Trends in Denial of Service Attack Technology

CERT® Coordination Center

Kevin J. Houle, CERT/CC
George M. Weaver, CERT/CC

In collaboration with:
Neil Long
Rob Thomas

v1.0
October 2001

1 Abstract

In November of 1999, the CERT® Coordination Center (CERT/CC) sponsored the Distributed Systems Intruder Tools (DSIT) Workshop where a group of security experts outlined the emerging threat of distributed denial of service (DDoS) attack technology. Since then, denial of service (DoS) attack technology has continued to evolve and continues to be used to attack and impact Internet infrastructures.

Advances in intruder automation techniques have led to a steady stream of new self-propagating worms in 2001, some of which have been used to deploy DoS attack technology. Windows end-users and Internet routing technology have both become more frequent targets of intruder activity. The control mechanisms for DDoS attack networks are changing to make greater use of Internet Relay Chat (IRC) technology. The impacts of DoS attacks are causing greater collateral damage, and widespread automated propagation itself has become a vehicle for causing denial of service.

While DoS attack technology continues to evolve, the circumstances enabling attacks have not significantly changed in recent years. DoS attacks remain a serious threat to the users, organizations, and infrastructures of the Internet.

The goal of this paper is to highlight recent trends in the deployment, use, and impact of DoS attack technology based on intruder activity and attack tools reported to and analyzed by the CERT/CC. This paper does not propose solutions, but rather aims to serve as a catalyst to raise awareness and stimulate further discussion of DoS related issues within the Internet community.

2 Introduction

The traditional intent and impact of DoS attacks is to prevent or impair the legitimate use of computer or network resources. Regardless of the diligence, effort, and resources spent securing against intrusion, Internet connected systems face a consistent and real threat from DoS attacks because of two fundamental characteristics of the Internet.

- The Internet is comprised of limited and consumable resources

  The infrastructure of interconnected systems and networks comprising the Internet is entirely composed of limited resources. Bandwidth, processing power, and storage capacities are all common targets for DoS attacks designed to consume enough of a target's available resources to cause some level of service disruption. An abundance of well-engineered resources may raise the bar on the degree an attack must reach to be effective, but today's attack methods and tools place even the most abundant resources in range for disruption.

- Internet security is highly interdependent

  DoS attacks are commonly launched from one or more points on the Internet that are external to the victim's own system or network. In many cases, the launch point consists of one or more systems that have been subverted by an intruder via a security-related compromise rather than from the intruder's own system or systems. As such, intrusion defense not only helps to protect Internet assets and the mission they support, but it also helps prevent the use of assets to attack other Internet-connected networks and systems. Likewise, regardless of how well defended your assets may be, your susceptibility to many types of attacks, particularly DoS attacks, depends on the state of security on the rest of the global Internet.

Defending against DoS attacks is far from an exact or complete science. Rate limiting, packet filtering, and tweaking software parameters can, in some cases, help limit the impact of DoS attacks, but usually only at points where the DoS attack is consuming fewer resources than are available. In many cases, the only defense is a reactive one where the source or sources of an ongoing attack are identified and prevented from continuing the attack. The use of source IP address spoofing during attacks and the advent of distributed attack methods and tools have provided a constant challenge for those who must respond to DoS attacks.

Early DoS attack technology involved simple tools that generated and sent packets from a single source aimed at a single destination. Over time, tools have evolved to execute single source attacks against multiple targets, multiple source attacks against single targets, and multiple source attacks against multiple targets.

Today, the most common DoS attack type reported to the CERT/CC involves sending a large number of packets to a destination causing excessive amounts of endpoint, and possibly transit, network bandwidth to be consumed. Such attacks are commonly referred to as packet flooding attacks. Single source against single target attacks are common, as are multiple source against single target attacks. Based on reported activity, multiple target attacks are less common.

The packet types used for packet flooding attacks have varied over time, but for the most part, several common packet types are still used by many DoS attack tools.

TCP floods – A stream of TCP packets with various flags set are sent to the victim IP address. The SYN, ACK, and RST flags are commonly used.

ICMP echo request/reply (e.g., ping floods) – A stream of ICMP packets are sent to a victim IP address.

UDP floods – A stream of UDP packets are sent to the victim IP address.

Because packet flooding attacks typically strive to deplete available processing or bandwidth resources, the packet rate and volume of data associated with the packet stream are important factors in determining the attack's degree of success. Some attack tools alter attributes of packets in the packet stream for a number of different reasons.

Source IP address – In some cases, a false source IP address, a method commonly called IP spoofing, is used to conceal the true source of a packet stream. In other cases, IP spoofing is used when packet streams are sent to one or more intermediate sites in order to cause responses to be sent toward a victim. The latter example is common for packet amplification attacks such as those based on IP directed broadcast packets (e.g., "smurf" or "fraggle").

Source/destination ports – TCP and UDP based packet flooding attack tools sometimes alter source and/or destination port numbers to make reacting with packet filtering by service more difficult.

Other IP header values – At the extreme, we have seen DoS attack tools that are designed to randomize most all IP header options for each packet in the stream, leaving just the destination IP address consistent between packets.

Packets with fabricated attributes are easily generated and delivered across the network. The TCP/IP protocol suite (IPv4) does not readily provide mechanisms to insure the integrity of packet attributes when packets are generated or during end-to-end transmission. Typically, an intruder need only have sufficient privilege on a system to execute tools and attacks capable of fabricating and sending packets with maliciously altered attributes.

In June of 1999, multiple source DoS, or DDoS, tools began to be deployed. It is from that point in time forward that we evaluate trends in DoS attack technology. Though the focus of this paper is the continuing evolution of DoS attack technology, it is important to note that older tools are still successfully employed by intruders to execute DoS attacks.

3 Timeline

What follows is a brief timeline to highlight some of the major trend events in attack technology evolution. A more granular timeline is required to capture all trend events since July 1999, but that is not the purpose here. For our purposes, we are only interested in a timeline that highlights trends associated with widespread Internet activity based on reports received by the CERT/CC.

1999

July

Widespread deployment of DDoS networks based on tools like 'trinoo' and 'Tribe Flood Network' via various RPC related vulnerabilities. Many of the initial deployments were done manually, with intruders carefully testing for and selecting hosts positioned with high bandwidth availability.

DDoS networks used classic handler/agent control topology with direct communication via custom TCP, UDP, and ICMP protocols. Packet flooding attacks used UDP floods, TCP SYN floods and ICMP echo request floods.

DDoS networks were linked together with hard-coded handler lists in the agents, and with local files at the handler containing agent IP addresses.

DDoS agents listened for inbound commands from the handler. IDS signatures and network scanners were able to detect the presence of these types of DDoS agents on networks.

CERT® Incident Note IN-99-07
Distributed Denial of Service Tools
http://www.cert.org/incident_notes/IN-99-07.html

August

Stacheldraht DDoS tool found in isolated incidents. Stacheldraht combined features of 'trinoo' and TFN and added encrypted communications between the attacker and the stacheldraht handlers. Stacheldraht also provided for automated update of agents.

Again, deployment involved selective targeting based on the packet generating capability of the target systems.

November

CERT/CC sponsored the DSIT Workshop, which resulted in a paper published in December describing the threats posed by DDoS attack technology.

> Results of the Distributed Intruder Tools Workshop
> http://www.cert.org/reports/dsit_workshop-final.html

December

Release of Tribe Flood Network 2000 (TFN2K). It included many features designed to make TFN control and attack traffic more difficult to detect and trace on a network.

Intruders had to work hard to deploy large DDoS attacks networks; much work was done to avoid detection and compromise of deployed attack networks and to provide for easier maintenance.

> CERT Advisory CA-1999-17
> Denial of Service Tools
> http://www.cert.org/advisories/CA-1999-17.html

2000

January

Stacheldraht becomes widely used after several months of underground development.

> CERT® Advisory CA-2000-01
> Denial of Service Developments
> http://www.cert.org/advisories/CA-2000-01.html

February

The mainstream media reported on the now-infamous February 2000 DDoS attacks that targeted several high-profile web sites.

April

Packet amplification attacks using nameservers became popular.

CERT® Incident Note IN-2000-04
Denial of Service Attacks using Nameservers
http://www.cert.org/incident_notes/IN-2000-04.html

DDoS tool ‘mstream’ found in the wild. It used a network topology similar to ‘trinoo.’ The attack payload used TCP ACK packets with randomized source information and a randomized destination port.

CERT® Incident Note IN-2000-05
“mstream” Distributed Denial of Service Tool
http://www.cert.org/incident_notes/IN-2000-05.html

May

VBS/LoveLetter outbreak further demonstrated the widespread success and impact of social engineering attacks based on malicious email attachments.

CERT® Advisories CA-2000-04
Love Letter Worm
http://www.cert.org/advisories/CA-2000-04.html

t0rnkit had a widespread impact and evolved to be used to deploy existing DDoS tools.

CERT® Incident Note IN-2000-10
Widespread Exploitation of rpc.statd and wu-ftpd Vulnerabilities
http://www.cert.org/incident_notes/IN-2000-10.html

August

The Trinity DDoS tool was deployed on compromised unix systems and was an early adopter of IRC as the core DDoS network control infrastructure.

November

Multiple Windows-based DDoS agents were actively deployed. These tools marked a shift from unix to Windows as an actively used host platform for DDoS agents.

2001

January

Ramen worm improved intruder tool distribution by automating propagation across hosts using a back-chaining model.

CERT® Incident Note IN-2001-01
Widespread Compromised via “ramen” Toolkit
http://www.cert.org/incident_notes/IN-2001-01.html

February

VBS/OnTheFly (Anna Kournikova) email attachment outbreak once again demonstrated the widespread impact of social engineering attacks.

CERT® Advisory CA-2001-03
VBS/OnTheFly (Anna Kournikova) Malicious Code
http://www.cert.org/advisories/CA-2001-03.html

The erkms and li0n worms were used to deploy DDoS tools via BIND vulnerabilities.

CERT® Incident Note IN-2001-03
Exploitation of BIND Vulnerabilities
http://www.cert.org/incident_notes/IN-2001-03.html

April

DDoS tool carko found in the wild. It was very similar to previously known variants of stacheldraht.

CERT® Incident Note IN-2001-04
"Carko" Distributed Denial-of-Service Tool
http://www.cert.org/incident_notes/IN-2001-04.html

May

The cheese worm spread as an attempted “patch worm” to remove backdoors installed by other attacks.

CERT® Incident Note IN-2001-05
The “cheese” Worm
http://www.cert.org/incident_notes/IN-2001-05.html

The w0rmkit worm propagated slowly, targeting previously compromised systems using well-known intruder backdoors.

The sadmind/IIS worm began to propagate by targeting two separate vulnerabilities on two separate operating system platforms.

CERT® Advisory CA-2001-11
sadmind/IIS Worm
http://www.cert.org/advisories/CA-2001-11.html

July

W32/Sircam email attachment outbreak demonstrates social engineering is still widely effective.

CERT® Advisory CA-2001-22
W32/Sircam Malicious Code
http://www.cert.org/advisories/CA-2001-22.html

More sophisticated worms began to propagate, including Leaves and Code Red. Leaves incorporated the ability to update and change functionality during propagation. Code Red included functionality to launch a TCP SYN DoS attacks against a specific target. Code Red also caused isolated DoS conditions due to high scanning and propagation rates.

CERT® Incident Note IN-2001-07
W32/Leaves: Exploitation of previously installed SubSeven Trojan Horses
http://www.cert.org/incident_notes/IN-2001-07.html

CERT® Incident Note IN-2001-08
"Code Red" Worm Exploiting Buffer Overflow In IIS Indexing Service DLL
http://www.cert.org/incident_notes/IN-2001-08.html

CERT® Advisory CA-2001-19
"Code Red" Worm Exploiting Buffer Overflow In IIS Indexing Service DLL
http://www.cert.org/advisories/CA-2001-19.html

CERT® Advisory CA-2001-20
Continuing Threats to Home Users
http://www.cert.org/advisories/CA-2001-20.html

CERT® Advisory CA-2001-23
Continued Threat of the "Code Red" Worm
http://www.cert.org/advisories/CA-2001-23.html

Several worms deployed IRC-based DDoS tools by exploiting a vulnerability in telnetd.

CERT® Advisory CA-2001-21
Buffer Overflow in telnetd
http://www.cert.org/advisories/CA-2001-21.html

August

Code Red II began to propagate much like the earlier Code Red.

CERT® Incident Note IN-2001-09
"Code Red II:" Another Worm Exploiting Buffer Overflow In IIS Indexing Service DLL
http://www.cert.org/incident_notes/IN-2001-09.html

CERT® Incident Note IN-2001-10
"Code Red" Worm Crashes IIS 4.0 Servers with URL Redirection Enabled
http://www.cert.org/incident_notes/IN-2001-10.html

Various IRC-based DDoS agents gained widespread use, including Knight/Kaiten, which has been found wrapped in a self-propagating worm.

September

Nimda worm outbreak began. Nimda combines attacks via email attachments, SMB networking, backdoors from previous attacks, exploitation of an Internet Explorer vulnerability, and exploitation of an IIS vulnerability to propagate widely. Like Code Red, propagation causes isolated DoS conditions.

CERT® Advisory CA-2001-26
Nimda Worm
http://www.cert.org/advisories/CA-2001-26.html

4 Trends

To discuss recent and emerging trends in DoS attack technology, we divided the issue into three distinct elements centered on the technology involved with the deployment, use, and impact of DoS tools.

Deployment

Deployment is an area of attack technology that has seen considerable change since 1999. As previously mentioned, DoS attack tools are commonly deployed on compromised systems. This deployment depends on the presence of exploitable vulnerabilities on systems and the ability of intruders to exploit those vulnerabilities. We have seen an increase in the sophistication and use of automated attacks, the use of blind targeting, and selective targeting of Windows-based systems and routers. We have also seen a significant decrease in the time window from when a vulnerability is discovered to when it is widely exploited.

Automation

Historically, like most attack tools, intruders often installed DoS tools onto compromised systems using mostly manual means. Over time, intruders have developed and employed a higher degree of automation in multiple aspects of DoS attack technology deployment.

Widespread scanning to identify victim systems was the initial phase of automation most often employed by intruders. Earlier scanning tools produced lists of potentially vulnerable hosts. The next step was the addition of automated tools to attempt exploitation of potentially vulnerable hosts and record lists of compromised hosts. Both types of lists were, and often still are, used by intruders to exploit vulnerable systems and install attack tools.

In particular, we still see intruder tools that execute packet amplification attacks using lists of networks that are known to respond to IP directed broadcast packets. We also see intruders remotely execute packet flooding attacks from Microsoft Internet Information Server (IIS) systems using lists of hosts that are vulnerable and will allow remote HTTP requests to execute arbitrary commands.

More recently, intruders have developed and employed tools that utilize scripts to automate scanning, exploitation, and deployment. T0rnkit was perhaps one of the most successful examples of this class of tools. This type of automated deployment is singular in depth, meaning the attacks do not propagate to additional systems beyond the initially attacked systems without manual intervention by an intruder.

Beginning with the ramen worm, we have seen a movement toward tools that automate scanning, exploitation, deployment, and propagation. Such tools are actively being used to deploy DoS attack tools.

Automated propagation has taken form using three general propagation models.

- Central source propagation – The mechanism used to compromise a system executes an instruction to transfer a copy of the attack toolkit from a central location to the newly compromised system. Scripts then control the automated installation of the tools and initiation of another attack cycle. File transfer mechanisms commonly use HTTP, FTP, and RPC protocols. The 1i0n worm used central source propagation.

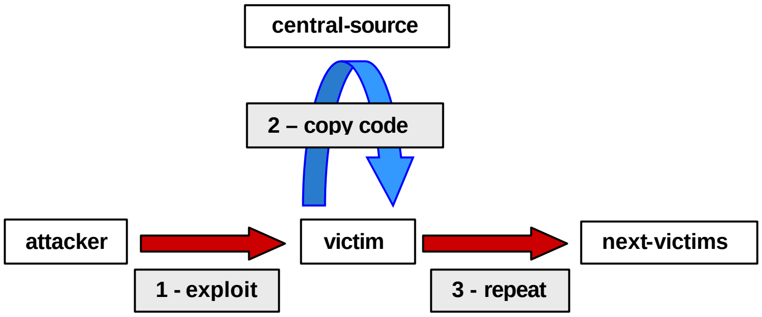

Figure 1 – Central source propagation

- Back-chaining propagation – The mechanism used to compromise a system executes an instruction to transfer a copy of the attack toolkit from the attacking host. For this to work, the attack tools on the attacking host include some method to accept a connection from and send a file to the victim host. We have seen simple port listeners that copy file contents across the network, full intruder-installed web servers, and the TFTP protocol used to support the back-channel file copy. The advantage of back-chaining propagation is it is more survivable than central source propagation; there is no single point of failure. The ramen worm used back-chaining propagation.

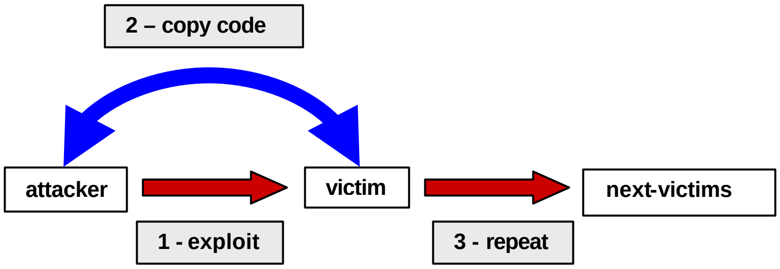

Figure 2 – Back-chaining propagation

- Autonomous propagation – Code Red, and the Morris worm of 1988 before it, are examples of autonomous propagation. The exploitation method includes injection of attack instructions directly into the processing of the victim host, causing the attack cycle to initiate again without any file retrieval from an external source.

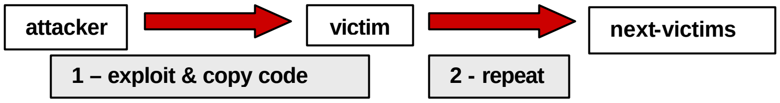

Figure 3 – Autonomous propagation

We have intentionally excluded the various email attachment “worms” from these models because they generally require some degree of human interaction to cause an attack cycle to initiate. We have seen such attacks grow in sophistication in terms of what the email attachment is capable of doing once executed, but the basic nature of the attack itself is still largely a social engineering attack and does not represent an area of significant technological advancement. Having said that, previous and recent successes of such attacks have demonstrated that security policies should not discount the effectiveness and threat posed by email attachment attacks in general. The potential certainly exists for such social engineering attacks to be used to deploy DoS tools on a widespread basis, but we have yet to see such a method develop into a real-world trend.

Windows-based Attack Targets

Automated attacks have historically targeted and leveraged vulnerabilities in unix-based operating systems, both professionally and end-user administered. Widespread attacks on Windows-based systems have historically employed some degree of social engineering to be successful. But more recently we've seen an increase in the use of Windows-based operating systems, related vulnerabilities, and end-users being targeted for remote exploitation of vulnerabilities and the deployment of DoS tools. We will discuss this trend based on two elements: blind targeting and selective targeting.

Recent self-propagating worms such as Code Red, Code Red II, and Nimda have used a blind targeting model, where target selection has been largely random with, at most, an emphasis on local or neighboring network block selection. These types of tools carry forward the basic random number generation algorithms used by many earlier widespread scanning and exploitation tools and are now actively being used against both unix-based and Windows-based systems.

Blind targeting attacks are usually highly automated and involve little human interaction during the execution of the attack. They also tend to be highly vulnerability-specific, often targeting systems that are vulnerable to one or a small number of particular exploitations. As such, the operating system platform or type of software on a system, which influences the presence of potentially exploitable vulnerabilities, often dictates a subset of Internet connected systems which can potentially fall victim to attacks. Other criteria are less central to the design and success of attacks based on blind targeting.

Attacks based on selective targeting may or may not incorporate high degrees of automation and vulnerability-specificity. Selective targeting is generally based on using some criteria other than the target operating system or potentially exploitable vulnerabilities to select a target or target sector for attack. Early DDoS tools, for example, were installed on carefully selected unix-based hosts. Systems were often manually tested for network connectivity, regular levels of network traffic, and available bandwidth before being used as handlers or agents in a DDoS network.

Today, intruder deployment efforts tend to pay less attention to target selection criteria. However, we have seen a trend toward Windows end-users being increasingly targeted both blindly and selectively. Through the typical model of intruder code re-use and evolutionary development, the intruder tools that target Windows systems have matured to the point where more advanced exploit technology for Windows-related vulnerabilities is enabling a wider array of Windows-based intruder tools.

There is a perception that Windows end-users are generally less technically sophisticated, less security conscious, and less likely to be protected against or prepared to respond to attacks than various other Internet populations such as professional system and network administrators. It is not our goal to prove a degree of truth to that perception, but we do take the liberty of asserting enough truth to the perception to provide a potential reason for the effectiveness of intruders specifically targeting Windows end-users.

In some cases, large populations of Windows end-users are relatively easy to identify. For example, it is not difficult to identify network block ranges for Internet Service Providers with known, large Windows end-user populations. Based on reports we have received, intruders are leveraging easily identifiable network blocks to selectively target and exploit Windows end-user systems.

Because of the increased targeting of Windows end-users, the CERT /CC published a tech tip entitled "Home Network Security" in July of 2001, and issued a related CERT® Advisory to raise awareness.

> Home Network Security
> http://www.cert.org/tech_tips/home_networks.html
>
> CERT® Advisory CA-2001-20
> Continuing Threats to Home Users
> http://www.cert.org/advisories/CA-2001-20.html

One common piece of advice to Windows end-users is to use personal firewall technology, either software or hardware-based, to protect their systems from external attack. It is important to note that technologies such as virtual private networks (VPN) may enable personal firewall technologies to be entirely bypassed by intruders. For example, end-users connected to America Online (AOL) over a DSL or cable modem connection may be assigned an IP address from an AOL network block in addition to the IP address obtained as a result of the DSL or cable modem connection. Traffic to the AOL-assigned address may be routed across a VPN to the end-user system in a way that may bypass some personal firewall technology, enabling intruders to remotely exploit vulnerabilities or misconfigurations such as unprotected file shares. We use AOL as an example due to its known, large Windows end-user population and it's well-known network block ranges, and have recorded incidents of Code Red and

Nimda propagation impacting AOL-connected hosts via VPN addressing. Other implementations of VPN technology, such as those deployed to provide enterprise or campus remote access, are also subject to remote attack that may bypass personal firewall technology. The security policy of the controlling end of the VPN will dictate the exposure of the VPN client system. In the case of an ISP, the security policy typically allows most all traffic to pass to the client, which is a point end-users should consider when protecting their systems.

Selective Targeting of Routers

One of the most recent and disturbing trends we have seen is an increase in intruder compromise and use of routers. We have received reports of intruders using vendor-supplied default passwords on poorly configured and deployed routers to gain unauthorized access to and control of routers. Several publicly available documents are available to provide novice intruders with a set of basic advice and commands to execute after compromising a router in order to modify the router's configuration. Reports indicate routers are being used by intruders as platforms for scanning activity, as proxy points for obfuscating connections to IRC networks, and as launch points for packet flooding DoS attacks.

Routers make attractive targets for intruders because they are generally more a part of the network infrastructure than computer systems and thus may be "safer" in the face of attacks from rival intruders. Additionally, routers are often less protected by security policy and monitoring technology than computer systems, enabling intruders to operate with less chance of being discovered.

Of extreme concern is the potential of routers being used for DoS attacks based on direct attacks against the routing protocols that interconnect the networks comprising the Internet. We believe this to be an imminent and real threat with a potentially high impact. Routing protocol attacks are being actively discussed in some intruder circles and have become agenda items at public conferences such as DefCon and Black Hat Briefings.

Time-To-Exploit Is Shrinking

Exacerbating the sophistication of attacks and the abundance and susceptibility of targets is a shrinking time-to-exploit. The window of opportunity between vulnerability discovery and widespread exploitation, when security fixes or workarounds can be applied to protect systems, is narrowing. This is, in part, due to the large existing code-base of attack tools than can be used to develop new tools as exploits are written for newly discovered vulnerabilities. Another element causing this trend is a trend toward non-disclosure within intruder communities. Rival groups will often keep new exploits and attack tools private to gain some advantage over other rival groups. Tools that are exposed to outside groups often become obsolete through competitive analysis and are quickly modified, making the lifetime of many attack tools very short. Anti-forensics techniques are now commonly employed in the design of intruder tools in an attempt to increase the lifetime of the tools by limiting the ability of others to determine the function of and defense against an attack tool. Thus, when public awareness of an exploit method or attack tool does rise, the method or tool is often already in some degree of widespread use.

Use

As previously mentioned, we continue to see DoS attacks launched using older single source and multiple source attack tools. However, we have seen some notable trends emerge in the development and use of DoS tools by intruders.

Control Channels

The early DDoS attack tools used networks of intruder controlled handlers that were used to send attack commands to an array of agents. The agents would then launch packet flooding attacks against victim sites. The communication channels between the intruder and the handler were generally such that the handler would listen for connections from the intruder and accept commands across the network. Likewise, the communication channels between the handler and the agents generally involved two communication channels. The handlers would listen for packets from the agents to allow the agents to register their IP address with the handlers. The agents then listen for commands from the handler. Communication channels were typically assigned to fixed and non-standard service port numbers.

For example, the trinoo DDoS tool used the following service ports for communications:

intruder → handler; destination port 27665/tcp

handler → agents; destination port 27444/udp

agents → handlers; destination port 31335/udp

Other tools, such as Stacheldraht, incorporated encryption technology into the communications channels in an attempt to better conceal the DDoS attack network.

The early design of DDoS network tools caused DDoS networks to be relatively easy to identify and disrupt. The agents had to maintain a list of one or more handlers, usually done via hard coded IP address lists, and send packets to register themselves with the handlers. Thus, intercepting an agent typically led to identification of the handler. The handlers had to maintain a list of agents to contact for attack initiation, so discovery of a handler usually led to identification and disruption of an entire DDoS network. Because handlers and agents typically listened for connections, it was possible to use network scanners to locate and identify handlers and agents. Also, the custom communications protocols used between the intruder and the handler, and the handler and the agent, were relatively easy to identify using network monitoring tools such as Intrusion Detection Systems (IDS).

The deficiencies in older DDoS tool design perhaps contributed to them not being widely used to actually execute DoS attacks. Deployment of these types of DDoS networks is time consuming, even with automated deployment techniques, and discovery of a single node often led to the demise of the entire attack network. As a result, we have observed more deployment activity than actual use of such DDoS attack tools.

Recently, we've seen an increase in intruder use of Internet Relay Chat (IRC) protocols and networks as the communications backbone for DDoS networks. The use of IRC essentially replaces the function of a handler in older DDoS network models. IRC-based DDoS networks are sometimes referred to as "botnets," referring to the concept of "bots" on IRC networks being software-driven participants rather than human participants.

The use of IRC networks and protocols makes it more difficult to identify DDoS networks. IRC networks and protocols allow DDoS agents placed on compromised systems to establish outbound connections to a standard service port (e.g., 6667/tcp) used by a legitimate network service. Agent communications to the control point may not be easily discernable from other legitimate network traffic. And, the agents do not incorporate a listening port that is easily detectable with network scanners. An intruder can establish a connection to the IRC server, again using legitimate communications channels, to control an array of DDoS agents. Security policies that control outbound access to standard IRC-related ports (e.g., 6660/tcp through 6669/tcp) may be able to detect and prevent unauthorized connections, but the popularity of IRC services, especially among end-user populations, means that such access controls are not widely implemented in security policies.

IRC networks and protocols also offer greater survivability for DDoS network use. The IRC server tracks the addresses for connected agents and facilitates communication between the intruder and the agents. The need for custom protocols and local tracking of agents is eliminated. Thus, discovery of a single agent may lead no further than the identification of one or more IRC servers and channel names used by the DDoS network. From there, identification of the DDoS network depends on the ability to track agents currently connected to the IRC server.

For public IRC networks, such as Efnet, Undernet, or DALnet, removing an IRC server to disable a DDoS network is not a realistic option. Thus, use of public IRC networks has the advantage of providing a more stable communications infrastructure for DDoS networks. On the other hand, public IRC networks do, to some degree, expose DDoS networks and agent locations to external identification by security teams who are able to respond in some capacity. So, intruders are also using private IRC servers to serve as the communications backbone for DDoS networks.

In some cases, we have seen use of bogus domain names registered and deployed explicitly to serve as a mechanism to direct agent connection points back to IRC servers. Such domain names have been seen registered using obviously false contact information in the public WHOIS databases. These “floating” domain names enable intruders to control agent connection points by reconfiguring the A record for a DNS name.

Some IRC-based DDoS agents also include the capability for an intruder to move the agent connection point by issuing a command to the agents. In other words, remote reconfiguration is being built into DDoS agents to aid intruder management of the DDoS networks. Regardless of that ability, it is trivial for intruders to alter the connection point in agent code and quickly redeploy DDoS agents that connect to a different IRC control point. As such, IRC-based DDoS networks tend to be largely compromised of expendable agents, that when discovered, do not compromise or greatly impact the effectiveness of the DDoS network.

Executing DoS Attacks

We have seen little change in the nature of the targets of DoS attacks. The Internet community, ranging from individual end-users to the largest organizations, continues to experience DoS attacks. What we have seen is a steady increase in the ability for intruders to easily deploy large DDoS attack networks. In the race of available consumable resources versus the ability to consume those resources, today’s DDoS networks continue to outpace available bandwidth in most cases.

Where packet filtering or rate limiting can be effective to control the impact of some types of DoS attacks, intruders are beginning to more often use legitimate, or expected, protocols and services as the vehicle for packet streams. Doing so makes filtering or rate limiting based on anomalous packets more difficult. In fact, filtering or rate limiting an attack that is using a legitimate and expected type of traffic may in fact complete the intruder’s task by causing legitimate services to be denied.

Although it is still used, we have noticed less emphasis on source IP address spoofing in DoS attacks. With highly distributed attack sources, that many times cross several autonomous system (AS) boundaries, the number of hosts involved as sources of an attack can be simply overwhelming and very difficult to address in response. Source IP address spoofing simply isn't a requirement to obfuscate large numbers of attack sources and enable the attacking party to avoid accountability for the attack.

Impact

Increased Blast Zone

In general, the impact of DoS attacks depends on the ability of the attack to consume available resources. As we've previously mentioned, today's attack methods and tools place even the most abundant resources in range for disruption. What we have seen is an increase in collateral damage, that is, damage not directly associated with the consumption of the target resource or resources.

For example, the adoption of security monitoring technology, or even basic service activity logging, causes a good deal of log information to be created on the typical network. We are aware of instances where large increases in activity related to security events such as Code Red or Nimda have created problems for backup systems due to sudden increases in log file volumes.

There are many Internet sites that interconnect with other networks, typically upstream networks, based on the provisioning of measured use circuits. That is, the cost of the bandwidth is to some degree based on how much bandwidth is actually used. DoS attacks and large increases in traffic as a result of security events can have direct financial impact by causing traffic levels and circuit costs to be raised.

Consolidation and outsourcing of hosted services has led to DoS attacks against a single element of a hosting infrastructure impacting multiple elements. For example, an attack against one website within a server farm has been known to impact many other websites by virtue of their network proximity to the attack target.

When Deployment Becomes the Attack

Very recent security events such as Code Red, Code Red II, and Nimda have demonstrated that the deployment phase of a highly automated attack tool can itself become a broad DoS attack against many parts of the global Internet. The scanning and propagation activity itself does not present a large problem. And in the instance of Code Red, the intended DoS attack was subverted. Collateral damage seemed to be the majority of the problem. In addition to the collateral damage issues previously discussed, networks with relatively high numbers of infected and active sources quickly became saturated due to address resolution protocol (ARP) storms caused by the worms' rapid scanning activity. This in itself caused locally isolated denials of service. There were also various networked devices such as printers and DSL modems that were unexpectedly impacted by Code Red and Nimda. In other cases, reactions to news of the widespread propagation of these worms caused some Internet sites to simply disconnect from the Internet entirely. This in effect achieves a DoS attack against those who chose to protect internal resources at the expense of Internet connectivity.

5 Summary

At the core, the problem of denial of service on the Internet has not significantly changed in recent years. Network resources remain limited and susceptible to consumption attacks, and systems still contain vulnerabilities, new and old, that either remain un-patched or are patched in a less than timely manner. Vendors continue to produce technology products that contain exploitable security vulnerabilities. Consumers continue to deploy technology products that contain security vulnerabilities, are misconfigured such that compromise is possible, or are simply insecurely managed. The CERT/CC continues to record vulnerability and exploit lifecycles lasting two to three years despite security community and vendor efforts to raise awareness of serious security issues. The end result is there are still plenty of vulnerable systems on the Internet that can be used as launch points for DoS attacks.

DoS attack payloads have changed little since 1999. Most bandwidth DoS attack tools employ well-known, and well-used, types of packet streams to achieve their goals. There is really little incentive for intruders to improve on old DoS flooding algorithms because the old attack methods still work quite well. At an extreme, the types of packet streams which can exist is limited because there are a limited number of permutations in packet parameters that exhibit unique attack characteristics and most combinations have been tried.

What is changing, however, is the state of intruder tool deployment technology and methods, the design and control of DDoS networks, and the impact of DoS attacks.

Automation technology has enabled self-propagating worms to become common. We have seen an increase in the number of highly automated and vulnerability-specific attacks based on blind targeting. Selective targeting has shifted to place Windows end-users and, more significantly, the routing infrastructure of the Internet at greater risk.

While we still see the occasional deployment and use of older, more traditional DDoS attack networks, intruders are moving toward the use of IRC networks and protocols as the control infrastructure, or handler, for DDoS attack agents. Packet flooding streams continue to be comprised of well-known packet types, and attack tools continue to combine multiple types of packet streams as attack options.

As DoS attacks increase in potential and real impact, collateral damage has also increased in numerous ways. Automation has reached the point where attack tool propagation can by itself become a DoS attack.

6 Conclusion

Evolution in intruder tools is a long-standing trend and it will continue. And, DoS attacks by their very nature are difficult to defend against and will continue to be an attractive and effective form of attack. Automation of attack tool deployment and ease of management will continue to be areas of focused evolution for DoS tools. It is also likely, at least in the short term, that advancements in DoS attack technology will take shape in the form of protocol-specific attacks, such as attacks on routing protocols, rather than as significant innovations in basic characteristics of packet flooding streams.

While we do not propose solutions for the issues discussed in this paper, it is important to recognize and understand trends in attack technology in order to effectively and appropriately evolve defense and response strategies. We encourage Internet sites to carefully consider the trends we have discussed and evaluate how security policies, procedures, and technologies may need to change to address the current trends in DoS attack technology.