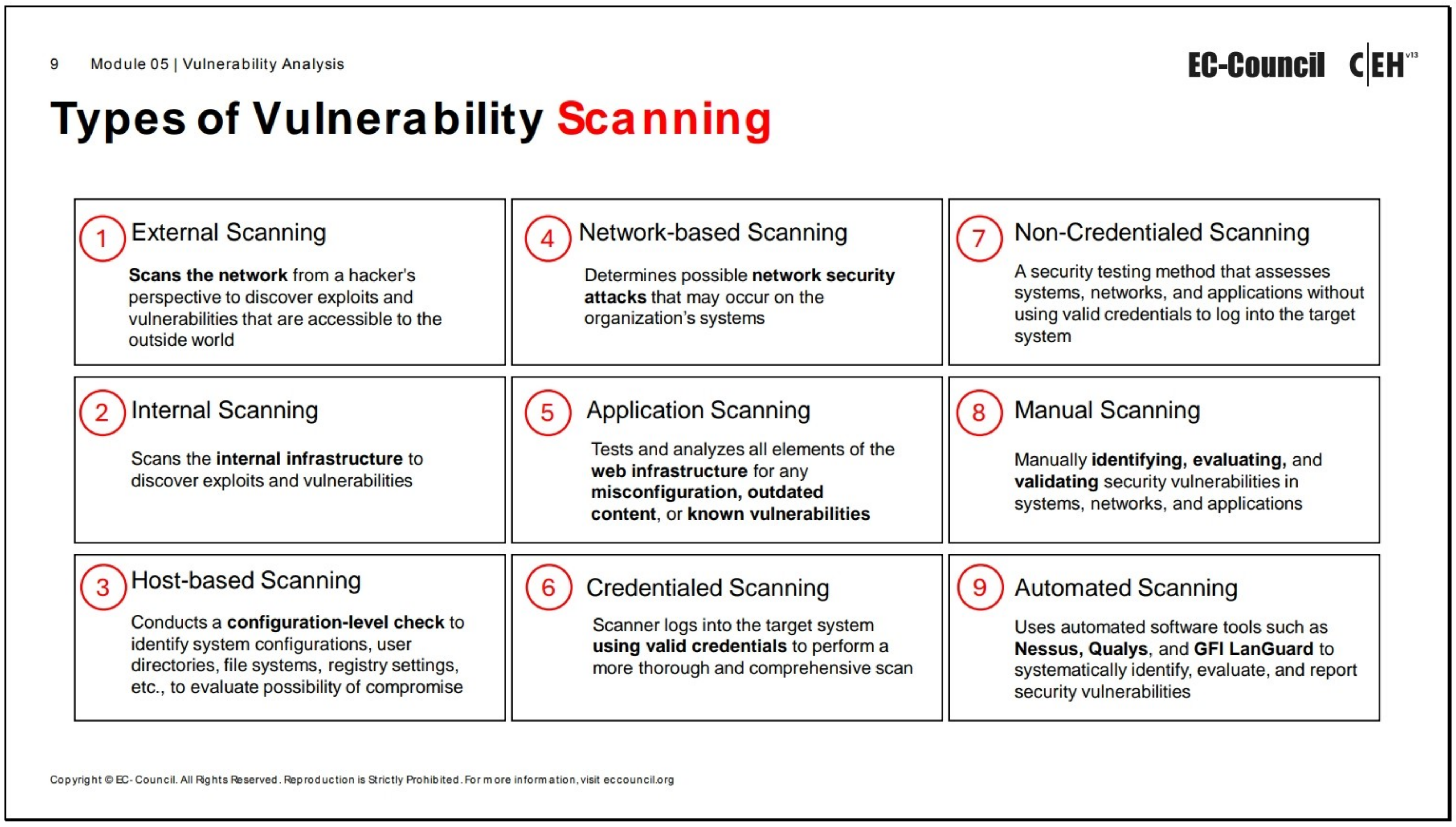
Limitations of Vulnerability Scanning

The following are some of the limitations of vulnerability scanning:

- Vulnerability-scanning software is limited in its ability to detect vulnerabilities at a given point in time.
- Vulnerability-scanning software must be updated when new vulnerabilities are discovered or when improvements are made to the software being used.
- Software is only as effective as the maintenance performed on it by the software vendor and by the administrator who uses it.
- Vulnerability scanning does not measure the strength of security controls.

- Vulnerability-scanning software is not immune to software engineering flaws that might lead to serious vulnerabilities being missed.
- Human judgment is required to analyze the data after scanning and identify false positives and false negatives.
- Vulnerability-scanning software cannot define the impact of an identified vulnerability on different business operations.
- Vulnerability assessment reports are not always easy to understand and assess for risk factors and triage response.
- Vulnerability-scanning tools have a narrow focus and do not cover attack vectors such as social engineering.
- Vulnerability-scanning software is limited in its ability to perform live tests on web applications to detect errors or unexpected behavior.
- Vulnerability-scanning tools rely on known vulnerabilities and thus cannot detect zero-day threats.
- While vulnerability-scanning tools can identify vulnerabilities, they may not effectively prioritize them based on the specific context, such as the criticality of the affected system or the data it hosts.
- Vulnerability-scanning software is largely dependent on the comprehensiveness of the vulnerability databases it uses. If the database is not up-to-date or lacks entries, vulnerabilities may be missed.

The methodology used may have an impact on the test results. For example, vulnerability-scanning software that runs in the security context of the domain administrator will yield different results from software that runs in the security context of an authenticated or non-authenticated user. Similarly, diverse vulnerability-scanning software packages assess security differently and have unique features. This can influence the assessment results.



9 Module 05 | Vulnerability Analysis

Types of Vulnerability Scanning

1 External Scanning **Scans the network** from a hacker's perspective to discover exploits and vulnerabilities that are accessible to the outside world	4 Network-based Scanning Determines possible **network security attacks** that may occur on the organization's systems	7 Non-Credentialed Scanning A security testing method that assesses systems, networks, and applications without using valid credentials to log into the target system
2 Internal Scanning Scans the **internal infrastructure** to discover exploits and vulnerabilities	5 Application Scanning Tests and analyzes all elements of the **web infrastructure** for any **misconfiguration, outdated content**, or **known vulnerabilities**	8 Manual Scanning Manually **identifying, evaluating,** and **validating** security vulnerabilities in systems, networks, and applications
3 Host-based Scanning Conducts a **configuration-level check** to identify system configurations, user directories, file systems, registry settings, etc., to evaluate possibility of compromise	6 Credentialed Scanning Scanner logs into the target system **using valid credentials** to perform a more thorough and comprehensive scan	9 Automated Scanning Uses automated software tools such as **Nessus, Qualys,** and **GFI LanGuard** to systematically identify, evaluate, and report security vulnerabilities

Types of Vulnerability Scanning

Given below are the different types of vulnerability scanning:

- **External Scanning**

  External scanning examines the network from a hacker's point of view to identify exploits and vulnerabilities accessible to the outside world. These types of scans use external devices such as firewalls, routers, and servers. An external scan estimates the threat of network security attacks from outside the organization and determines the level of security of the external network and firewall.

  The following are some possible steps in performing an external scan:

  - Determine a set of rules for firewall and router configurations for the external network
  - Check whether the external server devices and network devices are mapped
  - Identify open ports and related services on the external network
  - Examine the patch levels on the server and external network devices
  - Review detection systems such as IDS, firewalls, and application-layer protection systems
  - Get information on DNS zones
  - Scan the external network through a variety of proprietary tools available on the Internet

- Examine web applications such as e-commerce and shopping cart software for vulnerabilities

- **Internal Scanning**

  Internal scanning involves scrutinizing the internal network to find exploits and vulnerabilities. The following are some of the possible steps in performing an internal scan:

  - Identify the open ports and related services on network devices, servers, and systems
  - Check the router configurations and firewall rule sets
  - List the internal vulnerabilities of the operating system and server
  - Scan for any Trojans that may be present in the internal environment
  - Check the patch levels on the organization's internal network devices, servers, and systems
  - Check for the existence of malware, spyware, and virus activity and document them
  - Evaluate physical security
  - Identify and review the remote management process and events
  - Assess the file-sharing mechanisms (e.g., NFS and SMB/CIFS shares)
  - Examine the antivirus implementation and events

- **Host-based Scanning**

  Host-based scanning is a type of security check that involves conducting a configuration-level examination to identify system configurations, user directories, file systems, registry settings, and other parameters to evaluate the possibility of compromise. These scans check the security of a particular network or server. Host-based scanners assess systems to identify vulnerabilities such as native configuration tables, incorrect registry or file permissions, and software configuration errors. Host-based scanning uses many commercial and open-source tools.

- **Network-based Scanning**

  Network scanning determines the possible network security attacks that may occur on an organization's system. These scans discover network resources and map the ports and services running in various areas of the network. They evaluate the organization's system for vulnerabilities such as missing patches, unnecessary services, weak authentication, and weak encryption. Network scanning professionals use firewalls and network scanners, such as Nessus, to identify open ports, recognize the services running on those ports, and detect vulnerabilities associated with these services. These scans help organizations identify points of entry and attack into a network by following the path and approach of a hacker. They help organizations determine how systems are vulnerable to Internet and intranet attacks and how an attacker can gain access to important information. A typical network scan conducts the following tests on a network:

- Checks the network topologies for inappropriate firewall configuration
- Examines the router filtering rules
- Identifies inappropriately configured database servers
- Tests individual services and protocols such as HTTP, SNMP, and FTP
- Reviews HTML source code for unnecessary information
- Performs bounds checking on variables

- **Application Scanning**

  Application scanning focuses on transactional web applications, traditional client-server applications, and hybrid systems. It analyzes all elements of an application infrastructure, including deployment and communication within the client and server. This type of scan tests the web server infrastructure for any misconfiguration, outdated content, or known vulnerabilities. Security professionals use both commercial and open-source tools to perform such scans.

- **Database Scanning**

  A database scan focuses on testing databases for the presence of any misconfiguration or known vulnerabilities. These scans mainly concentrate on testing various database technologies such as MySQL, MSSQL, Oracle, and PostgreSQL to identify data exposure or injection-type vulnerabilities. Security professionals use both commercial and open-source tools to perform such scans.

- **Wireless Network Scanning**

  Wireless network scanning determines the vulnerabilities in an organization's wireless networks. In the past, wireless networks used weak and defective data encryption mechanisms. Although wireless network standards have evolved, many networks still use weak and outdated security mechanisms and are open to attack. This type of scan tests wireless networks and identifies rogue networks that may exist within an organization's perimeter.

- **Distributed Scanning**

  This type of scanning, employed by organizations with assets such as servers and clients at different locations, involves simultaneously scanning the distributed organization assets, such as client and server applications, using appropriate synchronization techniques. Synchronization plays a critical role in this type of scanning. By synchronizing the test runs, all the separate assets situated at multiple locations can be tested at the same time.

- **Credentialed/Authenticated Scanning**

  Credentialed vulnerability scanning is a security testing method in which the scanner logs into the target system using valid credentials to perform a more thorough and comprehensive scan. This type of scanning provides deeper insights into potential vulnerabilities by accessing areas of the system that are not available to unauthenticated scans.

  Credentialed vulnerability scanning is crucial for identifying and mitigating security risks that could be overlooked by unauthenticated scans, offering a more thorough and accurate assessment of an organization's security posture.

- **Non-Credentialed/Unauthenticated Scanning**

  Non-credentialed vulnerability scanning is a security testing method that assesses systems, networks, and applications without using valid credentials to log into the target system. This type of scanning simulates an external attacker attempting to find and exploit vulnerabilities without having access to user accounts or privileged information.

  Non-credentialed vulnerability scanning is an essential component of a comprehensive security assessment strategy. It helps organizations understand their exposure to external threats and identify immediate vulnerabilities that can be exploited without credentials, thereby strengthening their perimeter defenses.

- **Manual Scanning**

  Manual vulnerability scanning refers to the process of manually identifying, evaluating, and validating security vulnerabilities in systems, networks, and applications without relying solely on automated tools. This approach involves human expertise to perform in-depth analysis, often uncovering issues that automated scanners might miss.

  This type of scanning involves the following:

  - Inspecting source code for insecure coding practices, logic errors, and potential backdoors.
  - Manually checking system and application configurations to ensure they adhere to security best practices. This includes reviewing configuration files, settings, and permissions.
  - Using techniques such as penetration testing to interact with the system and uncover hidden vulnerabilities.

- **Automated Scanning**

  Automated vulnerability scanning refers to the use of software tools such as Nessus, Qualys, and GFI LanGuard to systematically identify, evaluate, and report security vulnerabilities in systems, networks, applications, or devices without requiring manual intervention. These tools use predefined rules, databases of known vulnerabilities, and various scanning techniques to detect potential security issues efficiently and consistently. They can be configured to run scans on a regular schedule, ensuring that new vulnerabilities are identified promptly.

- **Cloud-based Scanning**

  This type of scanning focuses on evaluating overall security of the cloud infrastructure according to the cloud service provider's best practices or guidelines. This scanning involves identifying cloud infrastructure vulnerabilities and mitigating them through access control mechanisms and proper security measures complying with the standards. This type of scanning is frequently performed to identify the risks associated with the assets deployed over the cloud. It also assists security professionals to detect weak entry points on the cloud, through which the attackers can make their way into the organization's network.

- **Mobile Application Scanning**

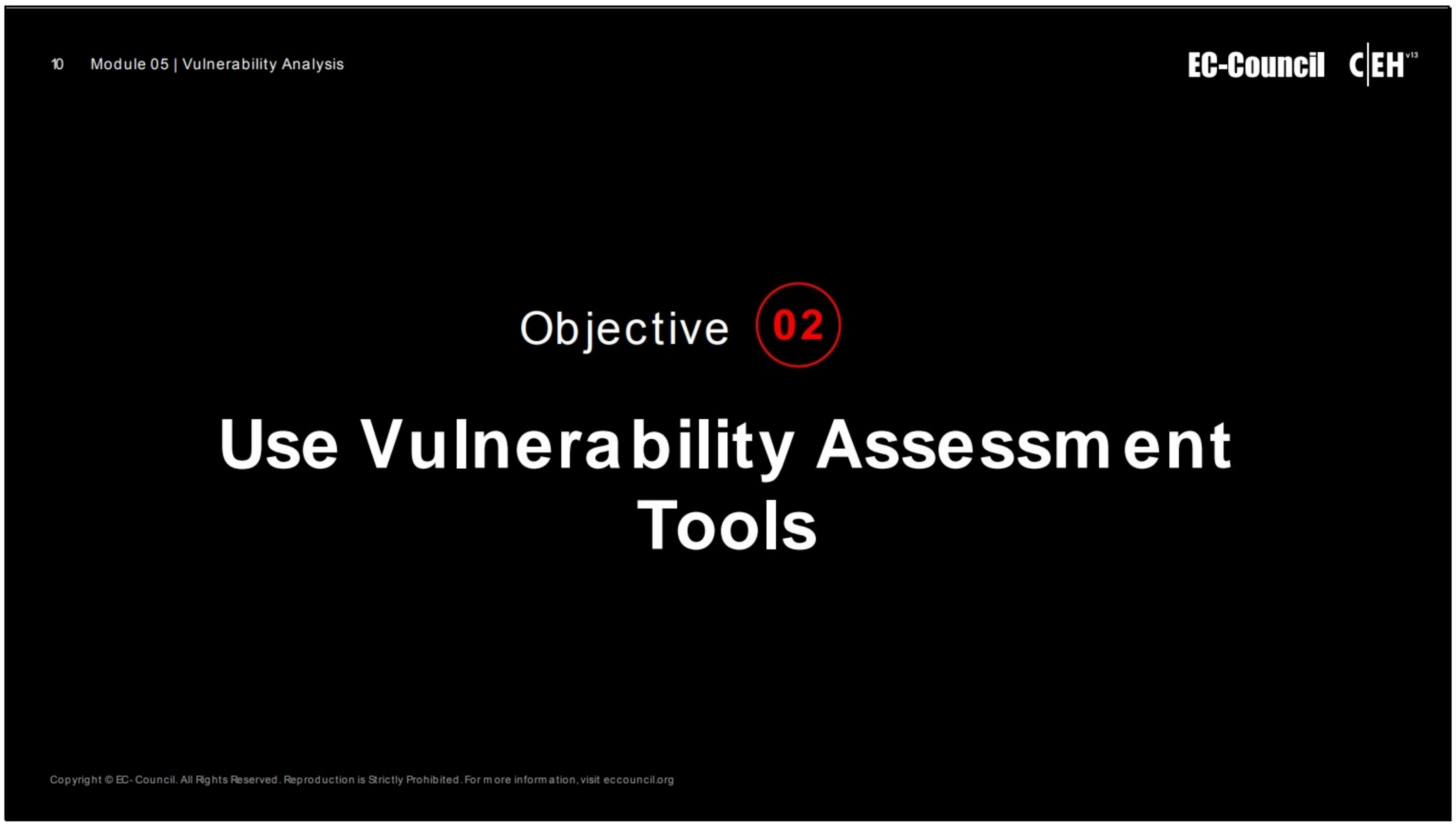
  Mobile application scanning aims at protecting the privacy of data across mobile applications and APIs. It is a must-have security practice for every organization that hosts publicly accessible applications. This type of scanning involves examining source code and internal security controls of mobile applications. Security professionals need to perform this type of scanning to evaluate and improve the overall application's strength against known and future threats to protect sensitive data. An effective scanning can minimize risks and assists in incorporating appropriate security controls to increase the safety of mobile applications.

- **Physical Security Vulnerability Scanning**

  Physical security vulnerability scanning involves conducting a comprehensive examination of physical assets to proactively identify various vulnerabilities associated with them. By identifying potential vulnerabilities across physical infrastructure and the current security posture, organizations can develop robust mitigation strategies. These assessments play an essential role in enhancing overall security, protecting physical assets against imminent threats and vulnerabilities. Additionally, these assessments provide actionable insights and help in making informed decisions to effectively minimize physical security threats.

- **IoT Device Vulnerability Scanning**

  IoT device vulnerability scanning provides insights into weaknesses across IoT devices and systems that are exposed to or connected to the Internet. Insecure placement or implementation of IoT devices can compromise their privacy and connect them to fraudulent networks. Ethical hackers should assess IoT devices, including their hardware, software, communication protocols, and data exchange, to identify potential vulnerabilities. This assessment can be accomplished using various techniques such as penetration testing, code review, and vulnerability scanning. An effective assessment can minimize vulnerabilities and enable ethical hackers to incorporate appropriate security controls, increasing the safety of IoT devices and systems within the organization.



10 Module 05 | Vulnerability Analysis

Objective 02

Use Vulnerability Assessment Tools

Copyright © EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited. For more information, visit eccouncil.org

Vulnerability Assessment Tools

Vulnerability assessment solutions are important tools for information security management as they identify all potential security weaknesses before an attacker can exploit them. There are different approaches and solutions available to perform a vulnerability assessment. Selecting an appropriate assessment approach plays a major role in mitigating the threats that an organization faces.

This section outlines the various approaches, solutions, and tools used to perform a vulnerability assessment.

Comparing Approaches to Vulnerability Assessment

There are four types of vulnerability assessment solutions: product-based solutions, service-based solutions, tree-based assessment, and inference-based assessment.

- **Product-Based Solutions**

  Product-based solutions are installed in the organization's internal network. They are installed either on a private or non-routable space or in the Internet-addressable portion of an organization's network. If they are installed on a private network (behind the firewall), they cannot always detect outside attacks.

- **Service-Based Solutions**

  Service-based solutions are offered by third parties, such as auditing or security consulting firms. Some solutions are hosted inside the network, while others are hosted outside the network. A drawback of this solution is that attackers can perform network vulnerability scans from the Internet/external network.

- **Tree-Based Assessment**

  In a tree-based assessment, the auditor selects different strategies for each machine or component of the information system. For example, the administrator selects a scanner for servers running Windows, databases, and web services but uses a different scanner for Linux servers. This approach relies on the administrator to provide a starting piece of intelligence, and then to start scanning continuously without incorporating any information found at the time of scanning.

- **Inference-Based Assessment**

  In an inference-based assessment, scanning starts by building an inventory of the protocols found on the machine. After finding a protocol, the scanning process starts to detect which ports are attached to services, such as an email server, web server, or database server. After finding services, it selects vulnerabilities on each machine and starts to execute only those relevant tests.

Characteristics of a Good Vulnerability Assessment Solution

Organizations need to select a proper and suitable vulnerability assessment solution to detect, assess, and protect their critical IT assets from various internal and external threats.

The characteristics of a good vulnerability assessment solution are as follows:

- Covers a wide range of assets, including networks, servers, endpoints, web applications, and cloud services.
- Ensures correct outcomes by testing the network, network resources, ports, protocols, and operating systems
- Uses a well-organized inference-based approach for testing
- Automatically scans and checks against continuously updated databases
- Creates brief, actionable, customizable reports, including reports of vulnerabilities by severity level, and trend analysis
- Supports multiple networks
- Suggests appropriate remedies and workarounds to correct vulnerabilities
- Imitates the outside view of attackers to gain its objective
- Identifies vulnerabilities with minimal false positives and negatives, ensuring resources are not wasted on non-issues or overlooking real threats.
- Supports real-time scanning and receives frequent updates on new vulnerabilities to ensure the system can identify and protect against the latest threats.
- Supports automation of scanning and response processes and integrates with security orchestration tools to streamline vulnerability management.

- Incorporates risk-based prioritization, considering the severity of the vulnerability, the criticality of the affected asset, and the context within the organization's environment to help prioritize remediation efforts effectively.

Working of Vulnerability Scanning Solutions

Any organization needs to handle and process large volumes of data to conduct business. These large volumes of data contain privileged information of that particular organization. Attackers try to identify vulnerabilities that they can exploit, and then use these to gain access to critical data for illegal purposes. Vulnerability analysis analyzes and detects risk-prone areas in the organizational network. This analysis uses various tools and reports on the vulnerabilities present in the network.

Vulnerability scanning solutions perform vulnerability penetration tests on the organizational network in three steps:

- **Locating nodes**: The first step in vulnerability scanning is to locate live hosts in the target network using various scanning techniques.
- **Performing service and OS discovery on them**: After detecting the live hosts in the target network, the next step is to enumerate the open ports and services along with the operating system on the target systems.
- **Testing those services and OS for known vulnerabilities**: Finally, after identifying the open services and the operating system running on the target nodes, they are tested for known vulnerabilities.

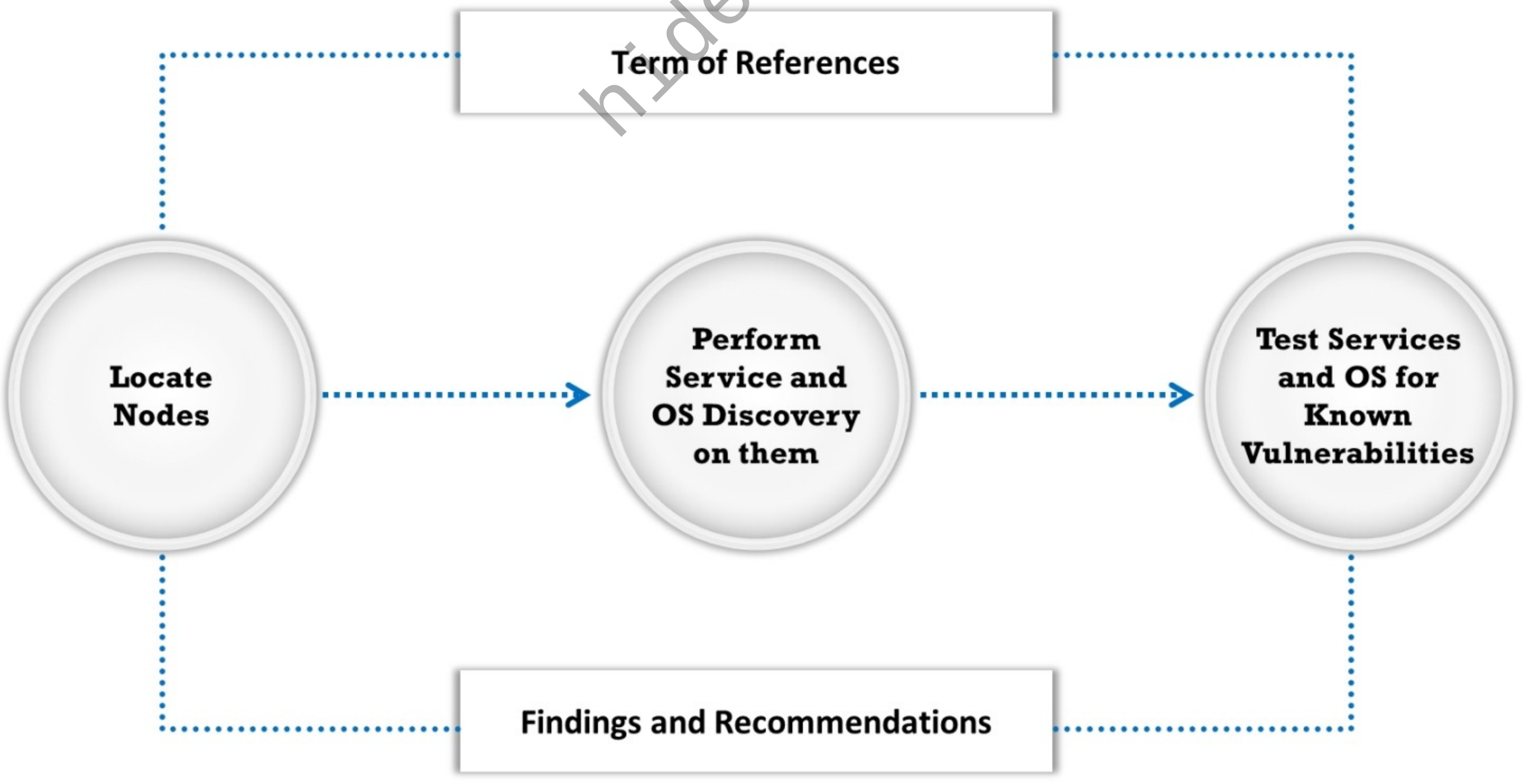

Term of References
Locate Nodes
Perform Service and OS Discovery on them
Test Services and OS for Known Vulnerabilities
Findings and Recommendations

Figure 5.6: The working of vulnerability scanning solutions

Types of Vulnerability Assessment Tools

There are six types of vulnerability assessment tools: host-based vulnerability assessment tools, application-layer vulnerability assessment tools, depth assessment tools, scope assessment tools, active and passive tools, and location and data-examination tools.

- **Host-Based Vulnerability Assessment Tools**

  The host-based scanning tools are appropriate for servers that run various applications, such as the Web, critical files, databases, directories, and remote accesses. These host-based scanners can detect high levels of vulnerabilities and provide required information about the fixes (patches). A host-based vulnerability assessment tool identifies the OS running on a particular host computer and tests it for known deficiencies. It also searches for common applications and services.

- **Depth Assessment Tools**

  Depth assessment tools are used to discover and identify previously unknown vulnerabilities in a system. Generally, tools such as fuzzers, which provide arbitrary input to a system's interface, are used to identify vulnerabilities to an unstable depth. Many of these tools use a set of vulnerability signatures to test whether a product is resistant to a known vulnerability or not.

- **Application-Layer Vulnerability Assessment Tools**

  Application-layer vulnerability assessment tools are designed to serve the needs of all kinds of operating system types and applications. Various resources pose a variety of security threats and are identified by the tools designed for that purpose. Observing system vulnerabilities through the Internet using an external router, firewall, or webserver is called an external vulnerability assessment. These vulnerabilities could be external DoS/DDoS threats, network data interception, or other issues. The analyst performs a vulnerability assessment and notes vulnerable resources. The network vulnerability information is updated regularly into the tools. Application-layer vulnerability assessment tools are directed towards web servers or databases.

- **Scope Assessment Tools**

  Scope assessment tools provide an assessment of the security by testing vulnerabilities in the applications and operating system. These tools provide standard controls and a reporting interface that allows the user to select a suitable scan. These tools generate a standard report based on the information found. Some assessment tools are designed to test a specific application or application type for vulnerability.

- **Active and Passive Tools**

  Active scanners perform vulnerability checks on the network functions that consume resources on the network. The main advantage of the active scanner is that the system administrator or IT manager has good control of the timing and the parameters of vulnerability scans. This scanner cannot be used for critical operating systems because it uses system resources that affect the processing of other tasks.

Passive scanners are those that do not considerably affect system resources, as they only observe system data and perform data processing on a separate analysis machine. A passive scanner first receives system data that provide complete information on the processes that are running and then assesses that data against a set of rules.

- **Location and Data Examination Tools**

  Listed below are some of the location and data examination tools:

  - **Network-Based Scanner:** Network-based scanners are those that interact only with the real machine where they reside and give the report to the same machine after scanning.
  - **Agent-Based Scanner:** Agent-based scanners reside on a single machine but can scan several machines on the same network.
  - **Proxy Scanner:** Proxy scanners are the network-based scanners that can scan networks from any machine on the network.
  - **Cluster scanner:** Cluster scanners are similar to proxy scanners, but they can simultaneously perform two or more scans on different machines in the network.

Choosing a Vulnerability Assessment Tool

Vendor-designed vulnerability assessment tools can be used to test a host or application for vulnerabilities. There are several available vulnerability assessment tools that include port scanners, vulnerability scanners, and OS vulnerability assessment scanners. Organizations must choose appropriate tools based on their test requirements.

Choose the tools that best satisfy the following requirements:

- Tools must be capable of testing anywhere from dozens to 30,000 different vulnerabilities, depending on the product
- The selected tool should have a sound database of vulnerabilities and frequently updated attack signatures
- Pick a tool that matches the environment and expertise
- Make sure to regularly update the scan engine to ensure the tool is aware of the latest known vulnerabilities
- Verify that the chosen vulnerability assessment tool has accurate network mapping, application mapping, and penetration tests. Not all tools can find the protocols running and analyze a network's performance.
- Ensure that the tool has several regularly updated vulnerability scripts for the platforms you are scanning
- Make sure that any patches are applied; failing to do so might lead to false positives
- Find out how many reports are returned, what information they contain, and whether they are exportable

- Check whether the tool has different levels of penetration to stop lockups
- The maintenance costs of tools can be offset by effectively using them
- Ensure that the vulnerability assessment tool can run its scans quickly and accurately
- Ensure that the tool can perform scans using multiple protocols
- Verify that the tool can understand and analyze the network topology to perform the assessment
- Bandwidth limitations are a major concern when dealing with large networks. Ensure the vulnerability assessment tool has high bandwidth allocation
- Ensure that the vulnerability assessment tool possess excellent query throttling features
- Ensure that the tool can also assess fragile systems and non-traditional assets

Criteria for Choosing a Vulnerability Assessment Tool

Vulnerabilities often arise based on several factors such as the complexity of software and systems, the number of software components in use, the level of scrutiny applied by the developers, and the evolving nature of cyber threats. Therefore, choosing the right vulnerability assessment tool is crucial for organizations to ensure the security of systems and data.

The criteria to follow when choosing or purchasing any vulnerability assessment tool are as follows:

- **Types of vulnerabilities being assessed**: The most important information at the time of evaluating any tool is to find out how many types of vulnerabilities it will discover.
- **Testing capability of scanning**: The vulnerability assessment tool must have the capacity to execute the entire selected test and must scan all the systems selected for scanning.
- **Ability to provide accurate reports**: The ability to prepare an accurate report is essential. Vulnerability reports should be short, clear, and should provide an easy method to mitigate the discovered vulnerability.
- **Efficient and accurate scanning**: Two essential aspects of scanner performance are how much time it takes for a single host and what resources are required, and the loss of services at the time of scanning. It is important to ensure accuracy and to be aware of the accuracy of the results.
- **Capability to perform a smart search**: How clever they are at the time of scanning is also a key factor in judging any vulnerability assessment tool.
- **Functionality for writing its own tests**: When a signature is not present for a recently found vulnerability, it is helpful if the vulnerability scanning tool allows the use of user-developed tests.
- **Test run scheduling**: It is important to be able to do test-run scheduling as it allows users to perform scanning when traffic on the network is light.

- **Speed:** Evaluate the tool's scanning speed to ensure timely detection of vulnerabilities without causing significant performance overhead on the network or organizational systems. It is also important to evaluate the quality of findings obtained by the tool.
- **Compatibility:** Ensure that the vulnerability assessment tool possesses compatibility across various dimensions to effectively scan and assess vulnerabilities in diverse IT environments. Also, evaluate the tool's compatibility with legacy systems since they are prone to frequent vulnerabilities.
- **Configuration support:** Alongside compatibility, the tool should also support all types of configurations or settings to conduct frequent scans across different environments, including on-premises and cloud. This support is crucial for any assessment tool to effectively address vulnerabilities based on the unique requirements of each organization.
- **Compliance:** Check whether the assessment tool meets relevant industry standards, regulations, best practices, and all business requirements. A tool with good compliance allows security auditors to assess and manage vulnerabilities in the IT infrastructure while adhering to legal requirements and industry guidelines.
- **Licensing model and cost:** Evaluate the licensing model offered by the vendor, whether it is based on the number of assets/devices, IP addresses, users, or modules/features. Choose a licensing model that best fits your organization's size, scale, and budget.
- **Scalability and performance:** Ensure the tool is capable of handling increasing scan volumes without significant degradation in network operations. It should efficiently utilize system resources such as CPU, memory, and network bandwidth to maintain scanning speed and accuracy, even if the number of assets and vulnerabilities grows.

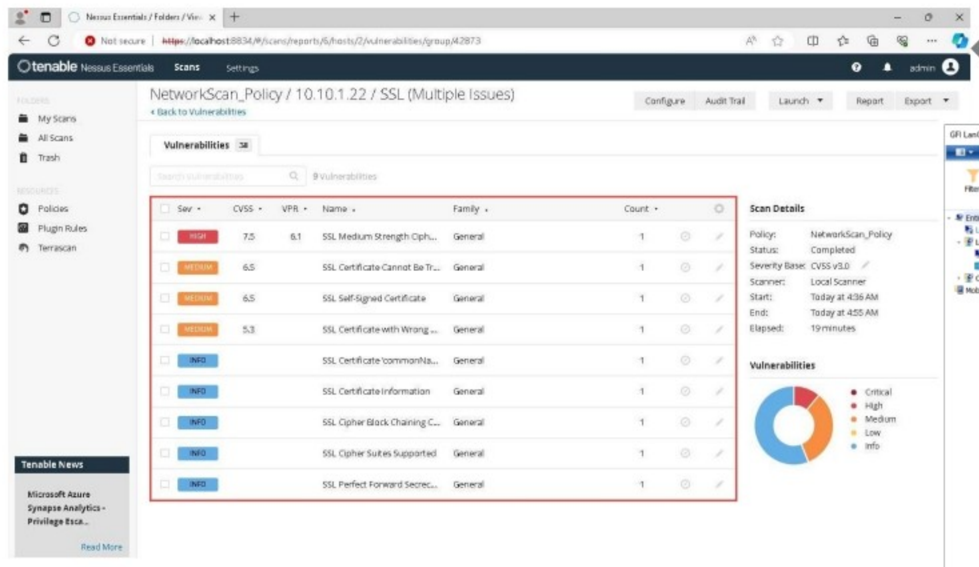
Best Practices for Selecting Vulnerability Assessment Tools

Some of the best practices that can be adopted for selecting vulnerability assessment tools are:

- Vulnerability assessment tools are used to secure and protect the organization's system or network. Ensure that they do not damage the network or system while running.
- Before using any vulnerability assessment tools, it is important to understand their function and to decide what information is needed before starting
- Security mechanisms for accessing from within and from outside the network are somewhat different, so decide the location for the scan based on the desired information
- At the time of scanning, enable logging and ensure that all outcomes and methodologies are annotated every time a scan is performed on any computer
- Users should frequently scan their systems for vulnerabilities and regularly monitor them for vulnerabilities and exploits

EC-Council C|EH v13

Vulnerability Assessment Tools: Nessus Essentials and GFI LanGuard



NetworkScan_Policy / 10.10.1.22 / SSL (Multiple Issues)

Sev	CVSS	VPR	Name	Family	Count
HIGH	7.5	6.1	SSL Medium Strength Ciph...	General	1
MEDIUM	6.5		SSL Certificate Cannot Be Tr...	General	1
MEDIUM	6.5		SSL Self-Signed Certificate	General	1
MEDIUM	5.3		SSL Certificate with Wrong ...	General	1
INFO			SSL Certificate 'commonNa...	General	1
INFO			SSL Certificate Information	General	1
INFO			SSL Cipher Block Chaining C...	General	1
INFO			SSL Cipher Suites Supported	General	1
INFO			SSL Perfect Forward Secrec...	General	1

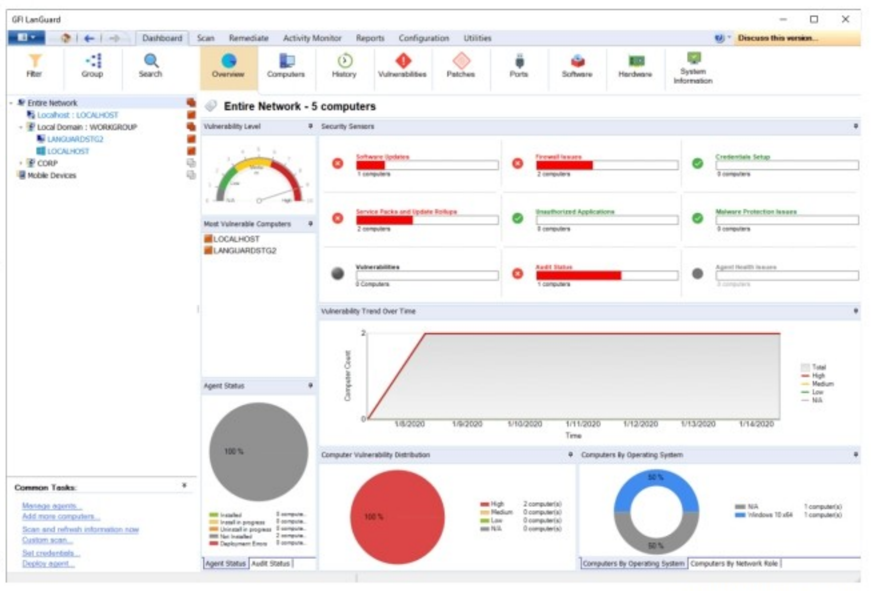
Scan Details

Policy:	NetworkScan_Policy
Status:	Completed
Severity Base:	CVSS v3.0
Scanner:	Local Scanner
Start:	Today at 4:36 AM
End:	Today at 4:55 AM
Elapsed:	19 minutes

https://www.tenable.com

Nessus Essentials is an assessment solution for identifying **vulnerabilities**, **configuration issues**, and **malware**, which can be used to penetrate networks

GFI LanGuard scans, detects, assesses, and rectifies **security vulnerabilities** in a network and connected devices

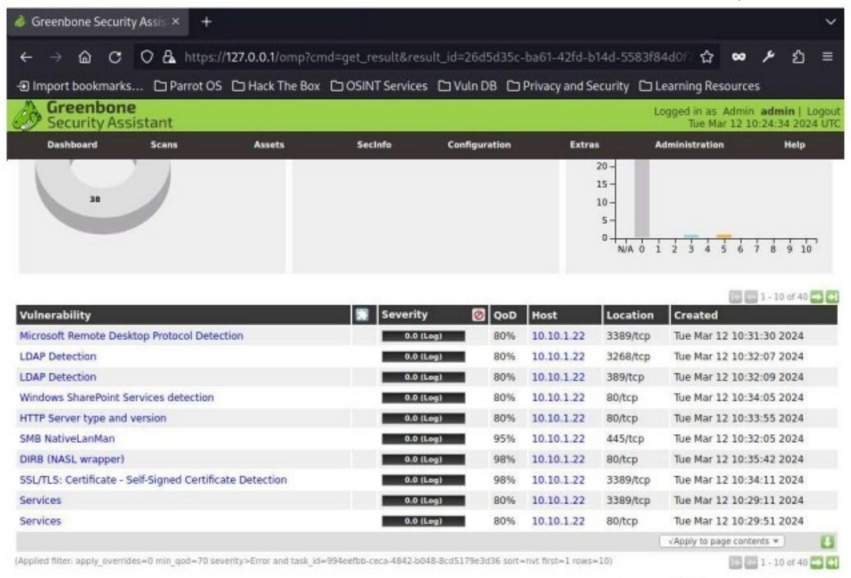


https://www.gfi.com

EC-Council C|EH v13

Vulnerability Assessment Tools: OpenVAS and Nikto

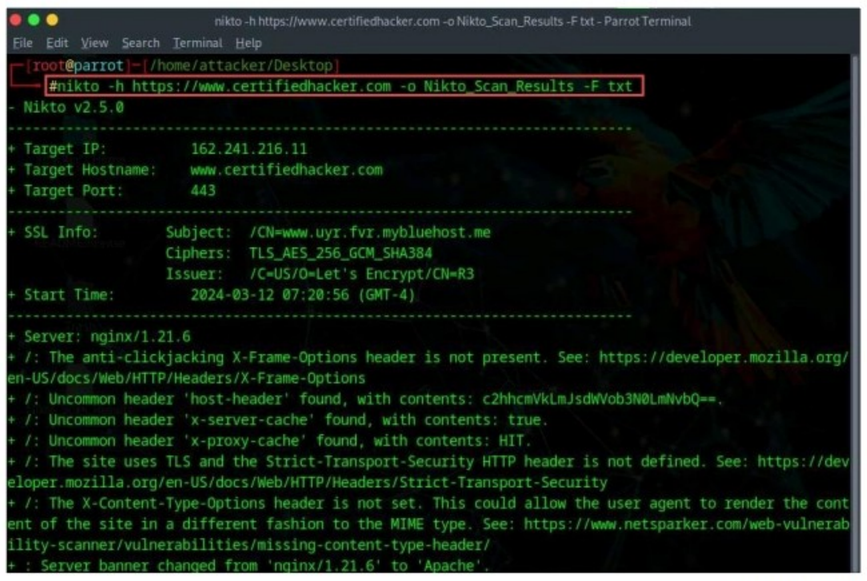
OpenVAS	A framework of several services and tools offering a comprehensive and powerful **vulnerability scanning** and **vulnerability management solution**
Nikto	A **web server assessment tool** that examines a web server to discover potential problems and security vulnerabilities



Vulnerability	Severity	QoD	Host	Location	Created
Microsoft Remote Desktop Protocol Detection	0.0 (Log)	80%	10.10.1.22	3389/tcp	Tue Mar 12 10:31:30 2024
LDAP Detection	0.0 (Log)	80%	10.10.1.22	3268/tcp	Tue Mar 12 10:32:07 2024
LDAP Detection	0.0 (Log)	80%	10.10.1.22	389/tcp	Tue Mar 12 10:32:09 2024
Windows SharePoint Services detection	0.0 (Log)	80%	10.10.1.22	80/tcp	Tue Mar 12 10:34:05 2024
HTTP Server type and version	0.0 (Log)	80%	10.10.1.22	80/tcp	Tue Mar 12 10:33:55 2024
SMB NativeLanMan	0.0 (Log)	95%	10.10.1.22	445/tcp	Tue Mar 12 10:32:05 2024
DIRB (NASL wrapper)	0.0 (Log)	98%	10.10.1.22	80/tcp	Tue Mar 12 10:35:42 2024
SSL/TLS: Certificate - Self-Signed Certificate Detection	0.0 (Log)	98%	10.10.1.22	3389/tcp	Tue Mar 12 10:34:11 2024
Services	0.0 (Log)	80%	10.10.1.22	3389/tcp	Tue Mar 12 10:29:11 2024
Services	0.0 (Log)	80%	10.10.1.22	80/tcp	Tue Mar 12 10:29:51 2024

(Applied filter: apply_overrides=0 min_qod=70 severity>Error and task_id=994eefbb-ceca-4842-b048-8cd5179e3d36 sort=nvt first=1 rows=10)

https://www.openvas.org



```
┌─[root@parrot]─[/home/attacker/Desktop]
└──╼ #nikto -h https://www.certifiedhacker.com -o Nikto_Scan_Results -F txt
- Nikto v2.5.0
---------------------------------------------------------------------------
+ Target IP:          162.241.216.11
+ Target Hostname:    www.certifiedhacker.com
+ Target Port:        443
---------------------------------------------------------------------------
+ SSL Info:        Subject:  /CN=www.uyr.fvr.mybluehost.me
                   Ciphers:  TLS_AES_256_GCM_SHA384
                   Issuer:   /C=US/O=Let's Encrypt/CN=R3
+ Start Time:         2024-03-12 07:20:56 (GMT-4)
---------------------------------------------------------------------------
+ Server: nginx/1.21.6
+ /: The anti-clickjacking X-Frame-Options header is not present. See: https://developer.mozilla.org/
en-US/docs/Web/HTTP/Headers/X-Frame-Options
+ /: Uncommon header 'host-header' found, with contents: c2hhcmVkLmJsdWVob3N0LmNvbQ==.
+ /: Uncommon header 'x-server-cache' found, with contents: true.
+ /: Uncommon header 'x-proxy-cache' found, with contents: HIT.
+ /: The site uses TLS and the Strict-Transport-Security HTTP header is not defined. See: https://dev
eloper.mozilla.org/en-US/docs/Web/HTTP/Headers/Strict-Transport-Security
+ /: The X-Content-Type-Options header is not set. This could allow the user agent to render the cont
ent of the site in a different fashion to the MIME type. See: https://www.netsparker.com/web-vulnerab
ility-scanner/vulnerabilities/missing-content-type-header/
+ : Server banner changed from 'nginx/1.21.6' to 'Apache'.
```

https://cirt.net

Vulnerability Assessment Tools

An attacker performs vulnerability scanning to identify security loopholes in the target network that they can exploit to launch attacks. Security analysts can use vulnerability assessment tools to identify weaknesses present in the organization's security posture and remediate the identified vulnerabilities before an attacker exploits them.

Network vulnerability scanners help to analyze and identify vulnerabilities in the target network or network resources by using vulnerability assessment and network auditing. These tools also assist in overcoming weaknesses in the network by suggesting various remediation techniques.

The following are some of the most effective vulnerability assessment tools:

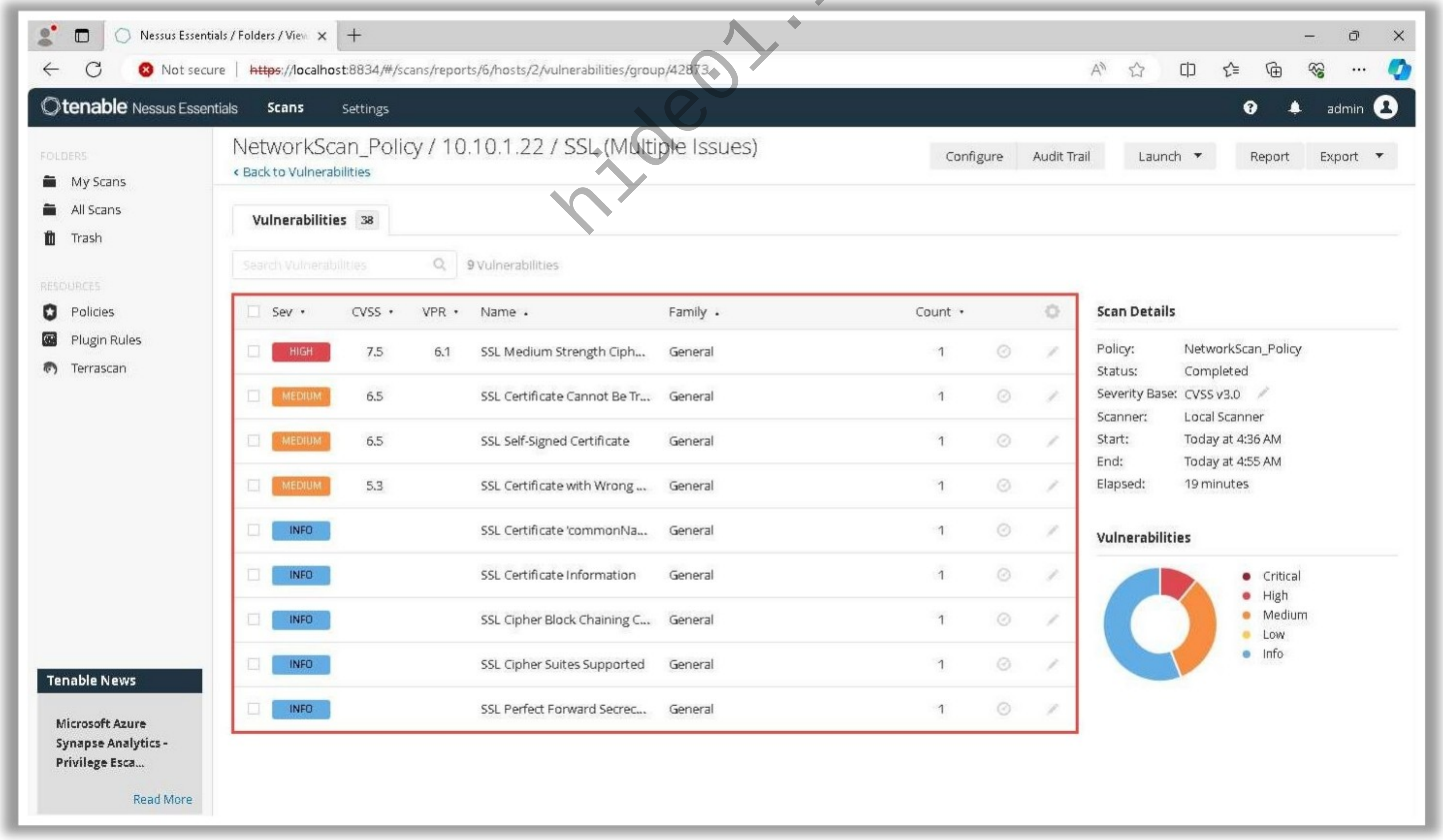
- **Nessus Essentials**

  Source: *https://www.tenable.com*

  Nessus Essentials is an assessment solution for identifying vulnerabilities, configuration issues, and malware, which can be used to penetrate networks. It also helps ethical hackers perform vulnerability, configuration, and compliance assessment. It supports various technologies such as OSes, network devices, hypervisors, databases, tablets/phones, web servers, and critical infrastructure.

  Features:

  - High-speed asset discovery
  - Vulnerability assessment
  - Malware and botnet detection
  - Scanning and auditing virtualized and cloud platforms



NetworkScan_Policy / 10.10.1.22 / SSL (Multiple Issues)

Sev	CVSS	VPR	Name	Family	Count
HIGH	7.5	6.1	SSL Medium Strength Ciph...	General	1
MEDIUM	6.5		SSL Certificate Cannot Be Tr...	General	1
MEDIUM	6.5		SSL Self-Signed Certificate	General	1
MEDIUM	5.3		SSL Certificate with Wrong ...	General	1
INFO			SSL Certificate 'commonNa...	General	1
INFO			SSL Certificate Information	General	1
INFO			SSL Cipher Block Chaining C...	General	1
INFO			SSL Cipher Suites Supported	General	1
INFO			SSL Perfect Forward Secrec...	General	1

Scan Details

Policy:	NetworkScan_Policy
Status:	Completed
Severity Base:	CVSS v3.0
Scanner:	Local Scanner
Start:	Today at 4:36 AM
End:	Today at 4:55 AM
Elapsed:	19 minutes

Vulnerabilities

Tenable News

Microsoft Azure Synapse Analytics - Privilege Esca...

Figure 5.7: Vulnerability scanning using Nessus Essentials

- **GFI LanGuard**

  Source: *https://www.gfi.com*

  GFI LanGuard scans for, detects, assesses, and rectifies security vulnerabilities in a network and its connected devices. This is done with minimal administrative effort. It scans the operating systems, virtual environments, and installed applications through vulnerability check databases. It enables analysis of the state of network security, identifies risks, and offers solutions before the system can be compromised.

  Features:

  - Patch management for operating systems and third-party applications
  - Vulnerability assessment
  - A Web reporting console
  - Track latest vulnerabilities and missing updates
  - Integration with security applications
  - Network device vulnerability checks
  - Network and software auditing
  - Support for virtual environments

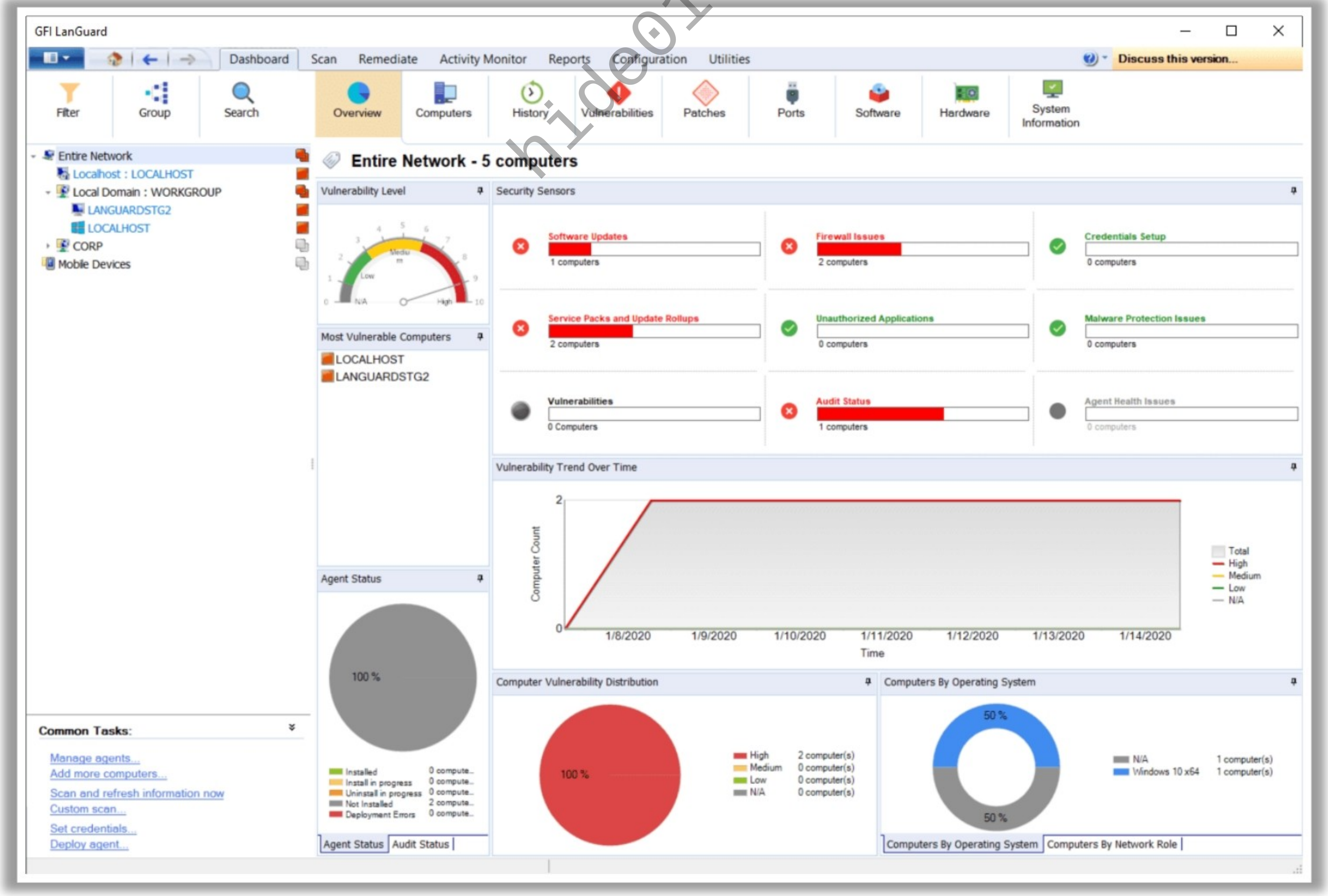


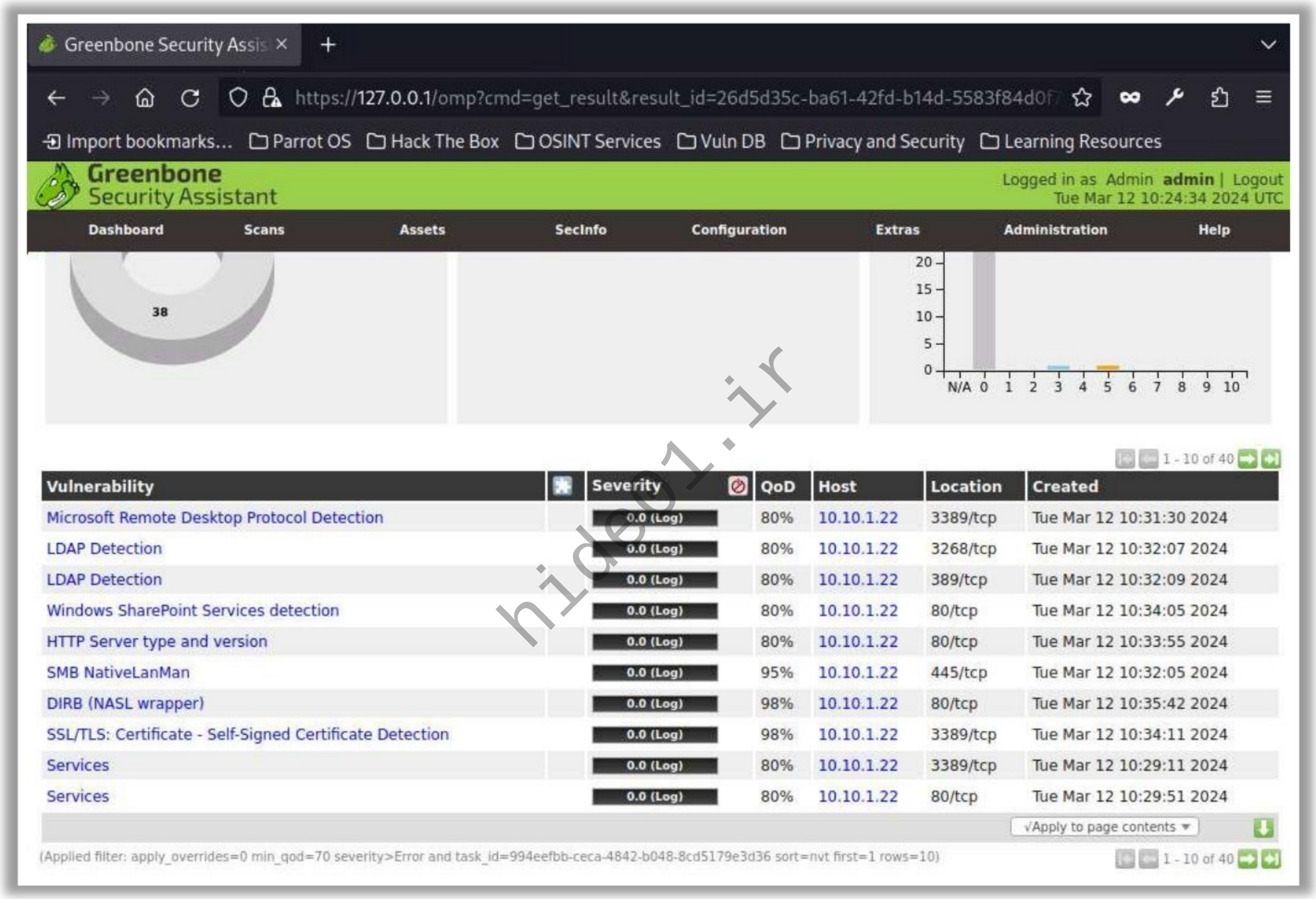
Figure 5.8: Vulnerability scanning using GFI LanGuard

- **OpenVAS**

  Source: *https://www.openvas.org*

  OpenVAS is a framework of several services and tools that offer a comprehensive and powerful vulnerability scanning and vulnerability management solution. The framework is part of Greenbone Network's commercial vulnerability management solution, developments from which have been contributed to the open-source community since 2009.

  The actual security scanner is accompanied by a regularly updated feed of Network Vulnerability Tests (NVTs), over 50,000 in total.



Vulnerability		Severity	QoD	Host	Location	Created
Microsoft Remote Desktop Protocol Detection		0.0 (Log)	80%	10.10.1.22	3389/tcp	Tue Mar 12 10:31:30 2024
LDAP Detection		0.0 (Log)	80%	10.10.1.22	3268/tcp	Tue Mar 12 10:32:07 2024
LDAP Detection		0.0 (Log)	80%	10.10.1.22	389/tcp	Tue Mar 12 10:32:09 2024
Windows SharePoint Services detection		0.0 (Log)	80%	10.10.1.22	80/tcp	Tue Mar 12 10:34:05 2024
HTTP Server type and version		0.0 (Log)	80%	10.10.1.22	80/tcp	Tue Mar 12 10:33:55 2024
SMB NativeLanMan		0.0 (Log)	95%	10.10.1.22	445/tcp	Tue Mar 12 10:32:05 2024
DIRB (NASL wrapper)		0.0 (Log)	98%	10.10.1.22	80/tcp	Tue Mar 12 10:35:42 2024
SSL/TLS: Certificate - Self-Signed Certificate Detection		0.0 (Log)	98%	10.10.1.22	3389/tcp	Tue Mar 12 10:34:11 2024
Services		0.0 (Log)	80%	10.10.1.22	3389/tcp	Tue Mar 12 10:29:11 2024
Services		0.0 (Log)	80%	10.10.1.22	80/tcp	Tue Mar 12 10:29:51 2024

(Applied filter: apply_overrides=0 min_qod=70 severity>Error and task_id=994eefbb-ceca-4842-b048-8cd5179e3d36 sort=nvt first=1 rows=10)

Figure 5.9: Vulnerability scanning using OpenVAS

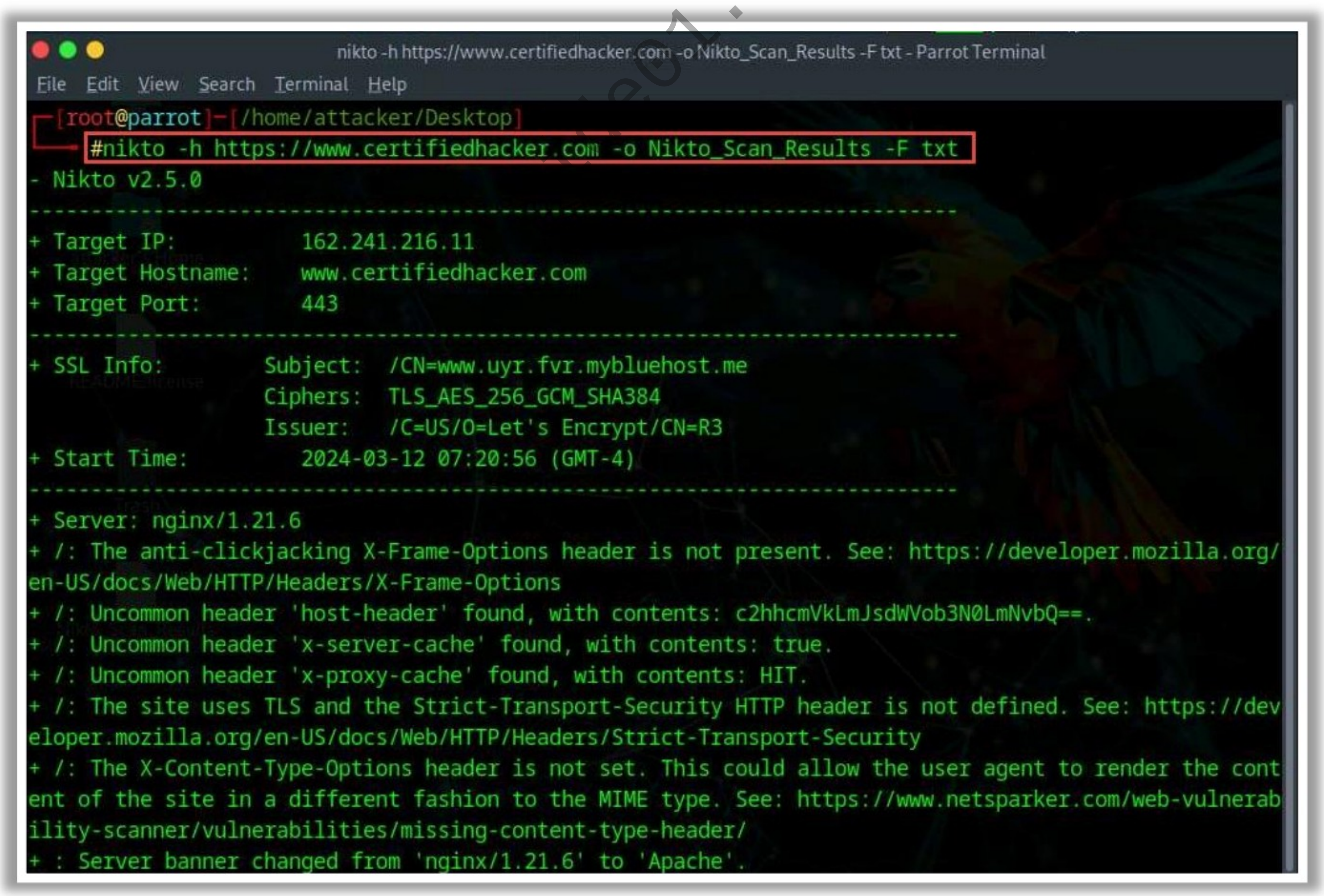
- **Nikto**

  Source: *https://cirt.net*

  Nikto is an Open Source (GPL) web server scanner that performs comprehensive tests against web servers for multiple items, including over 6700 potentially dangerous files or programs, checks for outdated versions of over 1250 servers, and checks for version specific problems on over 270 servers. It also looks at server configuration items such as the presence of multiple index files and the HTTP server options and will attempt to identify installed web servers and software.

Features:

- SSL Support (Unix with OpenSSL or maybe Windows with ActiveState's Perl/NetSSL)
- A full HTTP proxy support
- Checks for outdated server components
- Saves reports in plain text, XML, HTML, NBE or CSV
- A Template engine to easily customize reports
- Scans multiple ports on a server, or multiple servers via input file
- LibWhisker's IDS encoding techniques
- Identifies installed software via headers, favicons, and files
- Host authentication with Basic and NTLM
- Subdomain guessing
- Apache and cgiwrap username enumeration
- Scan tuning to include or exclude entire classes of vulnerability checks
- Guesses credentials for authorization realms (including many default ID and password combinations)


nikto -h https://www.certifiedhacker.com -o Nikto_Scan_Results -F txt - Parrot Terminal
File Edit View Search Terminal Help

```
┌─[root@parrot]─[/home/attacker/Desktop]
└──╼ #nikto -h https://www.certifiedhacker.com -o Nikto_Scan_Results -F txt
- Nikto v2.5.0
---------------------------------------------------------------------------
+ Target IP:          162.241.216.11
+ Target Hostname:    www.certifiedhacker.com
+ Target Port:        443
---------------------------------------------------------------------------
+ SSL Info:        Subject:  /CN=www.uyr.fvr.mybluehost.me
                   Ciphers:  TLS_AES_256_GCM_SHA384
                   Issuer:   /C=US/O=Let's Encrypt/CN=R3
+ Start Time:         2024-03-12 07:20:56 (GMT-4)
---------------------------------------------------------------------------
+ Server: nginx/1.21.6
+ /: The anti-clickjacking X-Frame-Options header is not present. See: https://developer.mozilla.org/
en-US/docs/Web/HTTP/Headers/X-Frame-Options
+ /: Uncommon header 'host-header' found, with contents: c2hhcmVkLmJsdWVob3N0LmNvbQ==.
+ /: Uncommon header 'x-server-cache' found, with contents: true.
+ /: Uncommon header 'x-proxy-cache' found, with contents: HIT.
+ /: The site uses TLS and the Strict-Transport-Security HTTP header is not defined. See: https://dev
eloper.mozilla.org/en-US/docs/Web/HTTP/Headers/Strict-Transport-Security
+ /: The X-Content-Type-Options header is not set. This could allow the user agent to render the cont
ent of the site in a different fashion to the MIME type. See: https://www.netsparker.com/web-vulnerab
ility-scanner/vulnerabilities/missing-content-type-header/
+ : Server banner changed from 'nginx/1.21.6' to 'Apache'.
```

Figure 5.10: Screenshot of Nikto

- **Qualys Vulnerability Management**

  Source: *https://www.qualys.com*

  Qualys VM is a cloud-based service that gives immediate, global visibility into where IT systems might be vulnerable to the latest Internet threats and how to protect them. It helps to continuously identify threats and monitor unexpected changes in a network before they turn into breaches.

  Features:

  - **Agent-based detection**

    Also works with the Qualys Cloud Agents, extending its network coverage to unscannable assets.

  - **Constant monitoring and alerts**

    When VM is paired with Continuous Monitoring (CM), InfoSec teams are proactively alerted about potential threats, so problems can be tackled before they turn into breaches.

  - **Comprehensive coverage and visibility**

    Continuously scans and identifies vulnerabilities for protecting IT assets on-premises, in the cloud, and at mobile endpoints. Its executive dashboard displays an overview of the security posture and gives access to remediation details. VM generates custom, role-based reports for multiple stakeholders, including automatic security documentation for compliance auditors.

  - **VM for the perimeter-less world**

    As enterprises adopt cloud computing, mobility, and other disruptive technologies for digital transformation, Qualys VM offers next-generation vulnerability management for these hybrid IT environments whose traditional boundaries have been blurred.

  - **Discover forgotten devices and organize the host assets**

    Qualys can help quickly determine what is running in different parts of the network—from the perimeter and corporate network to virtualized machines and cloud services. It can also identify unexpected access points, web servers, and other devices that can expose the network to attack.

  - **Scan for vulnerabilities everywhere, accurately and efficiently**

    Scan systems anywhere from the same console, including the perimeter, the internal network, and cloud environments.

  - **Identify and prioritize risks**

    Qualys, using trend analysis, Zero-Day, and Patch impact predictions, can identify the highest business risks.

- **Remediate vulnerabilities**

  Qualys's ability to track vulnerability data across hosts and time produces interactive reports that provide a better understanding of the security of the network.

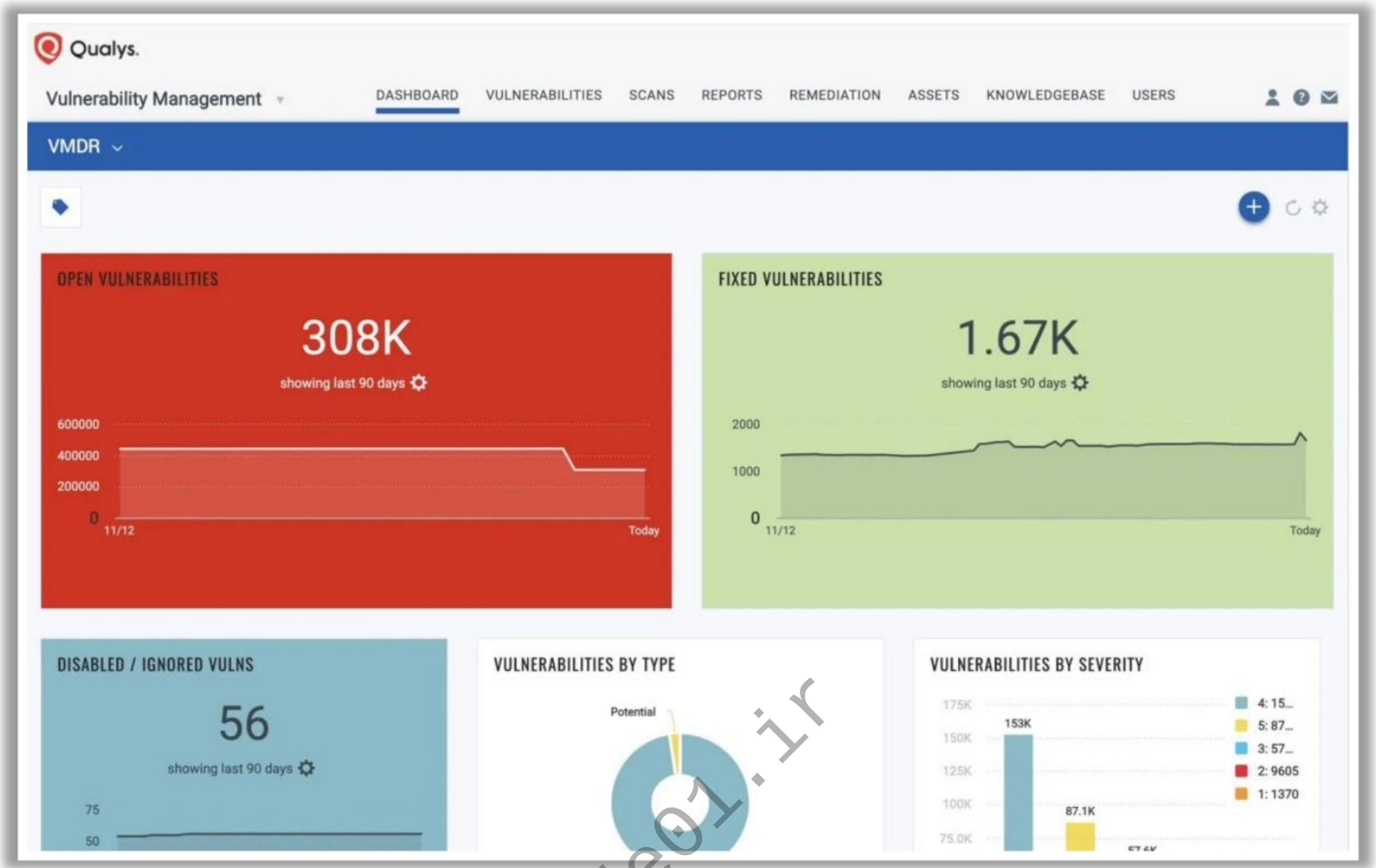


Figure 5.11: Vulnerability scanning using Qualys Vulnerability Management

Listed below are some of the additional vulnerability assessment tools:

- InsightVM (*https://www.rapid7.com*)
- Acunetix Web Vulnerability Scanner (*https://www.acunetix.com*)
- Nexpose (*https://www.rapid7.com*)
- Sniper (*https://sn1persecurity.com*)
- Tripwire IP360 (*https://www.tripwire.com*)
- SAINT Security Suite (*https://www.carson-saint.com*)
- BeSECURE (*https://www.beyondsecurity.com*)
- Core Impact Pro (*https://www.coresecurity.com*)
- Intruder (*https://www.intruder.io*)
- ManageEngine Vulnerability Manager Plus (*https://www.manageengine.com*)
- Astra Pentest (*https://www.getastra.com*)
- Skybox (*https://www.skyboxsecurity.com*)
- MaxPatrol TM (*https://www.ptsecurity.com*)

AI- Powered Vulnerability Assessment Tools

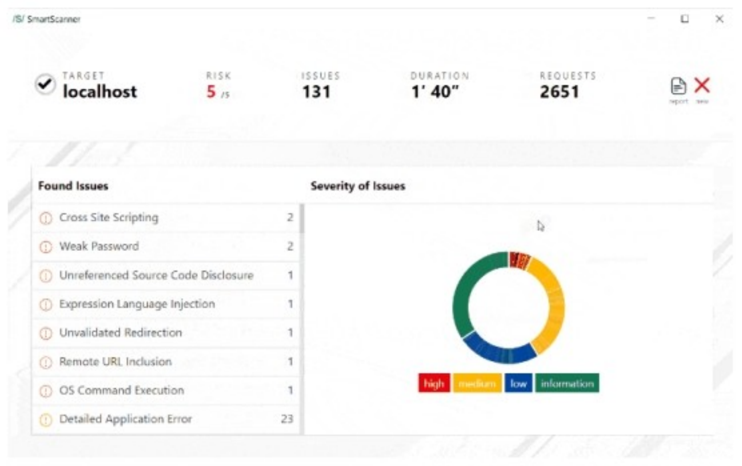
Equixly

- Equixly is a SaaS platform that integrates API security testing into the development workflow
- It leverages machine learning to **automate vulnerability assessments** and provide developers with actionable remediation plans

https://equixly.com

SmartScanner

- SmartScanner an automated vulnerability scanner designed to **identify and mitigate potential vulnerabilities in websites** utilizes AI and machine learning (ML) to enhance its threat detection capabilities



SmartScanner

TARGET	RISK	ISSUES	DURATION	REQUESTS
localhost	5 /5	131	1' 40"	2651

Found Issues	
Cross Site Scripting	2
Weak Password	2
Unreferenced Source Code Disclosure	1
Expression Language Injection	1
Unvalidated Redirection	1
Remote URL Inclusion	1
OS Command Execution	1
Detailed Application Error	23

https://www.thesmartscanner.com

Other Tools:

Tool	URL
CodeDefender	https://codedefender.ro
DryRun Security	*https://www.dryrun.security*
Hackules	https://hackules.com
Corgea	*https://corgea.com*
Pentest Copilot	*https://copilot.bugbase.ai*

AI-Powered Vulnerability Assessment Tools

Traditional vulnerability scanning tools often struggle to keep up with rapidly evolving cyber threats because of their reliance on predefined rules and signatures, leading to inefficient and error-prone processes. By contrast, AI-powered vulnerability assessments revolutionize security risk management by leveraging advanced technologies to automate and enhance vulnerability detection and remediation processes. AI-driven scanners can adapt to new threats, reduce false positives, provide more accurate and actionable insights, empower ethical hackers and security teams to address vulnerabilities proactively, and strengthen an organization's overall cybersecurity posture.

By contrast, AI-powered vulnerability scanners can continuously learn from new data, including emerging threats and attack technique patterns. This allows them to adapt and improve their detection capabilities over time. By leveraging machine-learning algorithms, these scanners can identify patterns, anomalies, and potential vulnerabilities more effectively than traditional tools.

Furthermore, AI-powered scanners can adapt to the specific needs and requirements of an organization by tailoring their scanning strategies and detection methods to a unique environment. This flexibility allows for more accurate and targeted vulnerability assessments, thereby reducing the number of false positives and negatives.

Comparison Point	Traditional Vulnerability Assessment	AI-Powered Vulnerability Assessment
Scope and Coverage	Focuses on known vulnerabilities based on predefined rules and signatures	Analyzes vast amounts of data to identify patterns and detect both known and unknown vulnerabilities
Prioritization	Prioritizes vulnerabilities based solely on technical severity without considering business context	Utilizes risk-based prioritization, considering factors such as asset criticality, compliance requirements, and likelihood of exploitation
Efficiency and Scalability	Time-consuming and resource-intensive, especially in large and complex environments	Leverages automation and algorithms to streamline the vulnerability assessment process
Adaptability	Relies on predefined rules and signatures, making it difficult to adapt to new threats and attack techniques	Continuously learns from new data, including emerging threats and attack methods

Table 5.3: Comparison between traditional VA & AI-powered VA

AI-Powered Vulnerability Assessment Tool: Equixly

Source: *https://equixly.com*

Equixly is an advanced AI-powered tool designed specifically for vulnerability assessment with a focus on securing APIs. It uses AI and ML to identify and eliminate blind spots, thereby ensuring robust protection against potential threats.

Key Features of Equixly for vulnerability management are as follows:

- **AI-Driven Vulnerability Detection**

  Equixly uses machine-learning algorithms to scan and identify vulnerabilities within APIs, ensuring that no potential threats are overlooked.

- **Automated Threat Analysis**

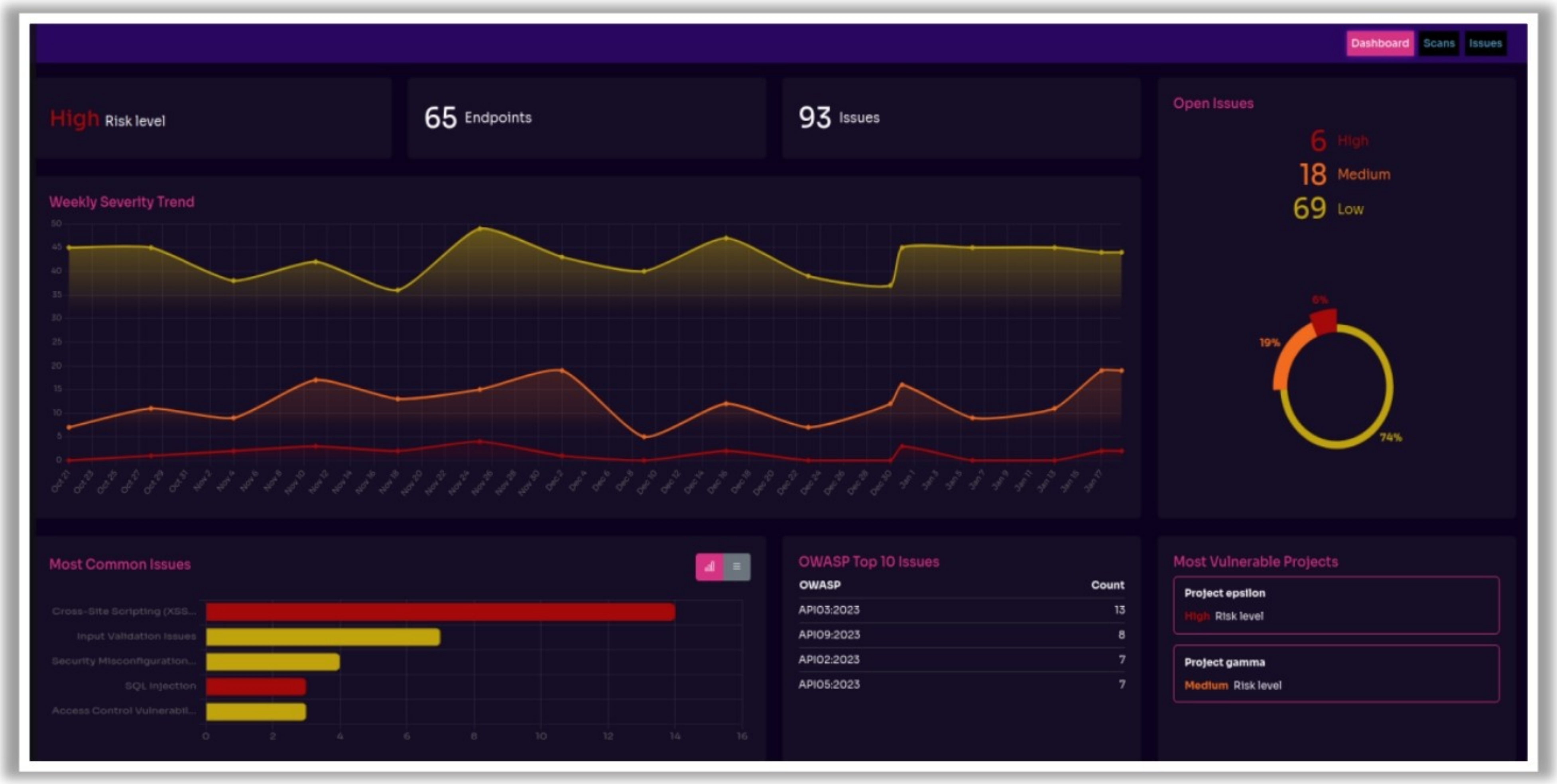
  This tool automates the process of analyzing threat data, enabling quicker identification and response to emerging security risks.

- **Real-Time Security Monitoring**

  It provides continuous monitoring of API environments, and offers real-time updates and alerts regarding potential vulnerabilities.

- **Adaptive Learning**

  Machine-learning models continuously learn from new data, improving the accuracy and efficiency of vulnerability detection over time.



OWASP Top 10 Issues

OWASP	Count
API03:2023	13
API09:2023	8
API02:2023	7
API05:2023	7

Figure 5.12: Screenshot of Equixly

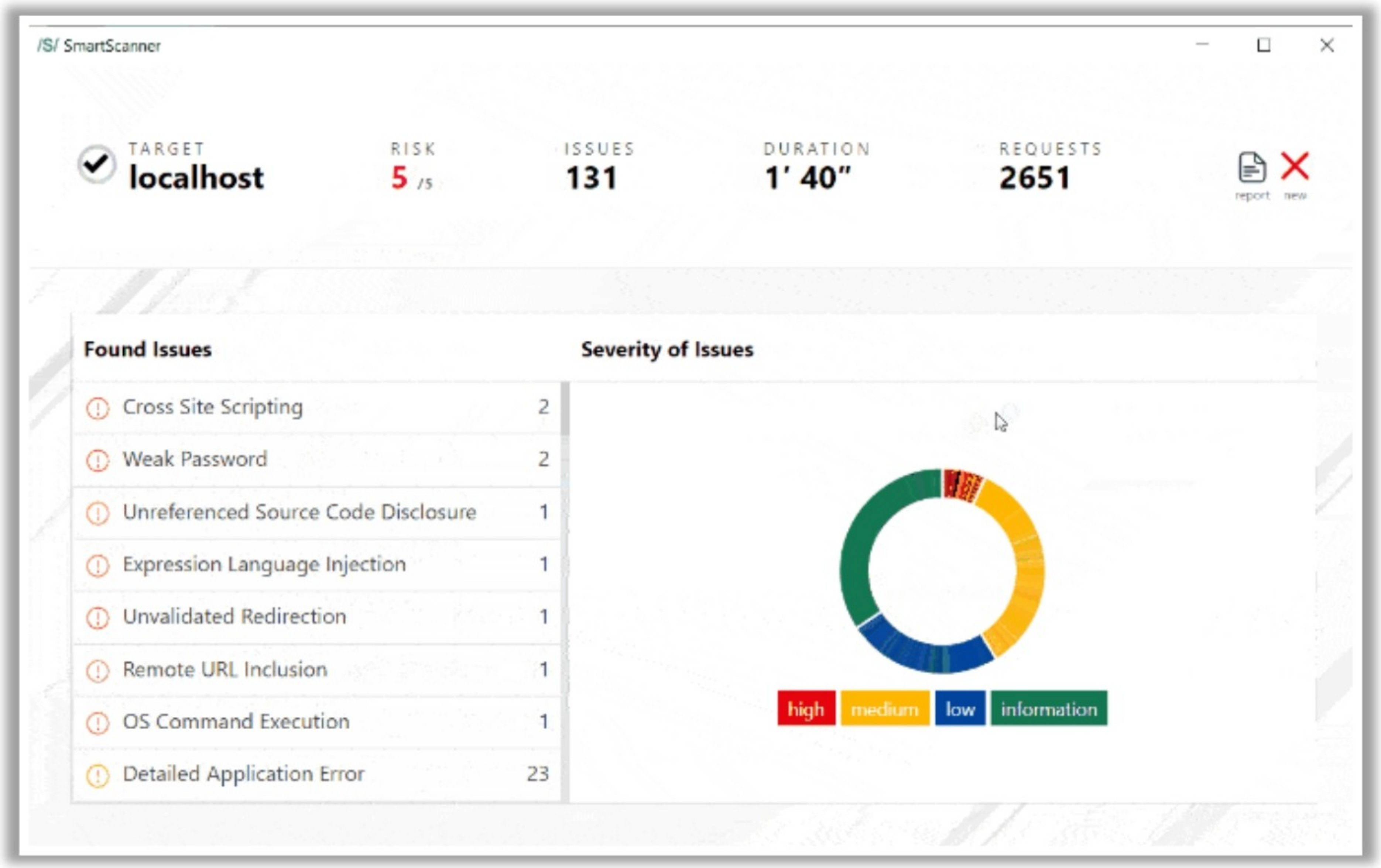
AI-Powered Automated Vulnerability Scanner: SmartScanner

Source: *https://www.thesmartscanner.com*

SmartScanner is an AI-powered automated vulnerability scanner designed to enhance website security. Advanced ML algorithms are used to monitor websites continuously for potential vulnerabilities and threats.

The key features of SmartScanner include:

- **Supervised and Unsupervised ML:** SmartScanner analyzes vast amounts of data using both supervised and unsupervised ML algorithms. This allows it to learn the patterns of benign and malicious activities, allowing it to distinguish between them.
- **Baseline Establishment:** AI models in SmartScanner establish baselines of normal behavior for each website it monitors. These baselines were then used to identify deviations that may indicate potential threats.
- **Anomaly Detection:** SmartScanner employs anomaly detection algorithms to flag activities that deviate from established baselines. This helps to identify and alert suspicious behaviors in real time.
- **Real-time Analytics and Response:** The AI-driven systems in SmartScanner provide real-time analytics of the websites it monitors. It can automatically respond by quickly mitigating the identified threats, thereby reducing the risk of successful attacks.



Found Issues	
Cross Site Scripting	2
Weak Password	2
Unreferenced Source Code Disclosure	1
Expression Language Injection	1
Unvalidated Redirection	1
Remote URL Inclusion	1
OS Command Execution	1
Detailed Application Error	23

Figure 5.13: Screenshot of SmartScanner

Additional AI-powered Vulnerability Assessment Tools

- **CodeDefender**

  Source: *https://codedefender.ro*

  CodeDefender is an AI-powered vulnerability assessment tool that helps organizations automatically detect, prioritize, and fix security vulnerabilities in their code bases. It integrates existing security tools to provide a comprehensive vulnerability-management solution.

- **Corgea**

  Source: *https://corgea.com*

  Corgea is an AI-powered platform that automatically generates and deploys security fixes for vulnerabilities detected in software code. It leverages machine-learning models to analyze vulnerability data and write secure code patches, thereby reducing the manual effort required by security teams.

- **Fluxguard**

  Source: *https://fluxguard.com*

  Fluxguard employs AI algorithms to automatically scan and detect vulnerabilities across diverse IT infrastructures, including networks, applications, and systems. It utilizes ML to conduct a behavioral analysis of network traffic and system interactions and identifies anomalous behaviors that could indicate potential vulnerabilities or attacks.

- **DryRun Security**

  Source: *https://www.dryrun.security*

  DryRun Security is a vulnerability assessment and penetration-testing platform that uses AI and automation to identify and validate security weaknesses in web applications and infrastructure.

- **Pentest Copilot**

  Source: *https://copilot.bugbase.ai*

  The Pentest Copilot is an AI-powered penetration-testing assistant that helps security teams conduct more efficient and effective vulnerability assessments. It automates various penetration-testing tasks, from reconnaissance to exploitation, and provides actionable insights for prioritizing and remediating identified vulnerabilities.

- **Beagle Security**

  Source: *https://beaglesecurity.com*

  Beagle Security is a comprehensive web application security testing platform that combines automated scanning and manual penetration testing. It uses AI and ML to detect a wide range of vulnerabilities, including the top 10 OWASP risks, and provides detailed reports to help organizations improve their application security.

- **Hackules**

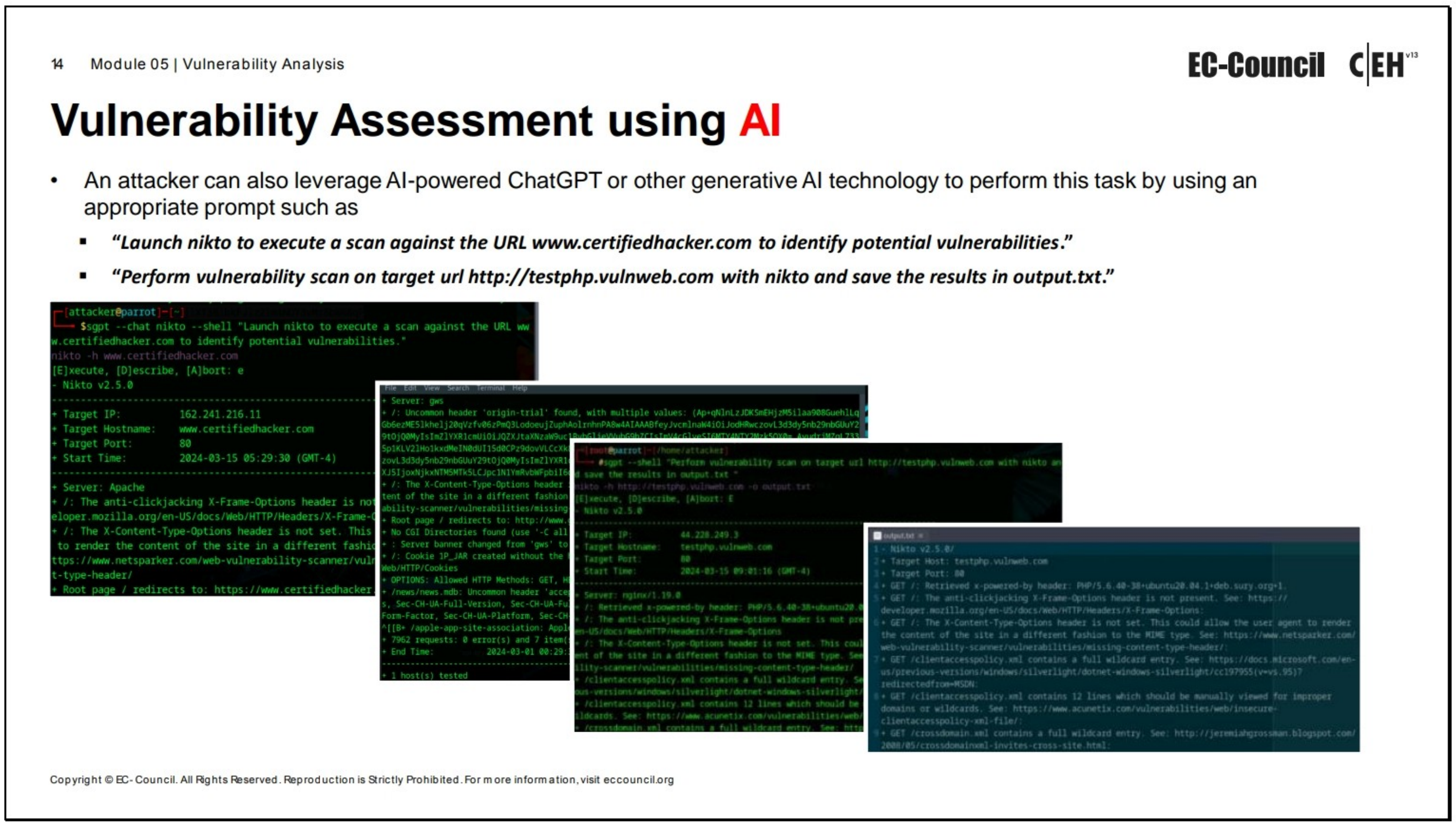
  Source: *https://hackules.com*

  Hackules is an AI-powered vulnerability assessment and penetration-testing platform that helps organizations identify and mitigate security weaknesses in their web applications and infrastructures. It uses advanced techniques such as NLP and ML to provide accurate and actionable security insights.

- **Coderbuds**

  Source: *https://coderbuds.com*

  CoderBuds are AI-driven code security platforms that help developers and security teams detect, prioritize, and fix vulnerabilities in their codebases. Its AI algorithm is integrated seamlessly with mainstream development tools, and CoderBuds conducts automated vulnerability scans, performs comprehensive risk assessments, and offers tailored remediation recommendations.



14 Module 05 | Vulnerability Analysis

EC-Council C|EH v13

Vulnerability Assessment using AI

- An attacker can also leverage AI-powered ChatGPT or other generative AI technology to perform this task by using an appropriate prompt such as
  - ***"Launch nikto to execute a scan against the URL www.certifiedhacker.com to identify potential vulnerabilities."***
  - ***"Perform vulnerability scan on target url http://testphp.vulnweb.com with nikto and save the results in output.txt."***

Copyright © EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited. For more information, visit eccouncil.org

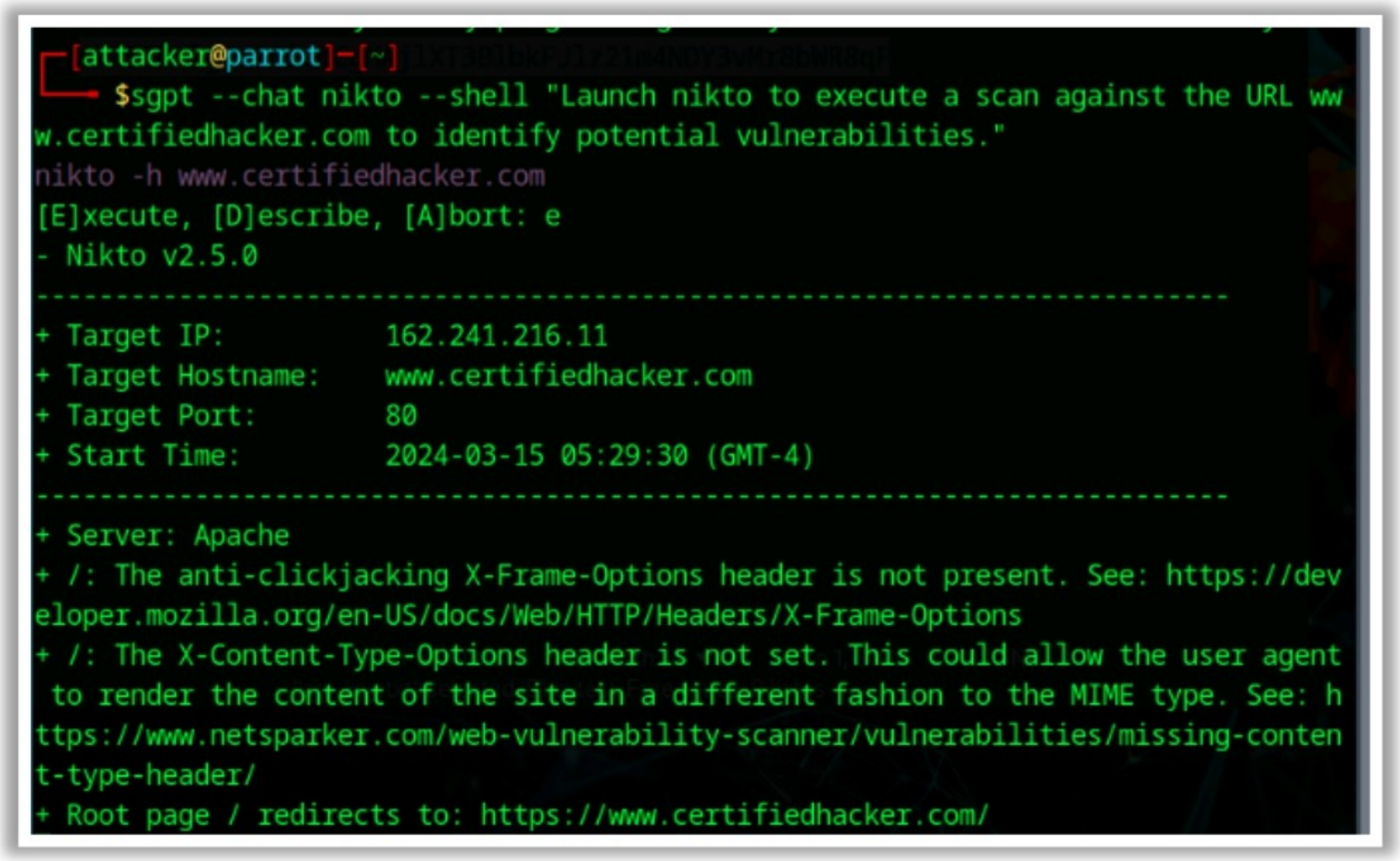
Vulnerability Assessment using AI

Attackers can leverage AI-powered technologies to enhance and automate their vulnerability scanning tasks. With the aid of AI, attackers can effortlessly perform vulnerability scanning to identify the potential vulnerabilities on target.

An attacker can use ChatGPT to perform this task by using an appropriate prompt such as:

Example #1:

"Launch nikto to execute a scan against the URL www.certifiedhacker.com to identify potential vulnerabilities."



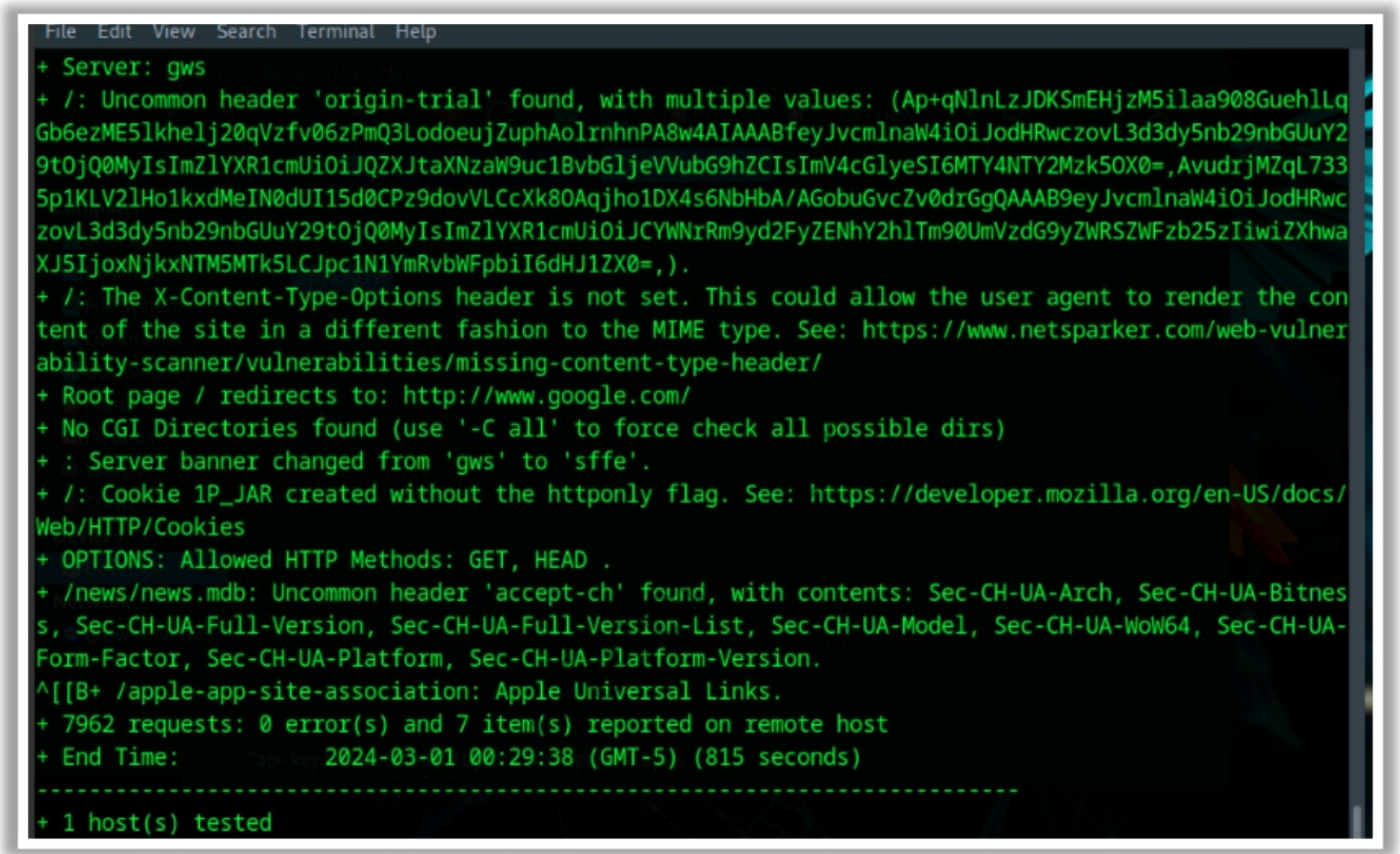
```
┌─[attacker@parrot]─[~]
└──╼ $sgpt --chat nikto --shell "Launch nikto to execute a scan against the URL ww
w.certifiedhacker.com to identify potential vulnerabilities."
nikto -h www.certifiedhacker.com
[E]xecute, [D]escribe, [A]bort: e
- Nikto v2.5.0
---------------------------------------------------------------------------
+ Target IP:          162.241.216.11
+ Target Hostname:    www.certifiedhacker.com
+ Target Port:        80
+ Start Time:         2024-03-15 05:29:30 (GMT-4)
---------------------------------------------------------------------------
+ Server: Apache
+ /: The anti-clickjacking X-Frame-Options header is not present. See: https://dev
eloper.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options
+ /: The X-Content-Type-Options header is not set. This could allow the user agent
 to render the content of the site in a different fashion to the MIME type. See: h
ttps://www.netsparker.com/web-vulnerability-scanner/vulnerabilities/missing-conten
t-type-header/
+ Root page / redirects to: https://www.certifiedhacker.com/
```

Figure 5.14: Launch nikto to execute a scan against the URL

The command scans the URL www.certifiedhacker.com for potential vulnerabilities using the Nikto web server scanner.

```
nikto -h www.certifiedhacker.com
```

- **`nikto`**: This command invokes Nikto, a web server scanner that performs comprehensive tests against web servers for potential vulnerabilities.
- **`-h www.certifiedhacker.com`**: This option specifies the target URL (www.certifiedhacker.com) to scan for vulnerabilities. Nikto will perform various checks and tests against the specified URL to identify potential security issues and vulnerabilities.



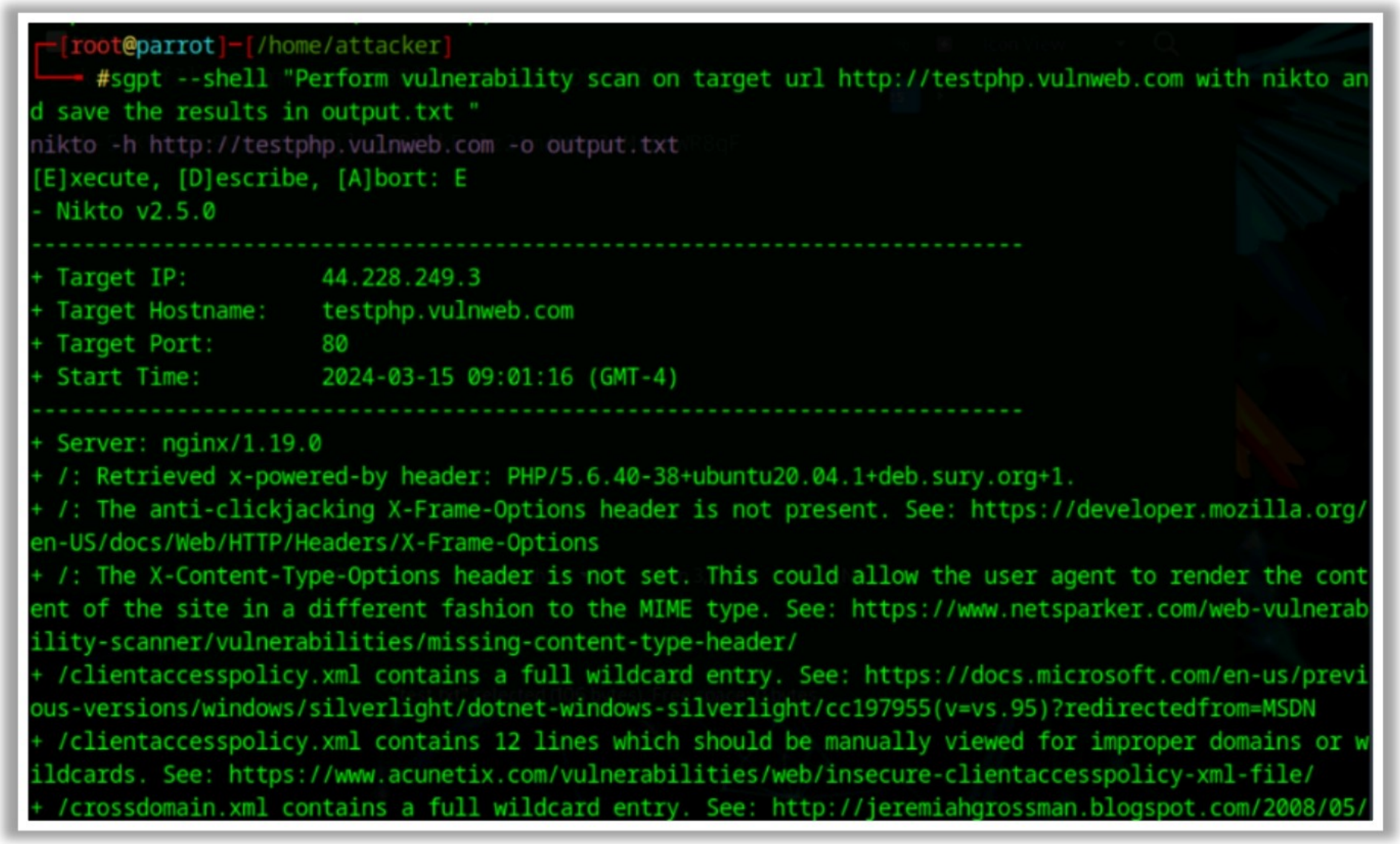
```
+ Server: gws
+ /: Uncommon header 'origin-trial' found, with multiple values: (Ap+qNlnLzJDKSmEHjzM5ilaa908GuehlLq
Gb6ezME5lkhelj20qVzfv06zPmQ3LodoeujZuphAolrnhnPA8w4AIAAABfeyJvcmlnaW4iOiJodHRwczovL3d3dy5nb29nbGUuY2
9tOjQ0MyIsImZlYXR1cmUiOiJQZXJtaXNzaW9uc1BvbGljeVVubG9hZCIsImV4cGlyeSI6MTY4NTY2Mzk5OX0=,AvudrjMZqL733
5p1KLV2lHo1kxdMeIN0dUI15d0CPz9dovVLCcXk8OAqjho1DX4s6NbHbA/AGobuGvcZv0drGgQAAAB9eyJvcmlnaW4iOiJodHRwc
zovL3d3dy5nb29nbGUuY29tOjQ0MyIsImZlYXR1cmUiOiJCYWNrRm9yd2FyZENhY2hlTm90UmVzdG9yZWRSZWFzb25zIiwiZXhwa
XJ5IjoxNjkxNTM5MTk5LCJpc1N1YmRvbWFpbiI6dHJ1ZX0=,).
+ /: The X-Content-Type-Options header is not set. This could allow the user agent to render the con
tent of the site in a different fashion to the MIME type. See: https://www.netsparker.com/web-vulner
ability-scanner/vulnerabilities/missing-content-type-header/
+ Root page / redirects to: http://www.google.com/
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ : Server banner changed from 'gws' to 'sffe'.
+ /: Cookie 1P_JAR created without the httponly flag. See: https://developer.mozilla.org/en-US/docs/
Web/HTTP/Cookies
+ OPTIONS: Allowed HTTP Methods: GET, HEAD .
+ /news/news.mdb: Uncommon header 'accept-ch' found, with contents: Sec-CH-UA-Arch, Sec-CH-UA-Bitnes
s, Sec-CH-UA-Full-Version, Sec-CH-UA-Full-Version-List, Sec-CH-UA-Model, Sec-CH-UA-WoW64, Sec-CH-UA-
Form-Factor, Sec-CH-UA-Platform, Sec-CH-UA-Platform-Version.
^[[B+ /apple-app-site-association: Apple Universal Links.
+ 7962 requests: 0 error(s) and 7 item(s) reported on remote host
+ End Time:           2024-03-01 00:29:38 (GMT-5) (815 seconds)
---------------------------------------------------------------------------
+ 1 host(s) tested
```

Figure 5.15: Nikto performing various checks and tests against the specified URL to identify potential security issues and vulnerabilities

Example #2:

"Perform vulnerability scan on target url http://testphp.vulnweb.com with nikto and save the results in output.txt."

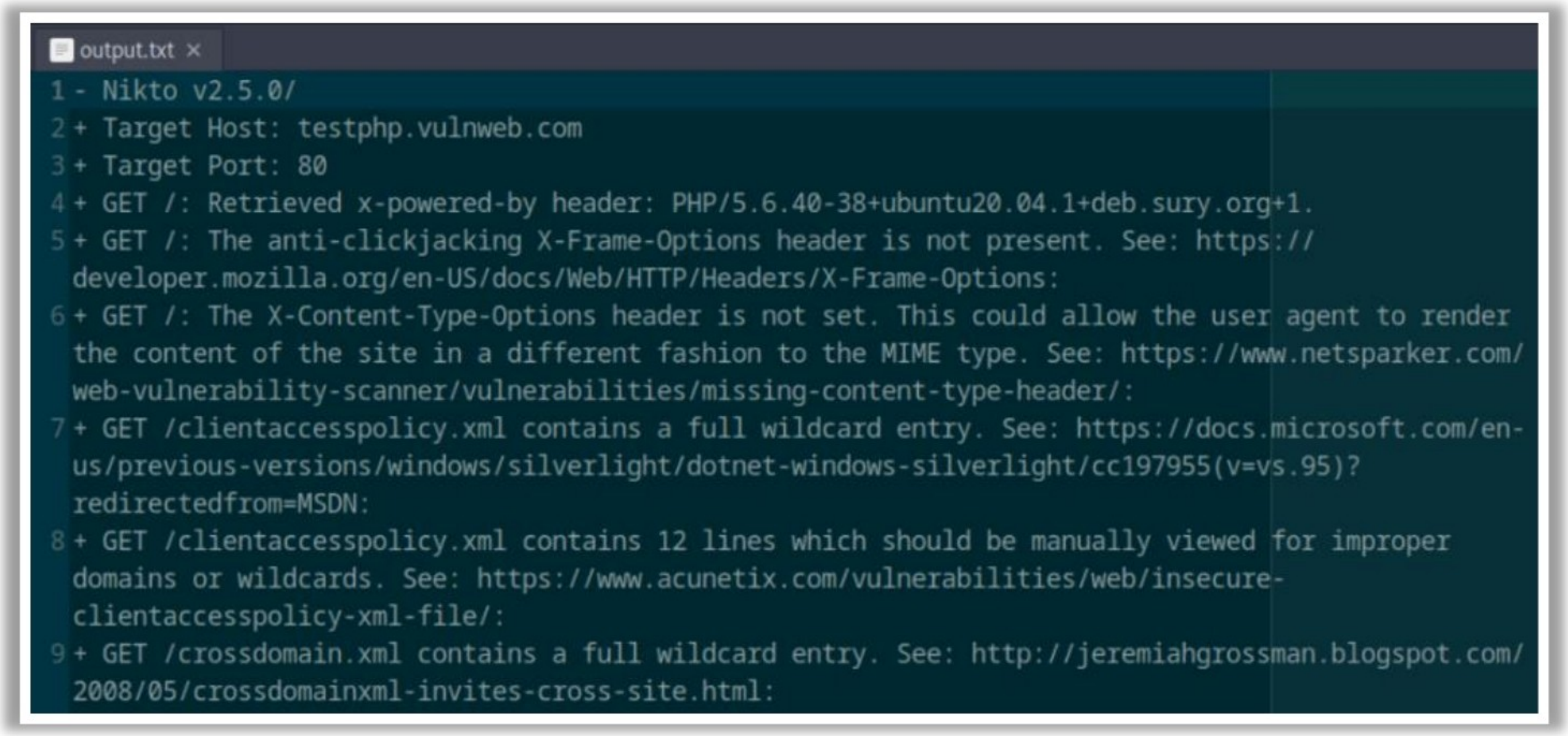
```
nikto -h http://testphp.vulnweb.com -o output.txt
```



```
┌─[root@parrot]─[/home/attacker]
└──╼ #sgpt --shell "Perform vulnerability scan on target url http://testphp.vulnweb.com with nikto an
d save the results in output.txt "
nikto -h http://testphp.vulnweb.com -o output.txt
[E]xecute, [D]escribe, [A]bort: E
- Nikto v2.5.0
---------------------------------------------------------------------------
+ Target IP:          44.228.249.3
+ Target Hostname:    testphp.vulnweb.com
+ Target Port:        80
+ Start Time:         2024-03-15 09:01:16 (GMT-4)
---------------------------------------------------------------------------
+ Server: nginx/1.19.0
+ /: Retrieved x-powered-by header: PHP/5.6.40-38+ubuntu20.04.1+deb.sury.org+1.
+ /: The anti-clickjacking X-Frame-Options header is not present. See: https://developer.mozilla.org/
en-US/docs/Web/HTTP/Headers/X-Frame-Options
+ /: The X-Content-Type-Options header is not set. This could allow the user agent to render the cont
ent of the site in a different fashion to the MIME type. See: https://www.netsparker.com/web-vulnerab
ility-scanner/vulnerabilities/missing-content-type-header/
+ /clientaccesspolicy.xml contains a full wildcard entry. See: https://docs.microsoft.com/en-us/previ
ous-versions/windows/silverlight/dotnet-windows-silverlight/cc197955(v=vs.95)?redirectedfrom=MSDN
+ /clientaccesspolicy.xml contains 12 lines which should be manually viewed for improper domains or w
ildcards. See: https://www.acunetix.com/vulnerabilities/web/insecure-clientaccesspolicy-xml-file/
+ /crossdomain.xml contains a full wildcard entry. See: http://jeremiahgrossman.blogspot.com/2008/05/
```

Figure 5.16: Nikto for performing comprehensive tests against web servers to identify potential vulnerabilities.

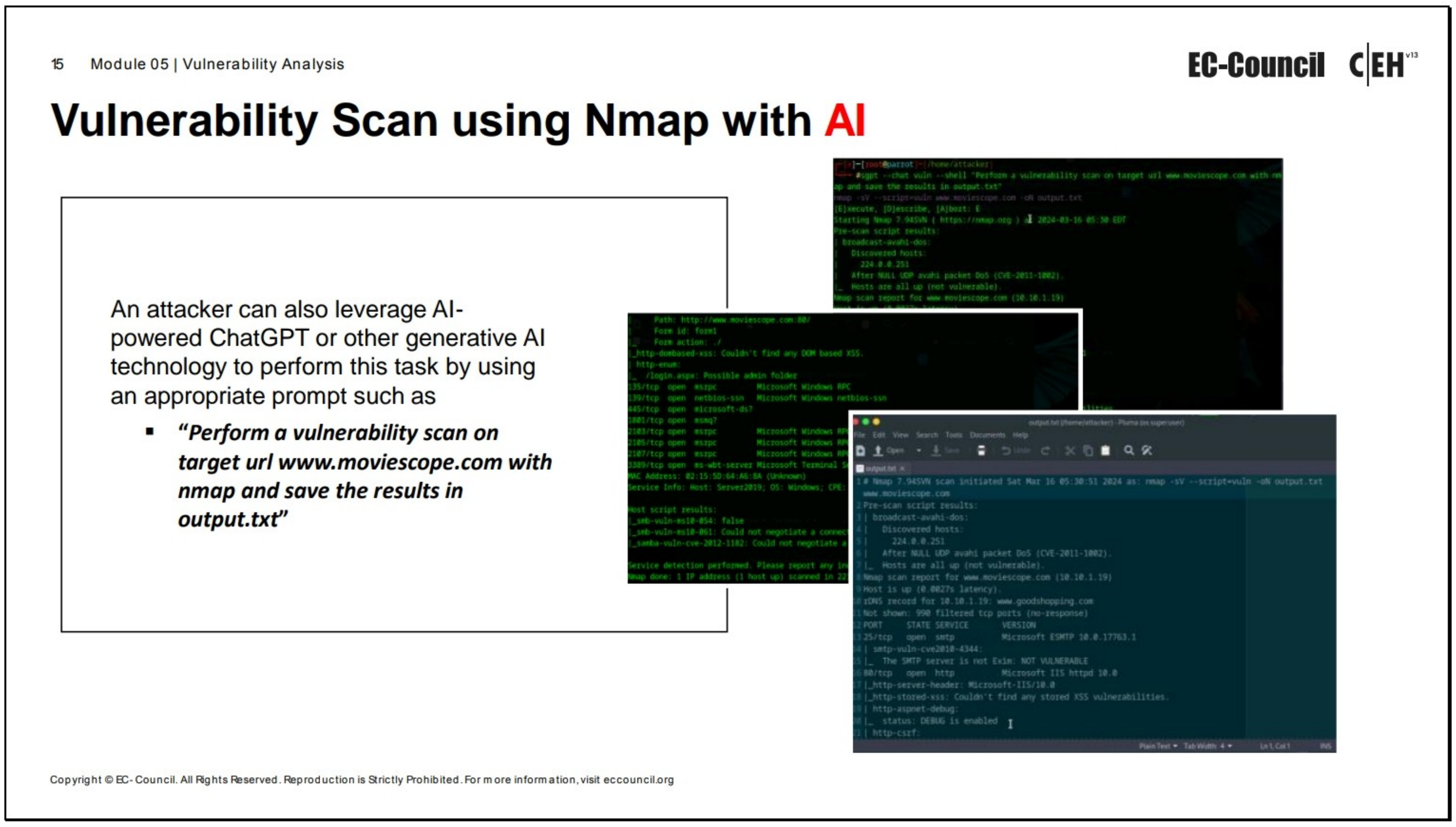
- `` `nikto` ``: This command invokes Nikto.
- `` `-h http://testphp.vulnweb.com` ``: This option specifies the target URL (http://testphp.vulnweb.com) to scan for vulnerabilities.
- `` `-o output.txt` ``: This option specifies the file where the scan results will be saved. In this case, the results will be saved in a file named "output.txt".



output.txt

```
- Nikto v2.5.0/
+ Target Host: testphp.vulnweb.com
+ Target Port: 80
+ GET /: Retrieved x-powered-by header: PHP/5.6.40-38+ubuntu20.04.1+deb.sury.org+1.
+ GET /: The anti-clickjacking X-Frame-Options header is not present. See: https://
developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options:
+ GET /: The X-Content-Type-Options header is not set. This could allow the user agent to render
the content of the site in a different fashion to the MIME type. See: https://www.netsparker.com/
web-vulnerability-scanner/vulnerabilities/missing-content-type-header/:
+ GET /clientaccesspolicy.xml contains a full wildcard entry. See: https://docs.microsoft.com/en-
us/previous-versions/windows/silverlight/dotnet-windows-silverlight/cc197955(v=vs.95)?
redirectedfrom=MSDN:
+ GET /clientaccesspolicy.xml contains 12 lines which should be manually viewed for improper
domains or wildcards. See: https://www.acunetix.com/vulnerabilities/web/insecure-
clientaccesspolicy-xml-file/:
+ GET /crossdomain.xml contains a full wildcard entry. See: http://jeremiahgrossman.blogspot.com/
2008/05/crossdomainxml-invites-cross-site.html:
```

Figure 5.17: File where the scan results will be saved output.txt



Vulnerability Scan using Nmap with AI

An attacker can also leverage AI-powered ChatGPT or other generative AI technology to perform this task by using an appropriate prompt such as

- ***"Perform a vulnerability scan on target url www.moviescope.com with nmap and save the results in output.txt"***

```
┌─[✗]─[root@parrot]─[/home/attacker]
└──╼ #sgpt --chat vuln --shell "Perform a vulnerability scan on target url www.moviescope.com with nm
ap and save the results in output.txt"
nmap -sV --script=vuln www.moviescope.com -oN output.txt
[E]xecute, [D]escribe, [A]bort: E
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-03-16 05:30 EDT
Pre-scan script results:
| broadcast-avahi-dos:
|   Discovered hosts:
|     224.0.0.251
|   After NULL UDP avahi packet DoS (CVE-2011-1002).
|_  Hosts are all up (not vulnerable).
Nmap scan report for www.moviescope.com (10.10.1.19)
```

```
|     Path: http://www.moviescope.com:80/
|     Form id: form1
|_    Form action: ./
|_http-dombased-xss: Couldn't find any DOM based XSS.
| http-enum:
|_  /login.aspx: Possible admin folder
135/tcp  open  msrpc         Microsoft Windows RPC
139/tcp  open  netbios-ssn   Microsoft Windows netbios-ssn
445/tcp  open  microsoft-ds?
1801/tcp open  msmq?
2103/tcp open  msrpc         Microsoft Windows RP
2105/tcp open  msrpc         Microsoft Windows RP
2107/tcp open  msrpc         Microsoft Windows RP
3389/tcp open  ms-wbt-server Microsoft Terminal S
MAC Address: 02:15:5D:64:A6:8A (Unknown)
Service Info: Host: Server2019; OS: Windows; CPE:

Host script results:
|_smb-vuln-ms10-054: false
|_smb-vuln-ms10-061: Could not negotiate a connect
|_samba-vuln-cve-2012-1182: Could not negotiate a

Service detection performed. Please report any in
Nmap done: 1 IP address (1 host up) scanned in 22
```

output.txt (/home/attacker) - Pluma (as superuser)

```
# Nmap 7.94SVN scan initiated Sat Mar 16 05:30:51 2024 as: nmap -sV --script=vuln -oN output.txt
www.moviescope.com
Pre-scan script results:
| broadcast-avahi-dos:
|   Discovered hosts:
|     224.0.0.251
|   After NULL UDP avahi packet DoS (CVE-2011-1002).
|_  Hosts are all up (not vulnerable).
Nmap scan report for www.moviescope.com (10.10.1.19)
Host is up (0.0027s latency).
rDNS record for 10.10.1.19: www.goodshopping.com
Not shown: 990 filtered tcp ports (no-response)
PORT      STATE SERVICE       VERSION
25/tcp    open  smtp          Microsoft ESMTP 10.0.17763.1
| smtp-vuln-cve2010-4344:
|_  The SMTP server is not Exim: NOT VULNERABLE
80/tcp    open  http          Microsoft IIS httpd 10.0
|_http-server-header: Microsoft-IIS/10.0
|_http-stored-xss: Couldn't find any stored XSS vulnerabilities.
| http-aspnet-debug:
|_  status: DEBUG is enabled
| http-csrf:
```

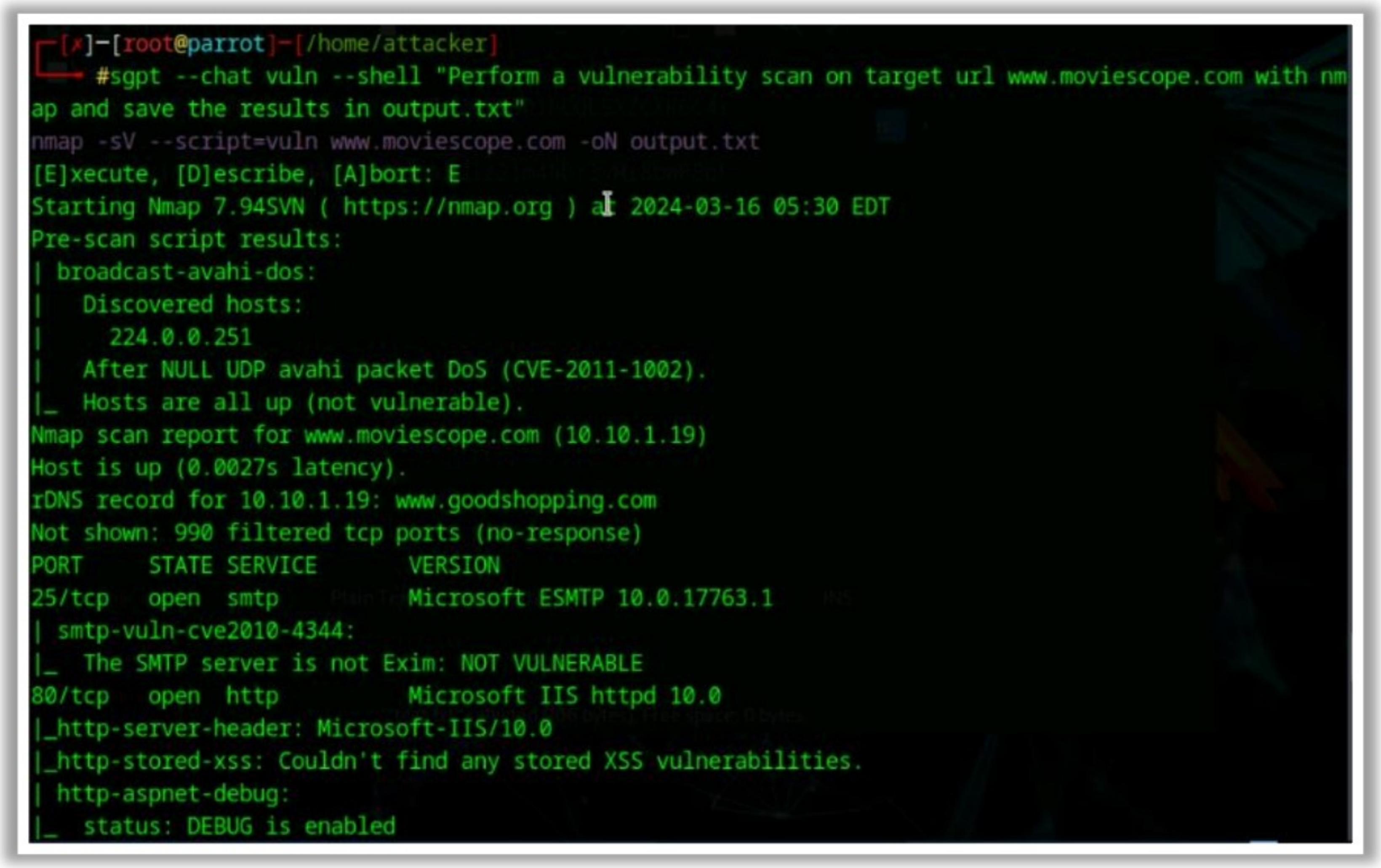
Vulnerability Scan using Nmap with AI

Attackers can leverage AI-powered technologies to enhance and automate their vulnerability scanning tasks. With the aid of AI, attackers can effortlessly perform vulnerability scanning using Nmap to identify the potential vulnerabilities on target.

For example,

An attacker can use ChatGPT to perform this task by using an appropriate prompt such as:

"Perform a vulnerability scan on target url www.moviescope.com with nmap and save the results in output.txt"

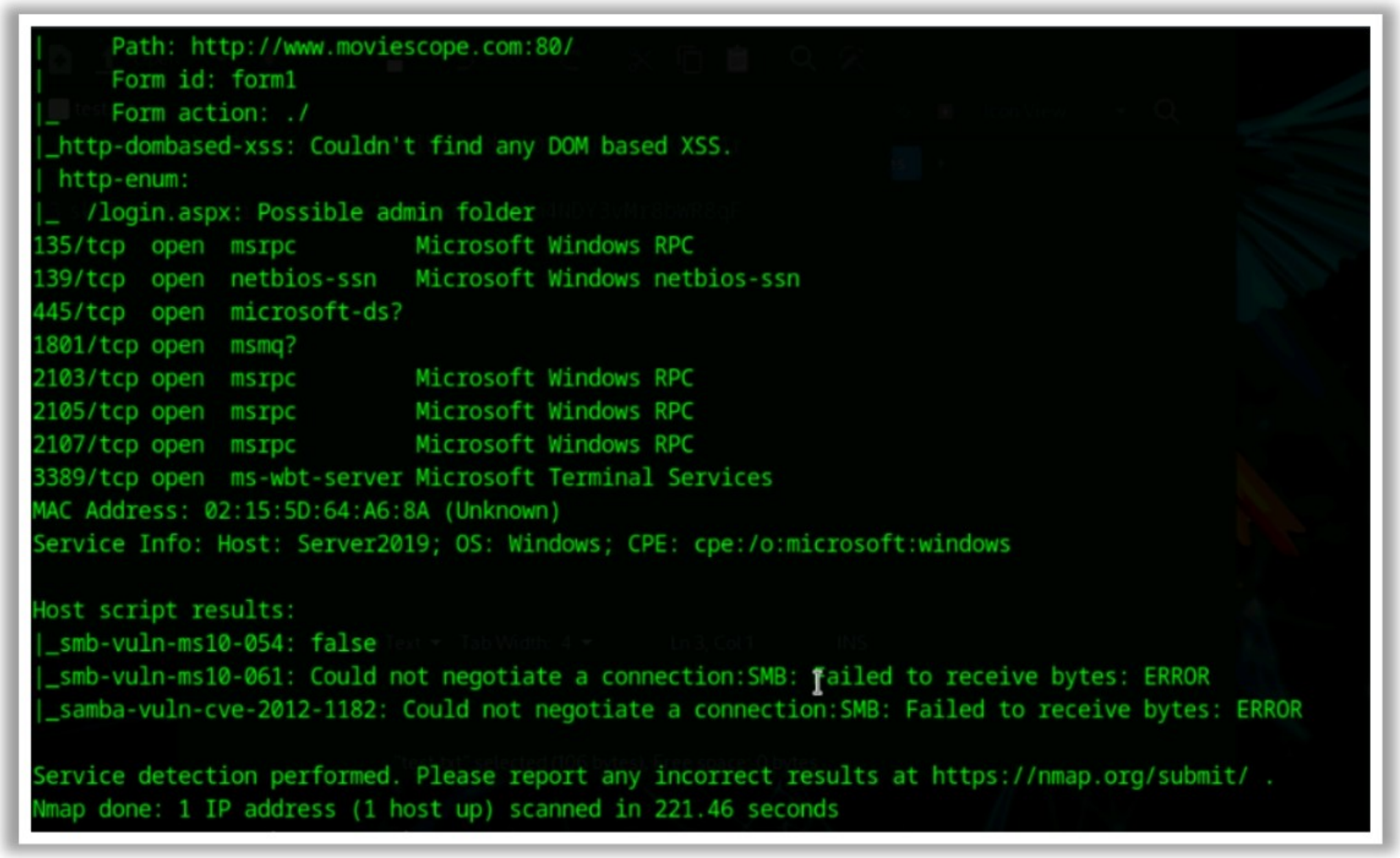


```
┌─[✗]─[root@parrot]─[/home/attacker]
└──╼ #sgpt --chat vuln --shell "Perform a vulnerability scan on target url www.moviescope.com with nm
ap and save the results in output.txt"
nmap -sV --script=vuln www.moviescope.com -oN output.txt
[E]xecute, [D]escribe, [A]bort: E
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-03-16 05:30 EDT
Pre-scan script results:
| broadcast-avahi-dos:
|   Discovered hosts:
|     224.0.0.251
|   After NULL UDP avahi packet DoS (CVE-2011-1002).
|_  Hosts are all up (not vulnerable).
Nmap scan report for www.moviescope.com (10.10.1.19)
Host is up (0.0027s latency).
rDNS record for 10.10.1.19: www.goodshopping.com
Not shown: 990 filtered tcp ports (no-response)
PORT     STATE SERVICE       VERSION
25/tcp   open  smtp          Microsoft ESMTP 10.0.17763.1
| smtp-vuln-cve2010-4344:
|_  The SMTP server is not Exim: NOT VULNERABLE
80/tcp   open  http          Microsoft IIS httpd 10.0
|_http-server-header: Microsoft-IIS/10.0
|_http-stored-xss: Couldn't find any stored XSS vulnerabilities.
| http-aspnet-debug:
|_  status: DEBUG is enabled
```

Figure 5.18: Perform a vulnerability scan on target URL

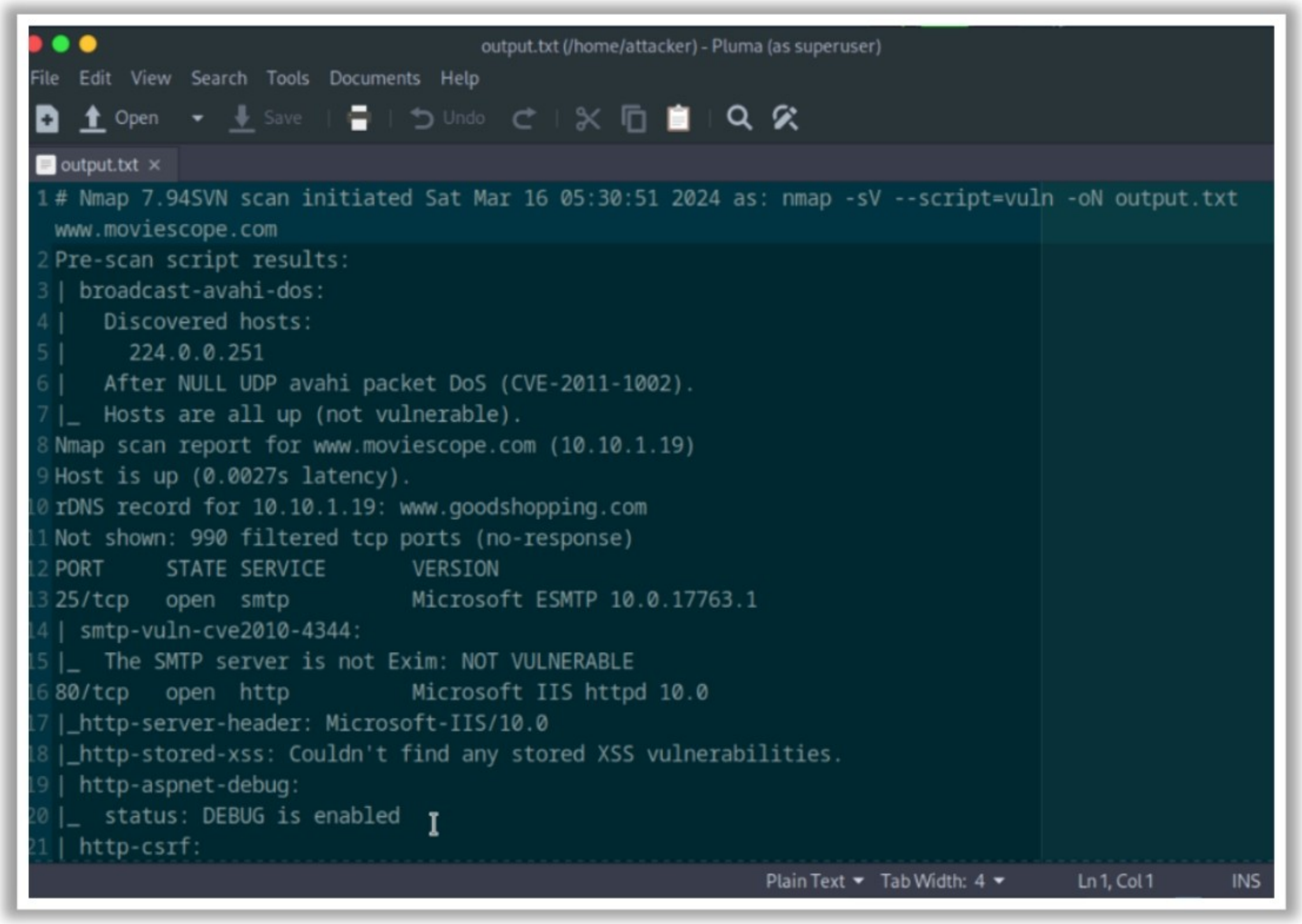
`nmap -sV -script=vuln www.moviescope.com -oN output.txt`

- **`` `nmap` ``**: This command invokes Nmap.
- **`` `--script=vuln` ``**: This option specifies the Nmap script to run, which focuses on vulnerability scanning.
- **`` `www.moviescope.com` ``**: This is the target URL where the vulnerability scan will be performed.
- **`` `-oN output.txt` ``**: This option specifies the file where the scan results will be saved. In this case, the results will be saved in a file named "output.txt".



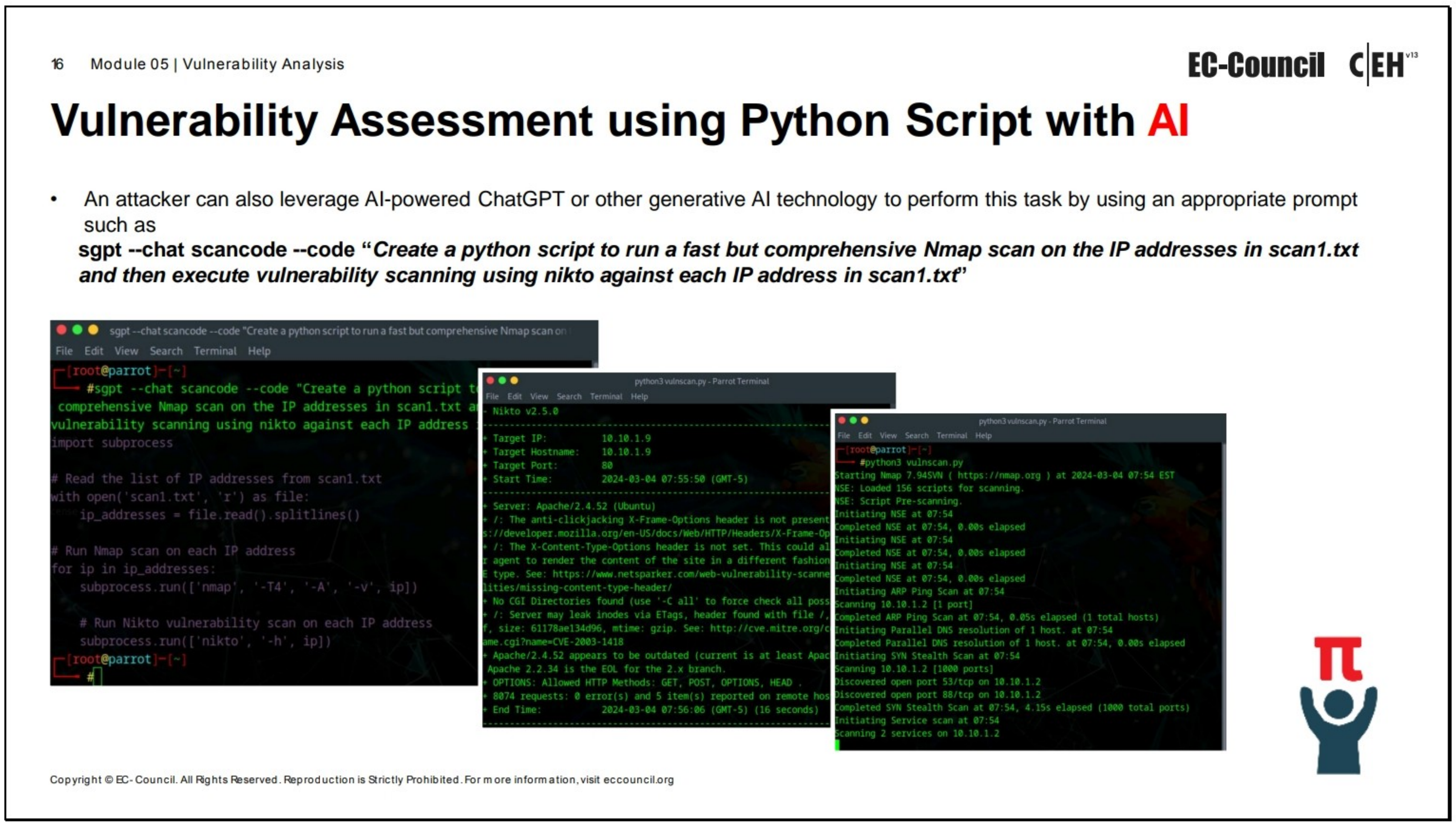
```
|     Path: http://www.moviescope.com:80/
|     Form id: form1
|_    Form action: ./
|_http-dombased-xss: Couldn't find any DOM based XSS.
| http-enum:
|_  /login.aspx: Possible admin folder
135/tcp  open  msrpc         Microsoft Windows RPC
139/tcp  open  netbios-ssn   Microsoft Windows netbios-ssn
445/tcp  open  microsoft-ds?
1801/tcp open  msmq?
2103/tcp open  msrpc         Microsoft Windows RPC
2105/tcp open  msrpc         Microsoft Windows RPC
2107/tcp open  msrpc         Microsoft Windows RPC
3389/tcp open  ms-wbt-server Microsoft Terminal Services
MAC Address: 02:15:5D:64:A6:8A (Unknown)
Service Info: Host: Server2019; OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
|_smb-vuln-ms10-054: false
|_smb-vuln-ms10-061: Could not negotiate a connection:SMB: Failed to receive bytes: ERROR
|_samba-vuln-cve-2012-1182: Could not negotiate a connection:SMB: Failed to receive bytes: ERROR

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 221.46 seconds
```

Figure 5.19: Output


output.txt (/home/attacker) - Pluma (as superuser)
File Edit View Search Tools Documents Help
Open
Save
Undo
output.txt
Plain Text
Tab Width: 4
Ln 1, Col 1
INS

```
# Nmap 7.94SVN scan initiated Sat Mar 16 05:30:51 2024 as: nmap -sV --script=vuln -oN output.txt
www.moviescope.com
Pre-scan script results:
| broadcast-avahi-dos:
|   Discovered hosts:
|     224.0.0.251
|   After NULL UDP avahi packet DoS (CVE-2011-1002).
|_  Hosts are all up (not vulnerable).
Nmap scan report for www.moviescope.com (10.10.1.19)
Host is up (0.0027s latency).
rDNS record for 10.10.1.19: www.goodshopping.com
Not shown: 990 filtered tcp ports (no-response)
PORT     STATE SERVICE       VERSION
25/tcp   open  smtp          Microsoft ESMTP 10.0.17763.1
| smtp-vuln-cve2010-4344:
|_  The SMTP server is not Exim: NOT VULNERABLE
80/tcp   open  http          Microsoft IIS httpd 10.0
|_http-server-header: Microsoft-IIS/10.0
|_http-stored-xss: Couldn't find any stored XSS vulnerabilities.
| http-aspnet-debug:
|_  status: DEBUG is enabled
| http-csrf:
```

Figure 5.20: file where the scan results are saved



Vulnerability Assessment using Python Script with AI

- An attacker can also leverage AI-powered ChatGPT or other generative AI technology to perform this task by using an appropriate prompt such as
sgpt --chat scancode --code "*Create a python script to run a fast but comprehensive Nmap scan on the IP addresses in scan1.txt and then execute vulnerability scanning using nikto against each IP address in scan1.txt*"

```
sgpt --chat scancode --code "Create a python script to run a fast but comprehensive Nmap scan on
File Edit View Search Terminal Help
┌─[root@parrot]─[~]
└──╼ #sgpt --chat scancode --code "Create a python script t
 comprehensive Nmap scan on the IP addresses in scan1.txt a
vulnerability scanning using nikto against each IP address
import subprocess

# Read the list of IP addresses from scan1.txt
with open('scan1.txt', 'r') as file:
    ip_addresses = file.read().splitlines()

# Run Nmap scan on each IP address
for ip in ip_addresses:
    subprocess.run(['nmap', '-T4', '-A', '-v', ip])

    # Run Nikto vulnerability scan on each IP address
    subprocess.run(['nikto', '-h', ip])
┌─[root@parrot]─[~]
└──╼ #
```

```
python3 vulnscan.py - Parrot Terminal
File Edit View Search Terminal Help
- Nikto v2.5.0
---------------------------------------------------------------------------
+ Target IP:          10.10.1.9
+ Target Hostname:    10.10.1.9
+ Target Port:        80
+ Start Time:         2024-03-04 07:55:50 (GMT-5)
---------------------------------------------------------------------------
+ Server: Apache/2.4.52 (Ubuntu)
+ /: The anti-clickjacking X-Frame-Options header is not present
s://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Op
+ /: The X-Content-Type-Options header is not set. This could al
r agent to render the content of the site in a different fashion
E type. See: https://www.netsparker.com/web-vulnerability-scanne
lities/missing-content-type-header/
+ No CGI Directories found (use '-C all' to force check all poss
+ /: Server may leak inodes via ETags, header found with file /,
f, size: 61178ae134d96, mtime: gzip. See: http://cve.mitre.org/c
ame.cgi?name=CVE-2003-1418
+ Apache/2.4.52 appears to be outdated (current is at least Apac
 Apache 2.2.34 is the EOL for the 2.x branch.
+ OPTIONS: Allowed HTTP Methods: GET, POST, OPTIONS, HEAD .
+ 8074 requests: 0 error(s) and 5 item(s) reported on remote hos
+ End Time:           2024-03-04 07:56:06 (GMT-5) (16 seconds)
---------------------------------------------------------------------------
```

```
python3 vulnscan.py - Parrot Terminal
File Edit View Search Terminal Help
┌─[root@parrot]─[~]
└──╼ #python3 vulnscan.py
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-03-04 07:54 EST
NSE: Loaded 156 scripts for scanning.
NSE: Script Pre-scanning.
Initiating NSE at 07:54
Completed NSE at 07:54, 0.00s elapsed
Initiating NSE at 07:54
Completed NSE at 07:54, 0.00s elapsed
Initiating NSE at 07:54
Completed NSE at 07:54, 0.00s elapsed
Initiating ARP Ping Scan at 07:54
Scanning 10.10.1.2 [1 port]
Completed ARP Ping Scan at 07:54, 0.05s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 07:54
Completed Parallel DNS resolution of 1 host. at 07:54, 0.00s elapsed
Initiating SYN Stealth Scan at 07:54
Scanning 10.10.1.2 [1000 ports]
Discovered open port 53/tcp on 10.10.1.2
Discovered open port 88/tcp on 10.10.1.2
Completed SYN Stealth Scan at 07:54, 4.15s elapsed (1000 total ports)
Initiating Service scan at 07:54
Scanning 2 services on 10.10.1.2
```

Vulnerability Assessment using Python Script with AI

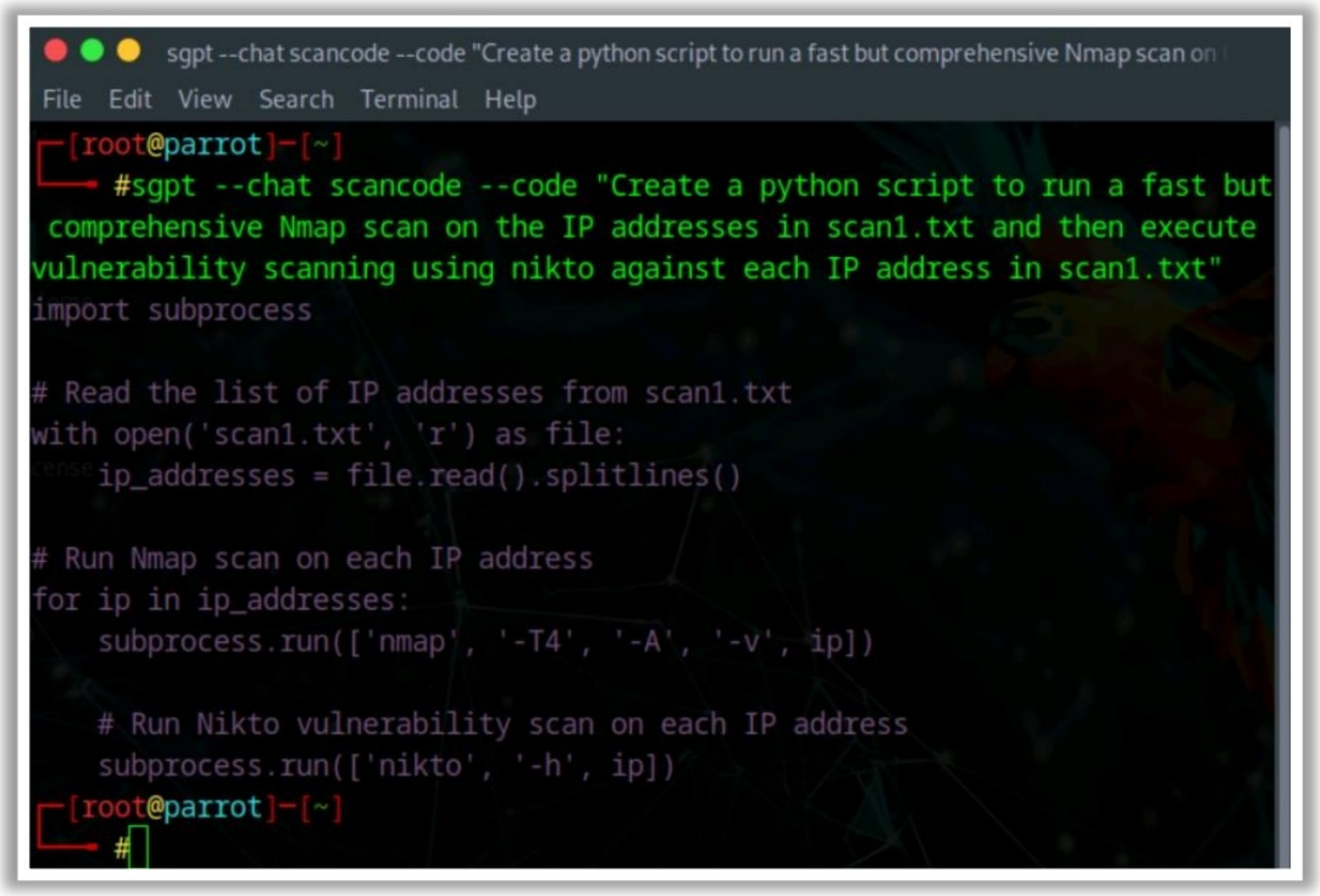
Attackers can leverage AI-powered technologies to enhance and automate their vulnerability scanning tasks. With the aid of AI, attackers can effortlessly create and run custom vulnerability scanning scripts and identify potential vulnerabilities on targets. By developing such custom scripts, attackers can efficiently execute a series of vulnerability scanning and associated commands to identify potential vulnerabilities on targets.

Using this script, an attacker can run fast, but comprehensive, Nmap scans followed by vulnerability scanning using Nikto against multiple IP addresses.

For example,

An attacker can use ChatGPT to perform this task by using an appropriate prompt such as:

"Create a python script to run a fast but comprehensive Nmap scan on the IP addresses in scan1.txt and then execute vulnerability scanning using nikto against each IP address in scan1.txt"



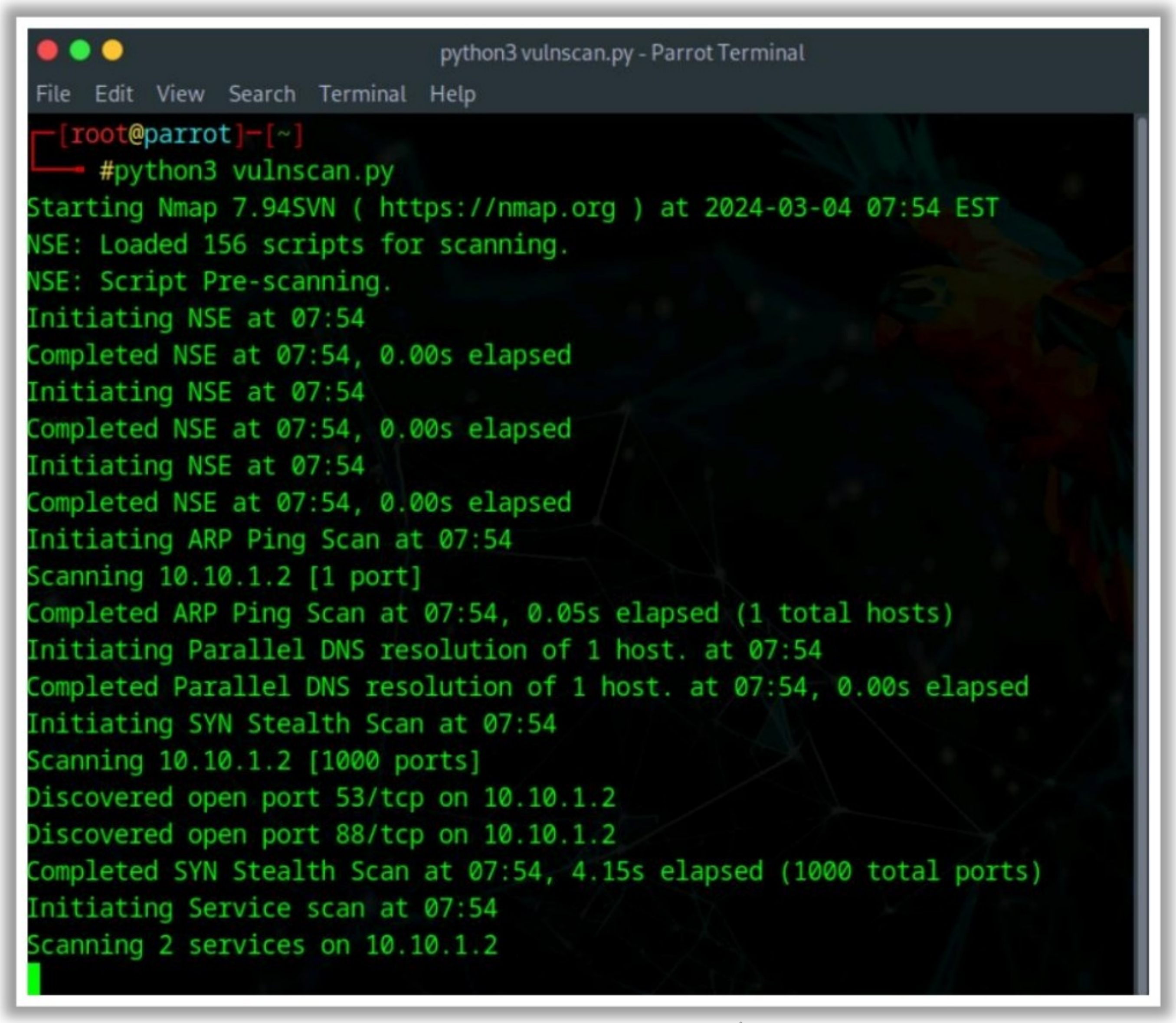
```
sgpt --chat scancode --code "Create a python script to run a fast but comprehensive Nmap scan on
File  Edit  View  Search  Terminal  Help
┌─[root@parrot]─[~]
└──╼ #sgpt --chat scancode --code "Create a python script to run a fast but
 comprehensive Nmap scan on the IP addresses in scan1.txt and then execute
vulnerability scanning using nikto against each IP address in scan1.txt"
import subprocess

# Read the list of IP addresses from scan1.txt
with open('scan1.txt', 'r') as file:
    ip_addresses = file.read().splitlines()

# Run Nmap scan on each IP address
for ip in ip_addresses:
    subprocess.run(['nmap', '-T4', '-A', '-v', ip])

    # Run Nikto vulnerability scan on each IP address
    subprocess.run(['nikto', '-h', ip])
┌─[root@parrot]─[~]
└──╼ #
```

Figure 5.21: Python script that executes nmap and nikto scans

The following Python script automates network scanning and vulnerability assessment tasks on the IP addresses listed in the scan1.txt file:

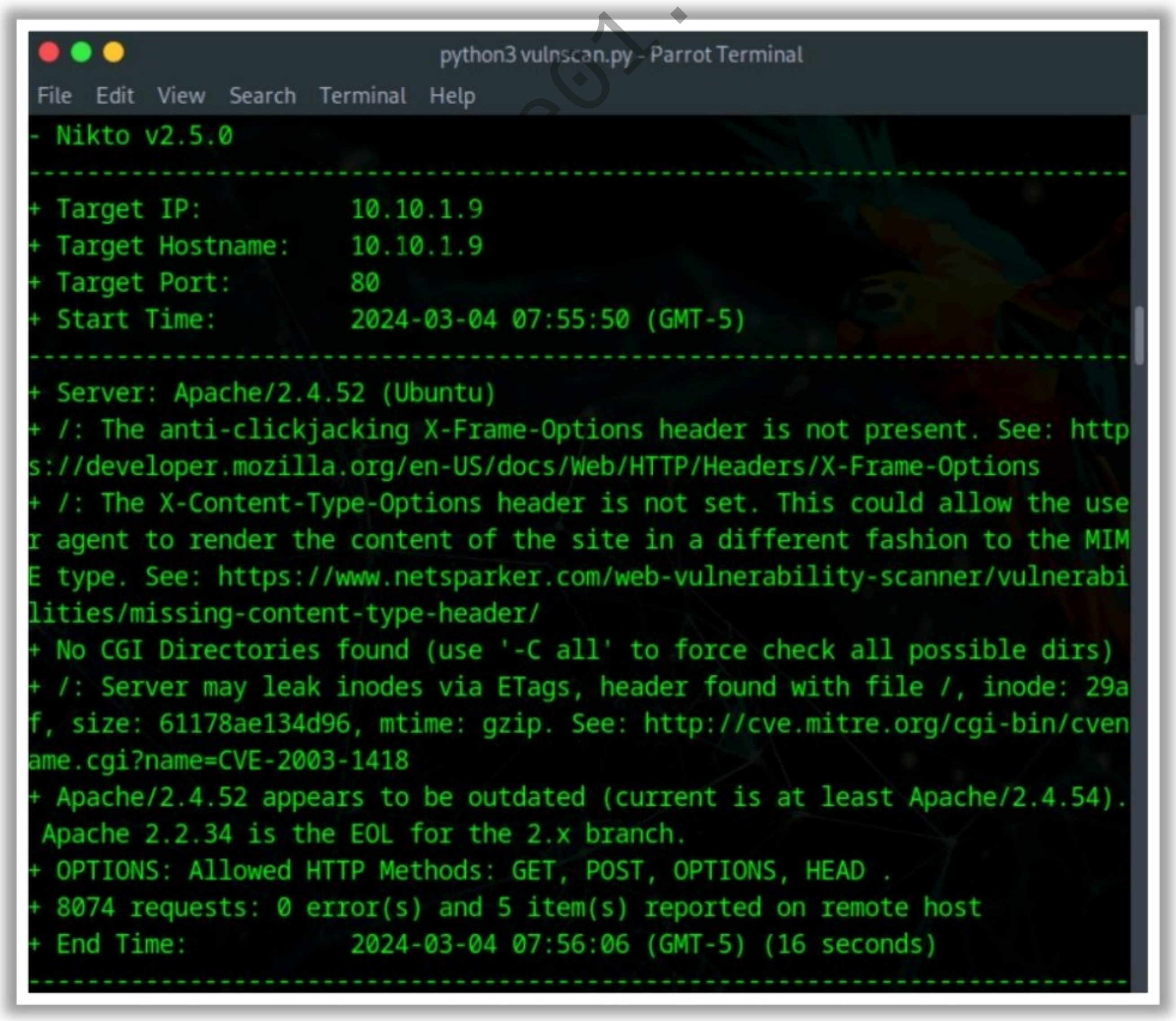
```
import subprocess
# Read the list of IP addresses from scan1.txt
with open('scan1.txt', 'r') as file:
   ip_addresses = file.read().splitlines()
# Run Nmap scan on each IP address
for ip in ip_addresses:
   subprocess.run(['nmap', '-T4', '-A', '-v', ip])
# Run Nikto vulnerability scan on each IP address
subprocess.run(['nikto', '-h', ip])
```

- The script first reads the list of IP addresses from the scan1.txt file.
- It then iterates through each IP address and executes an Nmap scan with the specified options (in this case, -T4 for timing template and -A for aggressive scan) using the subprocess.run() function.
- After completing the Nmap scan, it proceeds to execute a Nikto vulnerability scan on each IP address using the subprocess.run() function again.
- The results of both scans will be displayed in the console output.


python3 vulnscan.py - Parrot Terminal
File Edit View Search Terminal Help

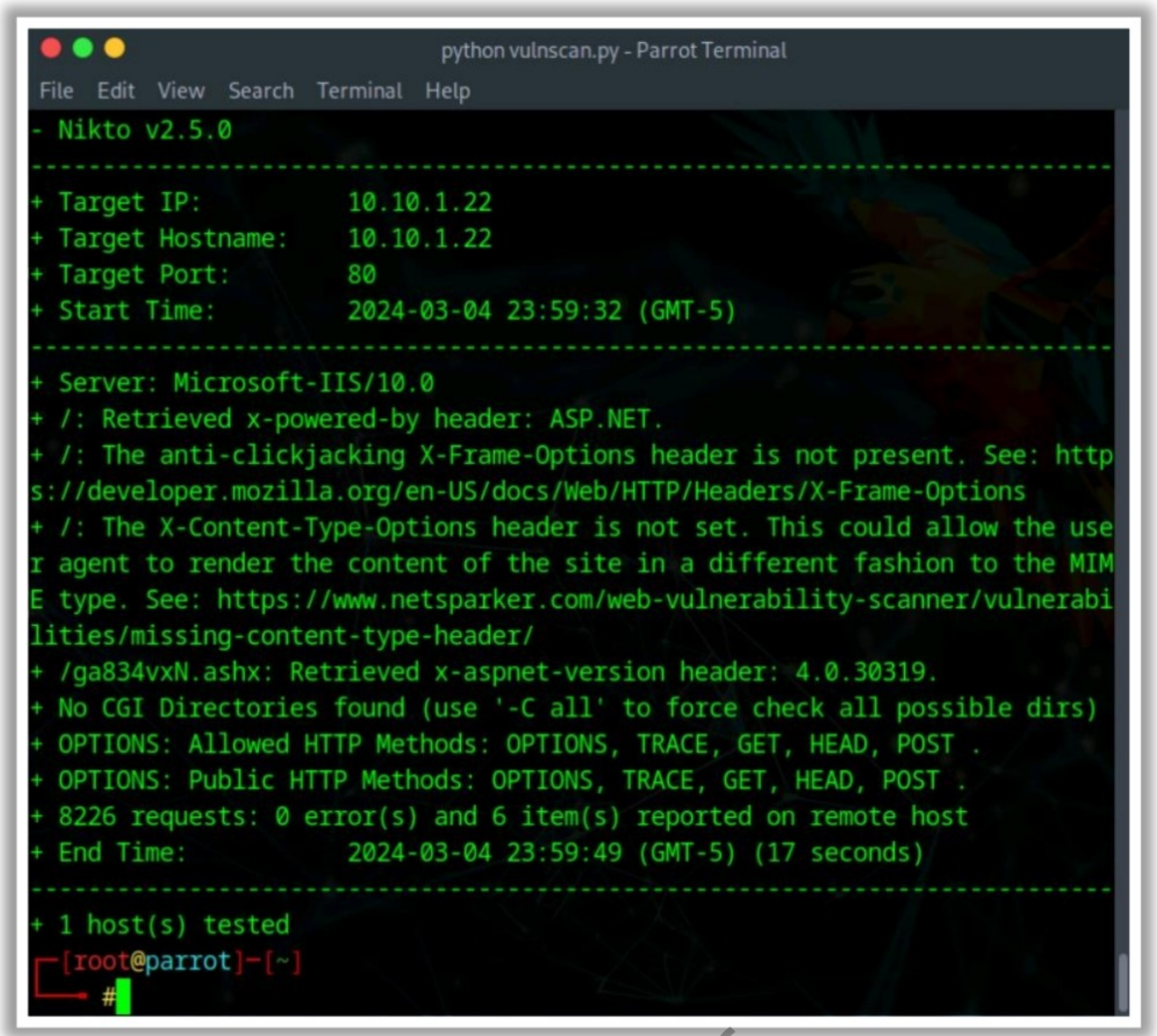
```
┌─[root@parrot]─[~]
└──╼ #python3 vulnscan.py
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-03-04 07:54 EST
NSE: Loaded 156 scripts for scanning.
NSE: Script Pre-scanning.
Initiating NSE at 07:54
Completed NSE at 07:54, 0.00s elapsed
Initiating NSE at 07:54
Completed NSE at 07:54, 0.00s elapsed
Initiating NSE at 07:54
Completed NSE at 07:54, 0.00s elapsed
Initiating ARP Ping Scan at 07:54
Scanning 10.10.1.2 [1 port]
Completed ARP Ping Scan at 07:54, 0.05s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 07:54
Completed Parallel DNS resolution of 1 host. at 07:54, 0.00s elapsed
Initiating SYN Stealth Scan at 07:54
Scanning 10.10.1.2 [1000 ports]
Discovered open port 53/tcp on 10.10.1.2
Discovered open port 88/tcp on 10.10.1.2
Completed SYN Stealth Scan at 07:54, 4.15s elapsed (1000 total ports)
Initiating Service scan at 07:54
Scanning 2 services on 10.10.1.2
```

Figure 5.22: Output of python script of nmap and nikto scans


python3 vulnscan.py - Parrot Terminal
File Edit View Search Terminal Help

```
- Nikto v2.5.0
---------------------------------------------------------------------------
+ Target IP:          10.10.1.9
+ Target Hostname:    10.10.1.9
+ Target Port:        80
+ Start Time:         2024-03-04 07:55:50 (GMT-5)
---------------------------------------------------------------------------
+ Server: Apache/2.4.52 (Ubuntu)
+ /: The anti-clickjacking X-Frame-Options header is not present. See: http
s://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options
+ /: The X-Content-Type-Options header is not set. This could allow the use
r agent to render the content of the site in a different fashion to the MIM
E type. See: https://www.netsparker.com/web-vulnerability-scanner/vulnerabi
lities/missing-content-type-header/
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ /: Server may leak inodes via ETags, header found with file /, inode: 29a
f, size: 61178ae134d96, mtime: gzip. See: http://cve.mitre.org/cgi-bin/cven
ame.cgi?name=CVE-2003-1418
+ Apache/2.4.52 appears to be outdated (current is at least Apache/2.4.54).
 Apache 2.2.34 is the EOL for the 2.x branch.
+ OPTIONS: Allowed HTTP Methods: GET, POST, OPTIONS, HEAD .
+ 8074 requests: 0 error(s) and 5 item(s) reported on remote host
+ End Time:           2024-03-04 07:56:06 (GMT-5) (16 seconds)
---------------------------------------------------------------------------
```

Figure 5.23: Output of python script of nmap and nikto scans


python vulnscan.py - Parrot Terminal
File Edit View Search Terminal Help

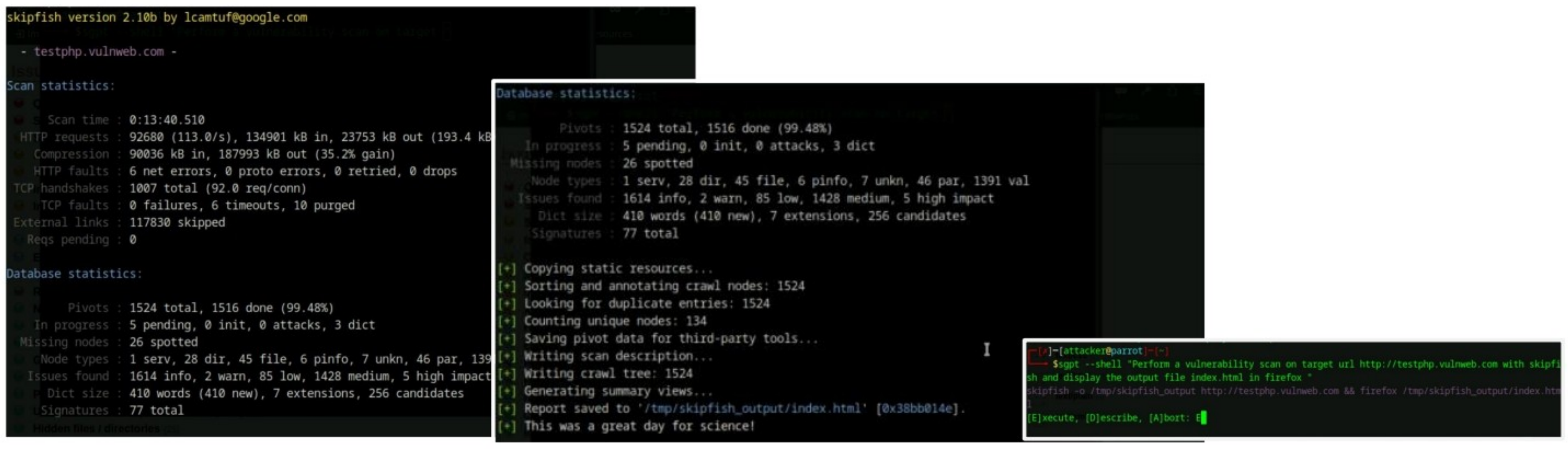
```
- Nikto v2.5.0
---------------------------------------------------------------------------
+ Target IP:          10.10.1.22
+ Target Hostname:    10.10.1.22
+ Target Port:        80
+ Start Time:         2024-03-04 23:59:32 (GMT-5)
---------------------------------------------------------------------------
+ Server: Microsoft-IIS/10.0
+ /: Retrieved x-powered-by header: ASP.NET.
+ /: The anti-clickjacking X-Frame-Options header is not present. See: http
s://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options
+ /: The X-Content-Type-Options header is not set. This could allow the use
r agent to render the content of the site in a different fashion to the MIM
E type. See: https://www.netsparker.com/web-vulnerability-scanner/vulnerabi
lities/missing-content-type-header/
+ /ga834vxN.ashx: Retrieved x-aspnet-version header: 4.0.30319.
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ OPTIONS: Allowed HTTP Methods: OPTIONS, TRACE, GET, HEAD, POST .
+ OPTIONS: Public HTTP Methods: OPTIONS, TRACE, GET, HEAD, POST .
+ 8226 requests: 0 error(s) and 6 item(s) reported on remote host
+ End Time:           2024-03-04 23:59:49 (GMT-5) (17 seconds)
---------------------------------------------------------------------------
+ 1 host(s) tested
┌─[root@parrot]─[~]
└──╼ #
```

Figure 5.24: Output of python script of nmap and nikto scans

This Python script automates network scanning and vulnerability assessment tasks on the IP addresses listed in the scan1.txt file, providing comprehensive insights into potential security vulnerabilities.

Vulnerability Scan using Skipfish with AI

- An attacker can also leverage AI-powered ChatGPT or other generative AI technology to perform this task by using an appropriate prompt such as
  - ***"Perform a vulnerability scan on target url http://testphp.vulnweb.com with Skipfish and display the output file index.html in Firefox "***

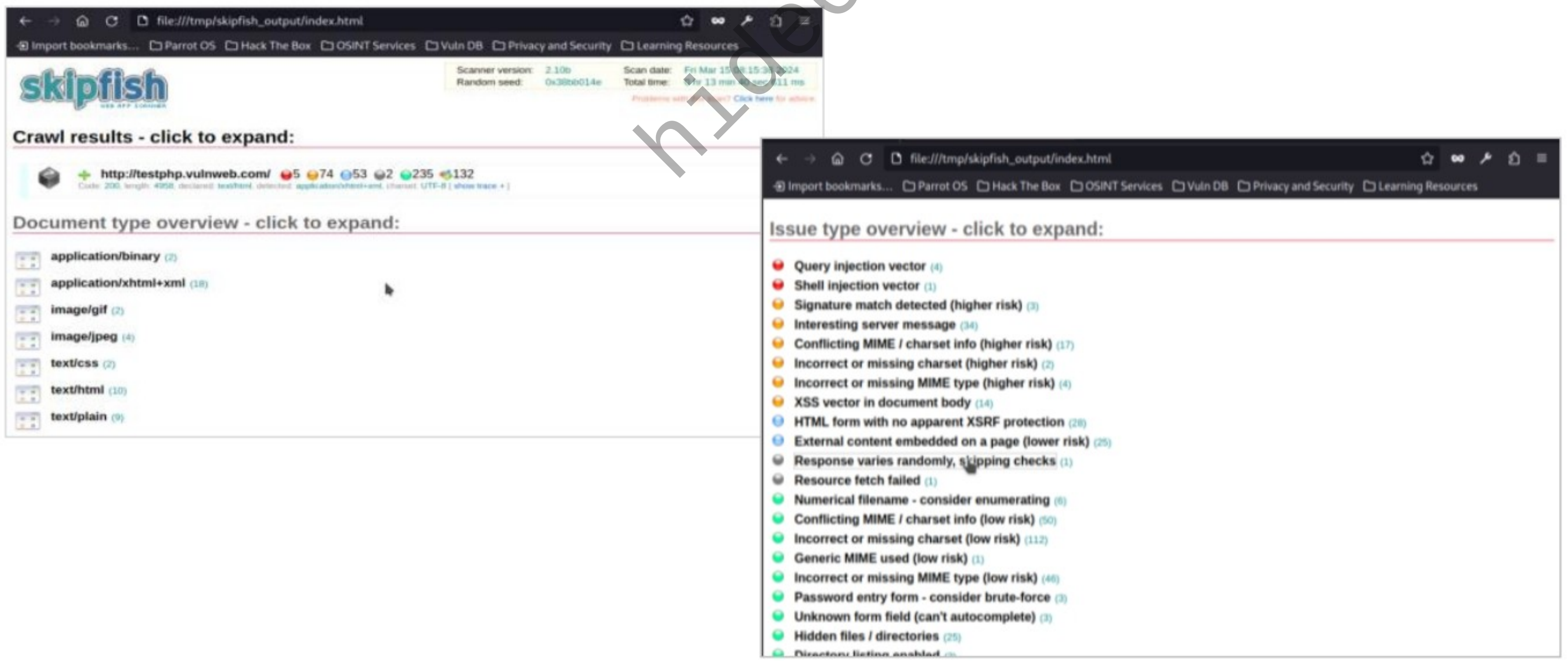


```
skipfish version 2.10b by lcamtuf@google.com

  - testphp.vulnweb.com -

Scan statistics:

      Scan time : 0:13:40.510
  HTTP requests : 92680 (113.0/s), 134901 kB in, 23753 kB out (193.4 kB
    Compression : 90036 kB in, 187993 kB out (35.2% gain)
    HTTP faults : 6 net errors, 0 proto errors, 0 retried, 0 drops
 TCP handshakes : 1007 total (92.0 req/conn)
     TCP faults : 0 failures, 6 timeouts, 10 purged
 External links : 117830 skipped
   Reqs pending : 0

Database statistics:

         Pivots : 1524 total, 1516 done (99.48%)
    In progress : 5 pending, 0 init, 0 attacks, 3 dict
  Missing nodes : 26 spotted
     Node types : 1 serv, 28 dir, 45 file, 6 pinfo, 7 unkn, 46 par, 139
   Issues found : 1614 info, 2 warn, 85 low, 1428 medium, 5 high impact
      Dict size : 410 words (410 new), 7 extensions, 256 candidates
     Signatures : 77 total
```

```
Database statistics:

         Pivots : 1524 total, 1516 done (99.48%)
    In progress : 5 pending, 0 init, 0 attacks, 3 dict
  Missing nodes : 26 spotted
     Node types : 1 serv, 28 dir, 45 file, 6 pinfo, 7 unkn, 46 par, 1391 val
   Issues found : 1614 info, 2 warn, 85 low, 1428 medium, 5 high impact
      Dict size : 410 words (410 new), 7 extensions, 256 candidates
     Signatures : 77 total

[+] Copying static resources...
[+] Sorting and annotating crawl nodes: 1524
[+] Looking for duplicate entries: 1524
[+] Counting unique nodes: 134
[+] Saving pivot data for third-party tools...
[+] Writing scan description...
[+] Writing crawl tree: 1524
[+] Generating summary views...
[+] Report saved to '/tmp/skipfish_output/index.html' [0x38bb014e].
[+] This was a great day for science!
```

```
┌─[✗]─[attacker@parrot]─[~]
└──╼ $sgpt --shell "Perform a vulnerability scan on target url http://testphp.vulnweb.com with skipfi
sh and display the output file index.html in firefox "
skipfish -o /tmp/skipfish_output http://testphp.vulnweb.com && firefox /tmp/skipfish_output/index.htm
l
[E]xecute, [D]escribe, [A]bort: E
```

Vulnerability Scan using Skipfish with AI (Cont'd)


file:///tmp/skipfish_output/index.html
Import bookmarks... Parrot OS Hack The Box OSINT Services Vuln DB Privacy and Security Learning Resources
skipfish

Scanner version: 2.10b Scan date: Fri Mar 15 08:15:38 2024
Random seed: 0x38bb014e Total time: 0 hr 13 min 40 sec 511 ms

Crawl results - click to expand:

http://testphp.vulnweb.com/ 5 74 53 2 235 132

Document type overview - click to expand:

- application/binary (2)
- application/xhtml+xml (18)
- image/gif (2)
- image/jpeg (4)
- text/css (2)
- text/html (10)
- text/plain (9)

Issue type overview - click to expand:

- Query injection vector (4)
- Shell injection vector (1)
- Signature match detected (higher risk) (3)
- Interesting server message (34)
- Conflicting MIME / charset info (higher risk) (17)
- Incorrect or missing charset (higher risk) (2)
- Incorrect or missing MIME type (higher risk) (4)
- XSS vector in document body (14)
- HTML form with no apparent XSRF protection (28)
- External content embedded on a page (lower risk) (25)
- Response varies randomly, skipping checks (1)
- Resource fetch failed (1)
- Numerical filename - consider enumerating (6)
- Conflicting MIME / charset info (low risk) (50)
- Incorrect or missing charset (low risk) (112)
- Generic MIME used (low risk) (1)
- Incorrect or missing MIME type (low risk) (46)
- Password entry form - consider brute-force (3)
- Unknown form field (can't autocomplete) (3)
- Hidden files / directories (25)
- Directory listing enabled

Vulnerability Scan using Skipfish with AI

Attackers can leverage AI-powered technologies to enhance and automate their vulnerability scanning tasks. With the aid of AI, attackers can effortlessly perform vulnerability scanning using Skipfish to identify potential vulnerabilities on a target.

For example,

An attacker can use ChatGPT to perform this task by using an appropriate prompt such as:

“Perform a vulnerability scan on target url http://testphp.vulnweb.com with Skipfish and display the output file index.html in Firefox.”

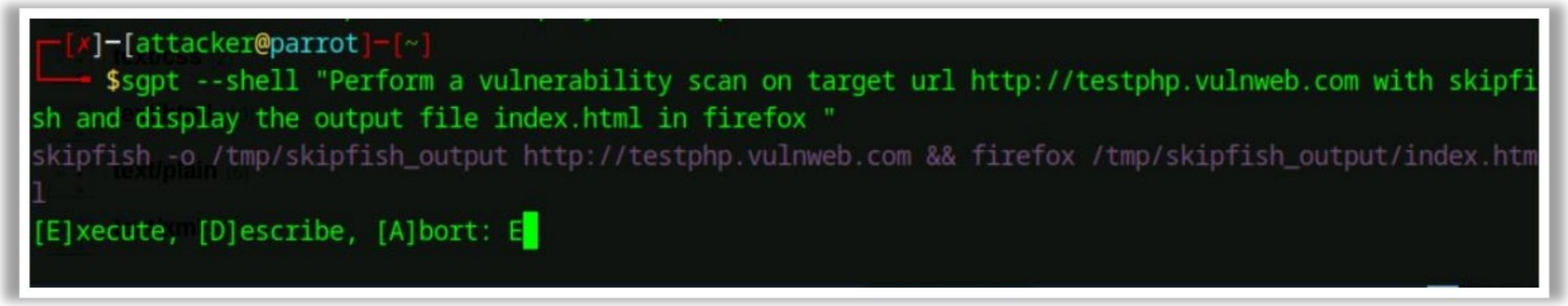

```
┌─[✗]─[attacker@parrot]─[~]
└──╼ $sgpt --shell "Perform a vulnerability scan on target url http://testphp.vulnweb.com with skipfi
sh and display the output file index.html in firefox "
skipfish -o /tmp/skipfish_output http://testphp.vulnweb.com && firefox /tmp/skipfish_output/index.htm
l
[E]xecute, [D]escribe, [A]bort: E
```

Figure 5.25: Prompt that executes the skipfish command

The following command automates vulnerability scanning on the target URL using Skipfish and displays the output file in Firefox:

```
skipfish -o /tmp/skipfish_output http://testphp.vulnweb.com && firefox
tmp/skipfish_output/index.html
```

- The script executes the **`skipfish`** command to perform a vulnerability scan on the target URL **`http://testphp.vulnweb.com`**.
- The **`-o /tmp/skipfish_output`** option specifies the output directory for storing the scan results.
- After completing the vulnerability scan, the script opens the output file **`index.html`** in Firefox using the **`firefox`** command.

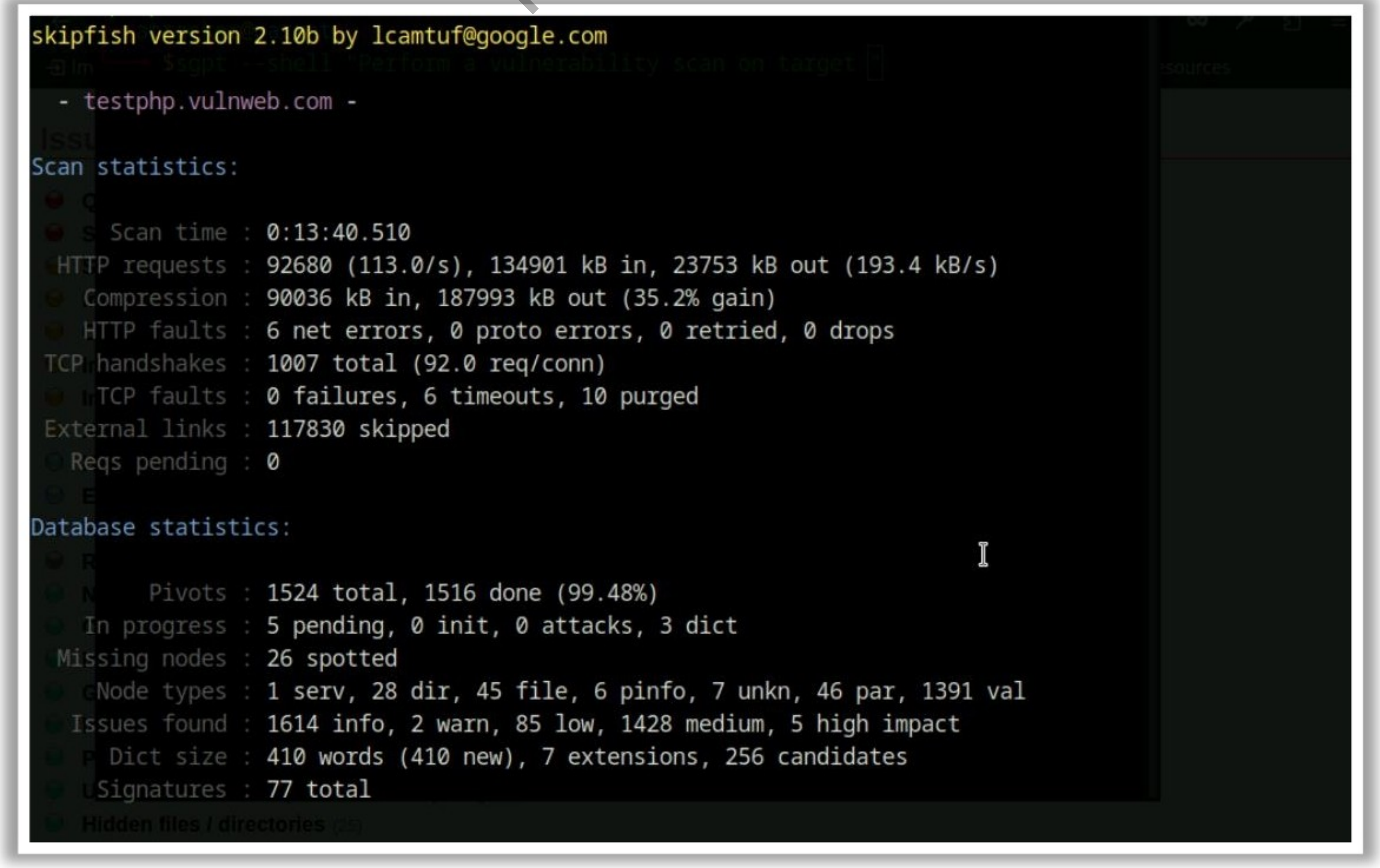
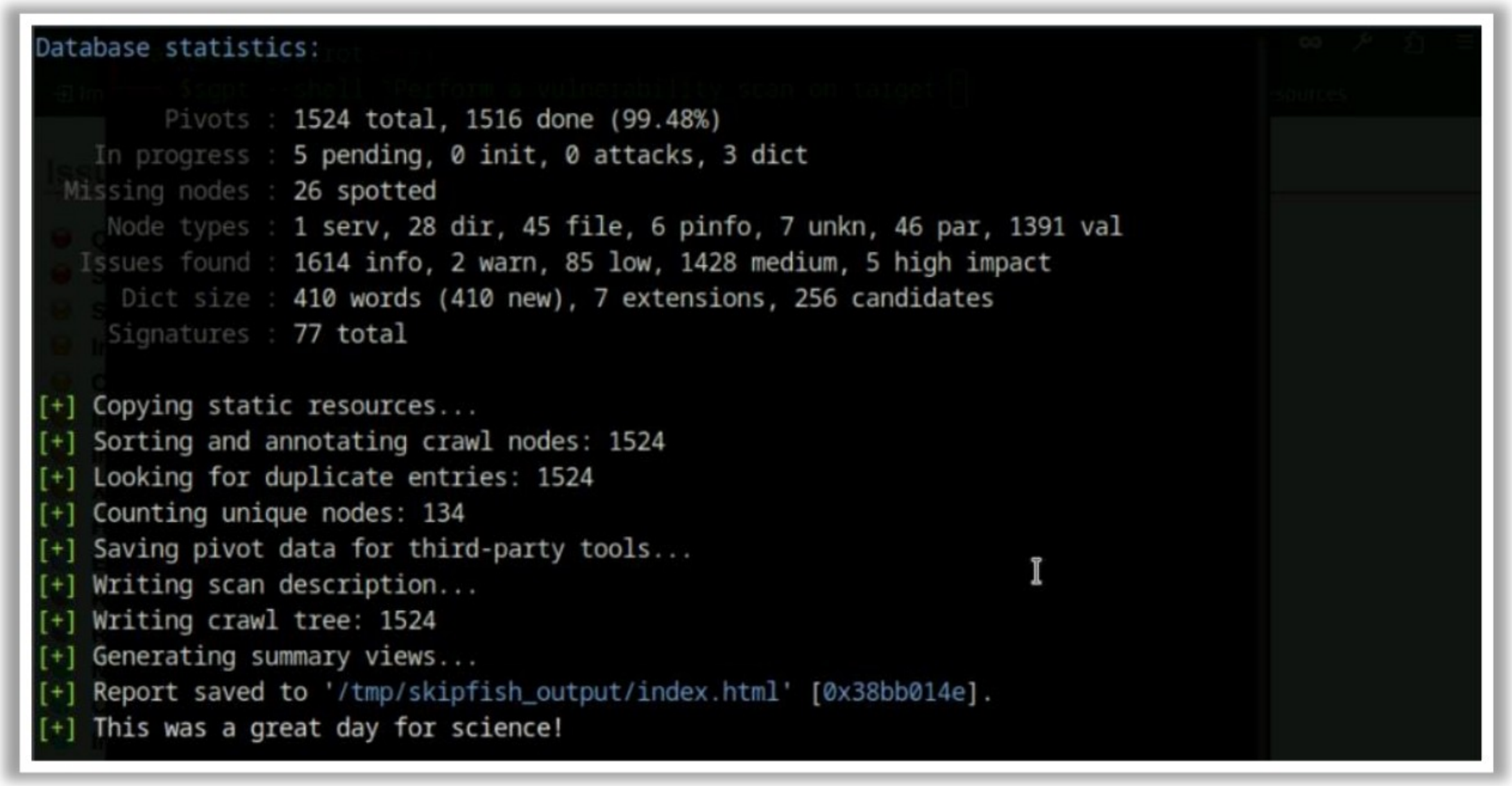
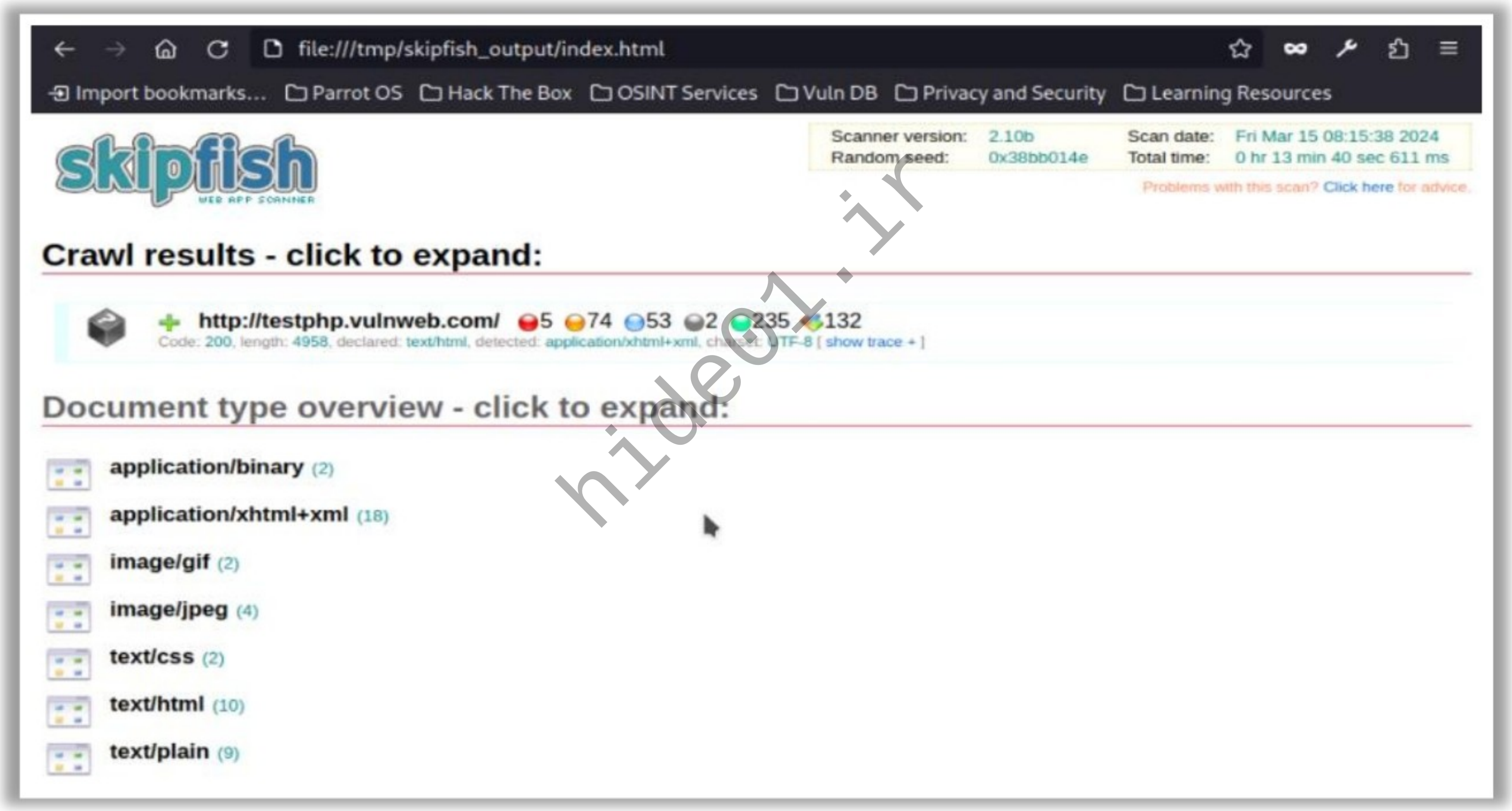

```
skipfish version 2.10b by lcamtuf@google.com

  - testphp.vulnweb.com -

Scan statistics:

      Scan time : 0:13:40.510
  HTTP requests : 92680 (113.0/s), 134901 kB in, 23753 kB out (193.4 kB/s)
    Compression : 90036 kB in, 187993 kB out (35.2% gain)
    HTTP faults : 6 net errors, 0 proto errors, 0 retried, 0 drops
 TCP handshakes : 1007 total (92.0 req/conn)
     TCP faults : 0 failures, 6 timeouts, 10 purged
 External links : 117830 skipped
   Reqs pending : 0

Database statistics:

         Pivots : 1524 total, 1516 done (99.48%)
    In progress : 5 pending, 0 init, 0 attacks, 3 dict
  Missing nodes : 26 spotted
     Node types : 1 serv, 28 dir, 45 file, 6 pinfo, 7 unkn, 46 par, 1391 val
   Issues found : 1614 info, 2 warn, 85 low, 1428 medium, 5 high impact
      Dict size : 410 words (410 new), 7 extensions, 256 candidates
     Signatures : 77 total
```

Figure 5.26: Output of prompt that executes the skipfish command



```
Database statistics:

      Pivots : 1524 total, 1516 done (99.48%)
 In progress : 5 pending, 0 init, 0 attacks, 3 dict
Missing nodes : 26 spotted
  Node types : 1 serv, 28 dir, 45 file, 6 pinfo, 7 unkn, 46 par, 1391 val
Issues found : 1614 info, 2 warn, 85 low, 1428 medium, 5 high impact
   Dict size : 410 words (410 new), 7 extensions, 256 candidates
  Signatures : 77 total

[+] Copying static resources...
[+] Sorting and annotating crawl nodes: 1524
[+] Looking for duplicate entries: 1524
[+] Counting unique nodes: 134
[+] Saving pivot data for third-party tools...
[+] Writing scan description...
[+] Writing crawl tree: 1524
[+] Generating summary views...
[+] Report saved to '/tmp/skipfish_output/index.html' [0x38bb014e].
[+] This was a great day for science!
```

Figure 5.27: Output of prompt that executes the skipfish command



Scanner version: 2.10b Scan date: Fri Mar 15 08:15:38 2024
Random seed: 0x38bb014e Total time: 0 hr 13 min 40 sec 611 ms

Problems with this scan? Click here for advice.

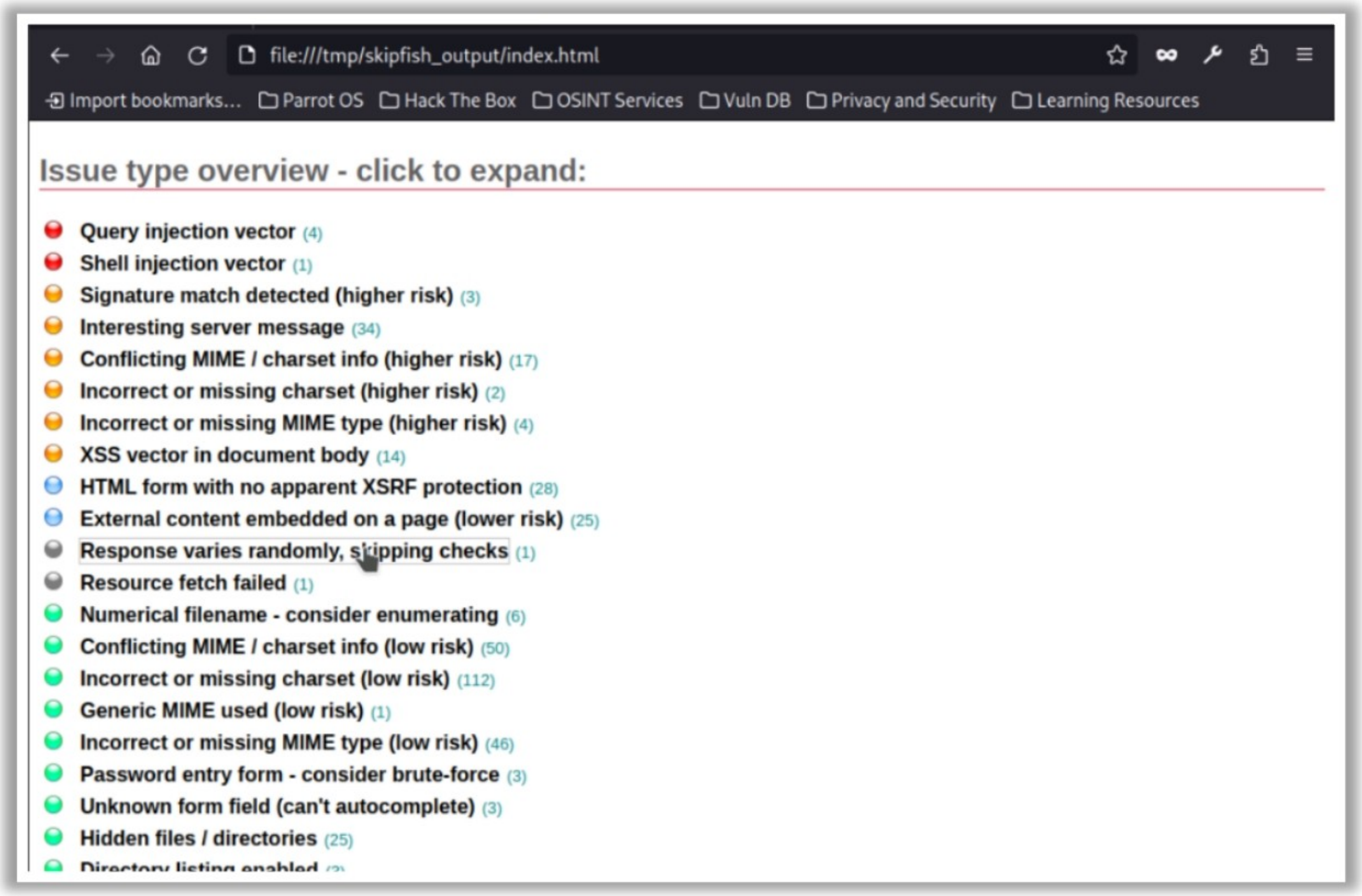
Crawl results - click to expand:

http://testphp.vulnweb.com/ 5 74 53 2 235 132
Code: 200, length: 4958, declared: text/html, detected: application/xhtml+xml, charset: UTF-8 [show trace +]

Document type overview - click to expand:

application/binary (2)
application/xhtml+xml (18)
image/gif (2)
image/jpeg (4)
text/css (2)
text/html (10)
text/plain (9)

Figure 5.28: Output of prompt that executes the skipfish command

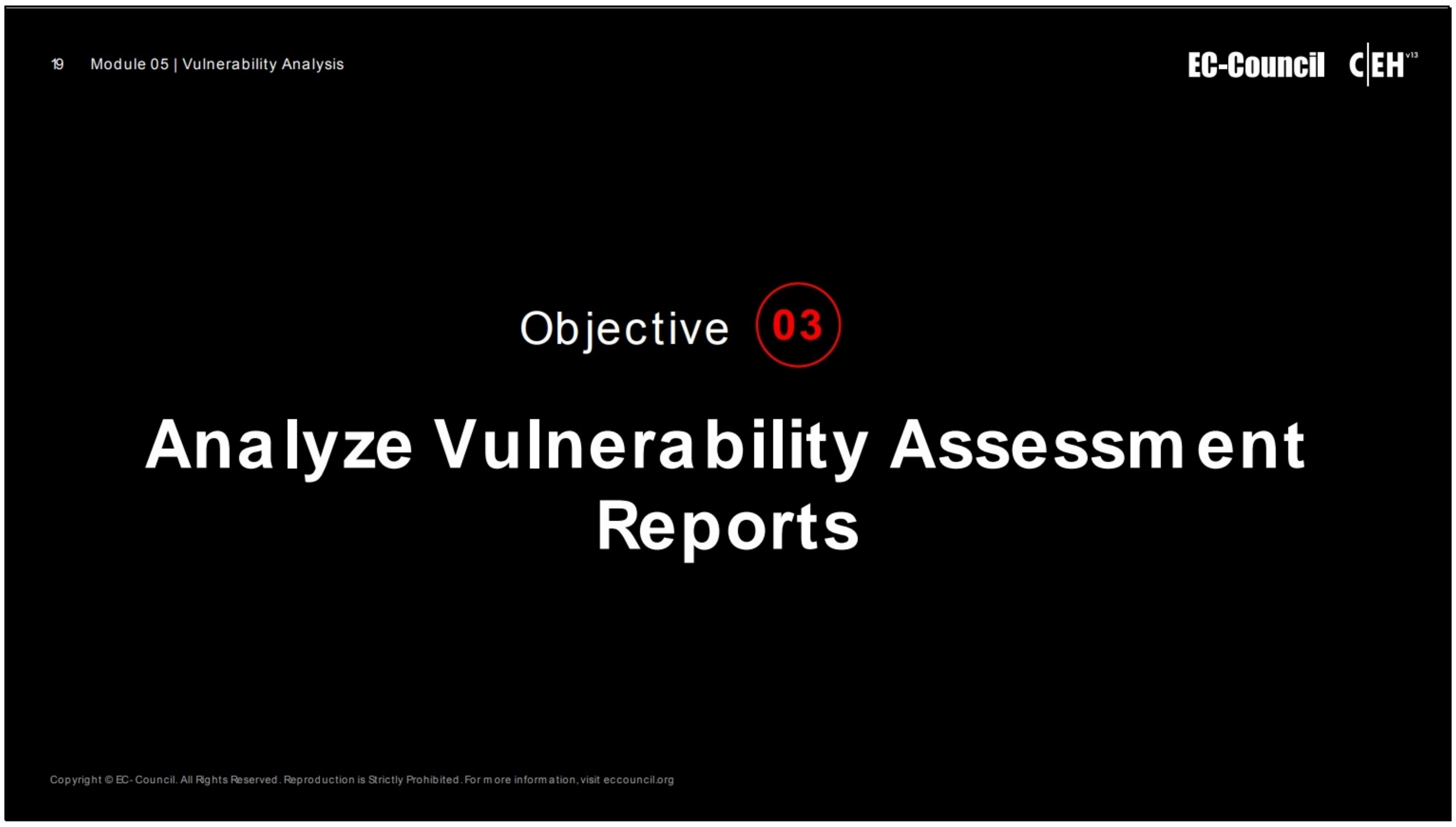


Issue type overview - click to expand:

- Query injection vector (4)
- Shell injection vector (1)
- Signature match detected (higher risk) (3)
- Interesting server message (34)
- Conflicting MIME / charset info (higher risk) (17)
- Incorrect or missing charset (higher risk) (2)
- Incorrect or missing MIME type (higher risk) (4)
- XSS vector in document body (14)
- HTML form with no apparent XSRF protection (28)
- External content embedded on a page (lower risk) (25)
- Response varies randomly, skipping checks (1)
- Resource fetch failed (1)
- Numerical filename - consider enumerating (6)
- Conflicting MIME / charset info (low risk) (50)
- Incorrect or missing charset (low risk) (112)
- Generic MIME used (low risk) (1)
- Incorrect or missing MIME type (low risk) (46)
- Password entry form - consider brute-force (3)
- Unknown form field (can't autocomplete) (3)
- Hidden files / directories (25)
- Directory listing enabled

Figure 5.29: Output of prompt that executes the skipfish command

This prompt automates vulnerability scanning on the target URL http://testphp.vulnweb.com using Skipfish and displays the output file in Firefox for further analysis.

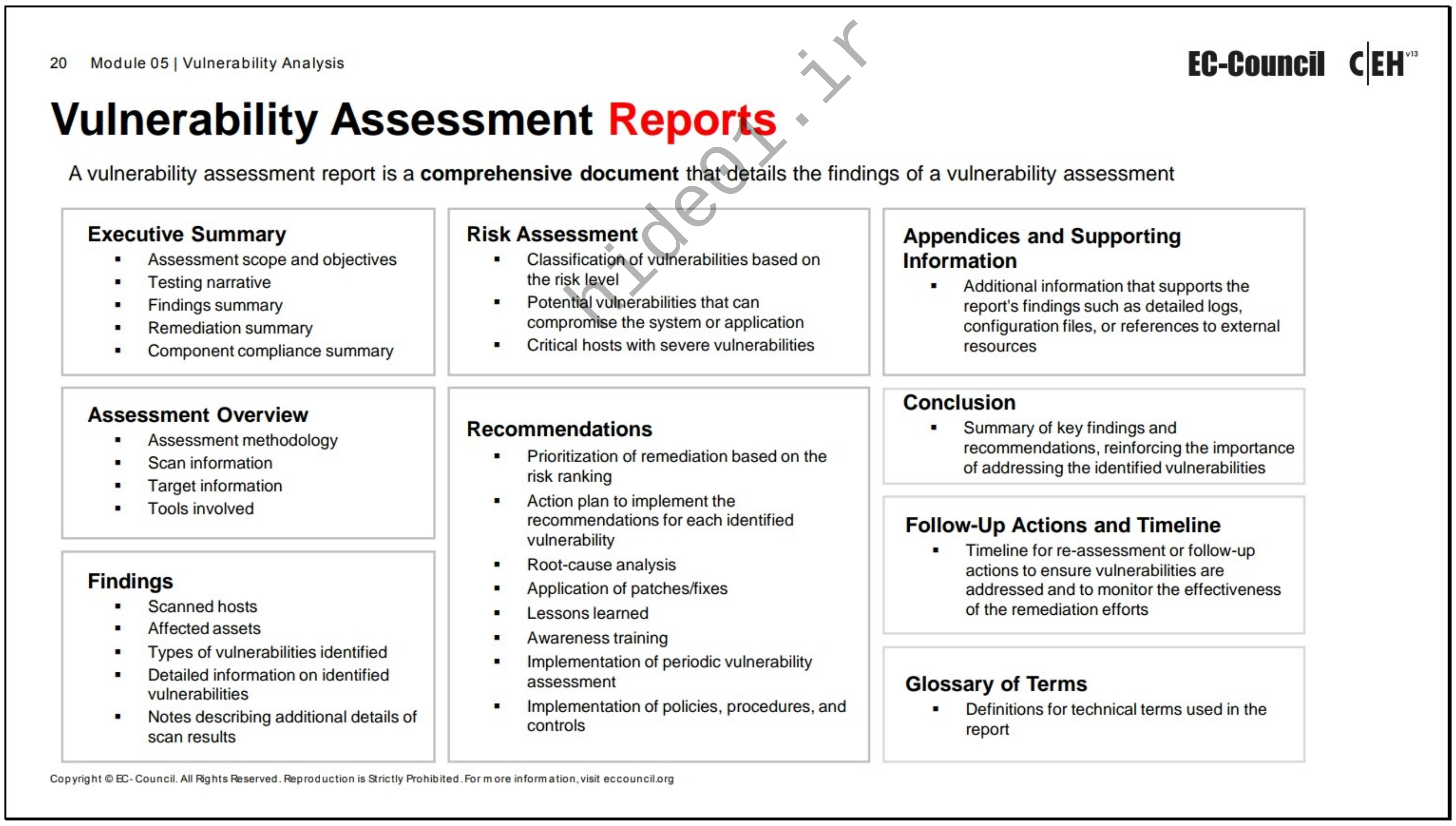


19 Module 05 | Vulnerability Analysis

EC-Council C|EH v13

Objective 03

Analyze Vulnerability Assessment Reports



20 Module 05 | Vulnerability Analysis

EC-Council C|EH v13

Vulnerability Assessment Reports

A vulnerability assessment report is a **comprehensive document** that details the findings of a vulnerability assessment

Executive Summary
- Assessment scope and objectives
- Testing narrative
- Findings summary
- Remediation summary
- Component compliance summary

Assessment Overview
- Assessment methodology
- Scan information
- Target information
- Tools involved

Findings
- Scanned hosts
- Affected assets
- Types of vulnerabilities identified
- Detailed information on identified vulnerabilities
- Notes describing additional details of scan results

Risk Assessment
- Classification of vulnerabilities based on the risk level
- Potential vulnerabilities that can compromise the system or application
- Critical hosts with severe vulnerabilities

Recommendations
- Prioritization of remediation based on the risk ranking
- Action plan to implement the recommendations for each identified vulnerability
- Root-cause analysis
- Application of patches/fixes
- Lessons learned
- Awareness training
- Implementation of periodic vulnerability assessment
- Implementation of policies, procedures, and controls

Appendices and Supporting Information
- Additional information that supports the report's findings such as detailed logs, configuration files, or references to external resources

Conclusion
- Summary of key findings and recommendations, reinforcing the importance of addressing the identified vulnerabilities

Follow-Up Actions and Timeline
- Timeline for re-assessment or follow-up actions to ensure vulnerabilities are addressed and to monitor the effectiveness of the remediation efforts

Glossary of Terms
- Definitions for technical terms used in the report

Vulnerability Assessment Reports

In the vulnerability assessment process, once all the phases are completed, the security team will review the results and process the information to prepare the final report. In this phase, the security team will try to disclose any identified vulnerabilities, document any variations and findings, and include all these in the final report along with remediation steps to mitigate the identified risks.

A vulnerability assessment report is a comprehensive document that details the findings of a vulnerability assessment. This report includes information about identified security weaknesses, their potential impact, severity, and recommendations for remediation. The purpose of the report is to provide stakeholders with a clear understanding of the security posture of the assessed systems, applications, or networks and to guide them in taking corrective actions to mitigate risks.

The report provides details of all the possible vulnerabilities with regard to the company's security policies. The vulnerabilities are categorized based on severity into three levels: High, Medium, and Low risk.

High-risk vulnerabilities are those that might allow unauthorized access to the network. These vulnerabilities must be rectified immediately before the network is compromised. The report describes different kinds of attacks that are possible given the organization's set of operating systems, network components, and protocols.

The vulnerability assessment report must include, but are not limited to, the following points:

- The vulnerability's name and its mapped CVE ID
- The date of discovery
- The score based on Common Vulnerabilities and Exposures (CVE) databases
- A detailed description of the vulnerability
- The impact of the vulnerability
- Details regarding the affected systems
- Details regarding the process needed to correct the vulnerability, including information patches, configuration fixes, and ports to be blocked.
- A proof of concept (PoC) of the vulnerability for the system (if possible)

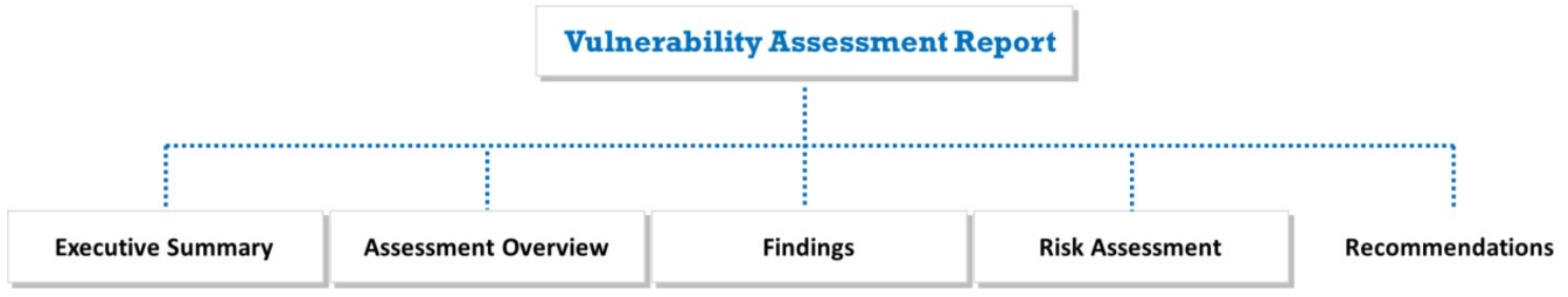


Figure 5.30: Components of a vulnerability assessment report

Components of a Vulnerability Assessment Report

A vulnerability assessment report provides detailed information regarding the vulnerabilities found in the computing environment. The report helps organizations identify the security posture of computing systems (such as web servers, firewalls, routers, email, and file services) and provide solutions to reduce system failures. An ethical hacker must be careful when analyzing vulnerability assessment reports to avoid false positives.

The assessment report helps organizations take mitigation steps to avoid risk proactively by identifying, tracking, and eliminating security vulnerabilities.

Vulnerability assessment reports are classified into two types:

- Security vulnerability reports
- Security vulnerability summaries

Security Vulnerability Report

This is a combined report of all the scanned devices and servers in the organization's network.

The security vulnerability report includes the following details:

- Newly found vulnerabilities
- Open ports and detected services
- Suggestions for remediation
- Links to patches

Security Vulnerability Summary

This report is produced for every device or server after scanning. It provides a summary of the scan result, which includes the following elements:

- Current security flaws
- Categories of vulnerabilities
- Newly detected security vulnerabilities
- Severity of vulnerabilities
- Resolved vulnerabilities

A vulnerability assessment report covers the following elements:

- **Executive Summary**
  - Assessment scope and objectives
    - Purpose of the vulnerability scanning
    - Scope of the scanning
  - Testing narrative
    - Operating systems upon which scanning is performed
    - IP addresses upon which scanning is performed
    - Types of scans performed
    - Date and time (Including start, end, and duration of scan)
  - Findings summary
    - Critical vulnerabilities identified (highlights based on risk level)
      - ✓ Number of vulnerabilities based on severity (graphical representation)

        - Identified operating systems
        - Performance of the systems and applications during the scan
        - Overall risk level
        - Critical issues that need to be addressed
    - Remediation summary
    - Component compliance summary
- **Assessment Overview**
    - Assessment methodology
    - Scan information: information such as the type of scan performed, tools used, versions, and the assets scanned.
    - Target information: Information about the target system's name and address.
    - Tools involved: Type of tools used to scan for vulnerabilities.
- **Findings**
    - Scanned hosts, including each host's detailed information
        - **<Node>**: Name and address of the host
        - <**OS>**: Operating system type
        - **<Date>**: Date of the test
        - **Vulnerable services:** Network services by their names and ports.
    - Affected assets
    - Types of vulnerabilities identified
    - Detailed information on identified vulnerabilities (including CVE ID, CVSS score, threat description, impact caused, remediation, and exploitability)
    - Notes describing additional details of scan results
- **Risk Assessment**
    - Classification of vulnerabilities based on the risk level: critical, high, moderate, or low
    - Potential vulnerabilities that can compromise the system or application
    - Critical hosts with severe vulnerabilities
- **Recommendations**
    - Prioritization of remediation based on the risk ranking
    - Action plan to implement the recommendations/remediation for each identified vulnerability

- Root-cause analysis
- Application of patches/fixes
- Lessons learned
- Awareness training
- Implementation of periodic vulnerability assessment
- Implementation of policies, procedures, and controls

▪ **Appendices and Supporting Information**

- Additional information that supports the report's findings or might be useful for technical teams during the remediation process, such as detailed logs, configuration files, or references to external resources.

▪ **Conclusion**

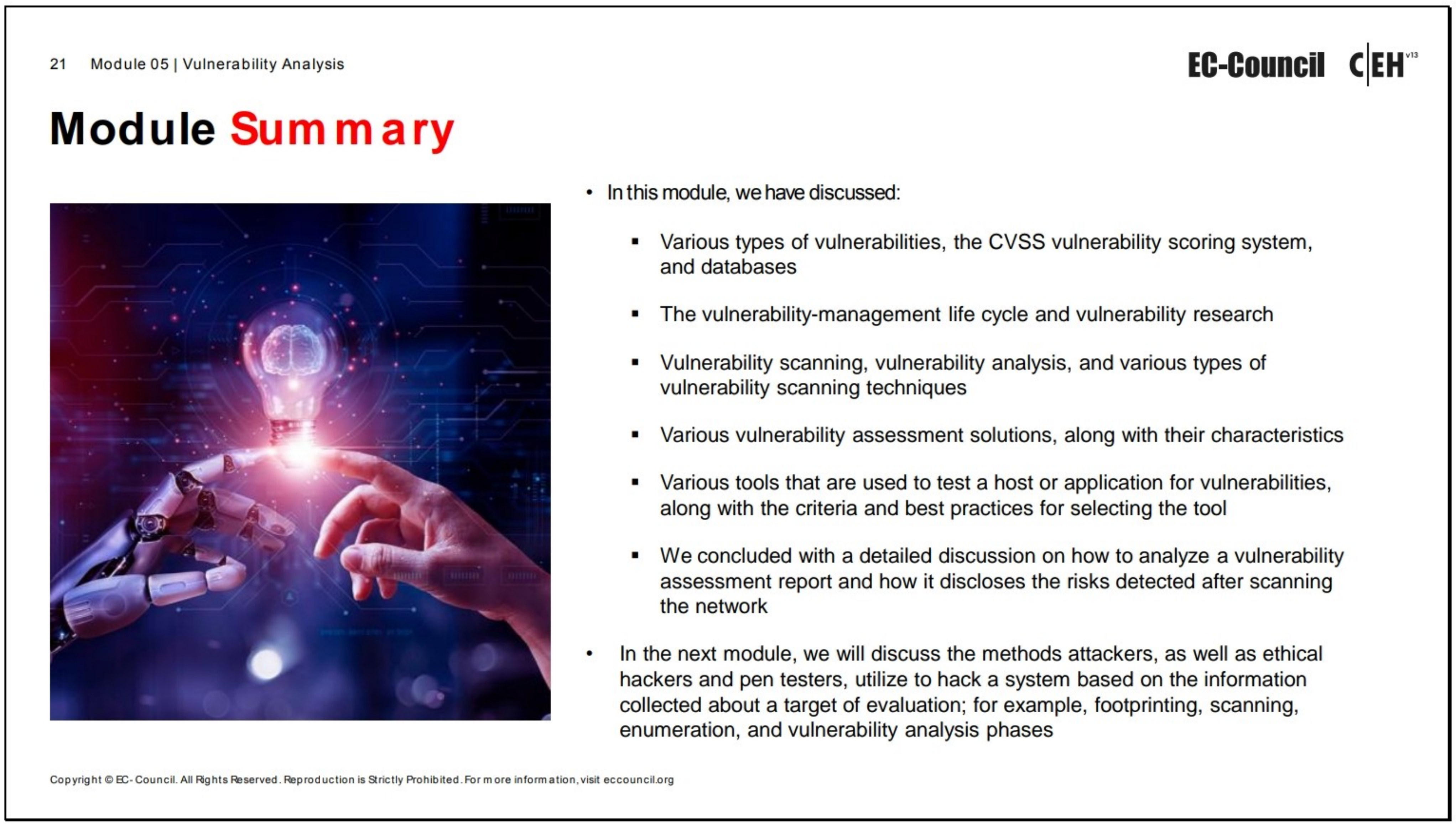
- Summary of key findings and recommendations, reinforcing the importance of addressing the identified vulnerabilities to improve the organization's security posture.

▪ **Follow-Up Actions and Timeline**

- Timeline for re-assessment or follow-up actions to ensure vulnerabilities are addressed and to monitor the effectiveness of the remediation efforts.

▪ **Glossary of Terms**

- Definitions for technical terms used in the report, making it accessible to readers with varying levels of cybersecurity knowledge.



EC-Council C|EH v13

Module Summary

- In this module, we have discussed:
  - Various types of vulnerabilities, the CVSS vulnerability scoring system, and databases
  - The vulnerability-management life cycle and vulnerability research
  - Vulnerability scanning, vulnerability analysis, and various types of vulnerability scanning techniques
  - Various vulnerability assessment solutions, along with their characteristics
  - Various tools that are used to test a host or application for vulnerabilities, along with the criteria and best practices for selecting the tool
  - We concluded with a detailed discussion on how to analyze a vulnerability assessment report and how it discloses the risks detected after scanning the network
- In the next module, we will discuss the methods attackers, as well as ethical hackers and pen testers, utilize to hack a system based on the information collected about a target of evaluation; for example, footprinting, scanning, enumeration, and vulnerability analysis phases

Module Summary

This module discussed various types of vulnerabilities, the CVSS vulnerability scoring system, and vulnerability databases. It also covered the vulnerability management lifecycle and vulnerability research, as well as vulnerability scanning, vulnerability analysis, and various types of vulnerability scanning techniques. It described various vulnerability assessment solutions along with their characteristics and described various vulnerability assessment tools that are used to test a host or application for vulnerabilities, along with the criteria and best practices for selecting the tool. Finally, this module ended with a detailed discussion on how to analyze a vulnerability assessment report and how it discloses the risks detected after scanning a network.

The next module will show how attackers, as well as ethical hackers and pen testers, attempt system hacking based on the information collected about a target in the footprinting, scanning, enumeration, and vulnerability analysis phases.