Global Information Assurance Certification Paper

NAPTHA: A new type of Denial of Service Attack
Brandi Copans
12/10/00

The Razor security team, a group that researches security vulnerabilities at BindView Corporation, has released information on a new group of Denial of Service vulnerabilities called NAPTHA. The tool designed to implement this attack is also called NAPTHA. NAPTHA has the capability to issue an asymmetrical attack that exploits vulnerabilities in TCP protocol. The end result of this attack is resource starvation on the effected system (3). CERT believes NAPTHA is dangerous because it can be implemented asymmetrically, can be done anonymously, and can be implemented as a Distributed Denial of Service attack (6).

Denial of Service Attacks

Denial of Service(DoS) attacks are designed to disrupt normal service of their victim. This goal is achieved by consuming resources, destroying or altering configurations, or causing the physical destruction of the target machine (2). For example, an attacker could unplug a machine, tie up CPU cycles of a machine by sending it thousands of requests, or flood a network with packets from several thousand machines. All of these actions would result in a Denial of Service because the target would not be able to service normal requests.

DoS attacks that are damaging enough to make headlines are usually Distributed Denial of Service (DDoS) attacks. The attacks experienced by Yahoo.com and Amazon.com in February of 2000 are examples of DDoS attacks (4). A Distributed Denial of Service attack is a Denial of Service attack launched simultaneously from various hosts on different networks. There are several software packages available to coordinate all of the participating hosts in the launch of a synchronized attack. These packages are installed on machines enlisted for the attack enabling the attacker to give commands from one location to which all hosts will respond. Because the DDoS is coming from several to thousands of machines, it is difficult to determine who masterminded the attack. NAPTHA can be implemented as a DDoS and it can be done anonymously.

Whether the attack is distributed or not, it is important to compare the number of resources required to cause the desired damage. An attack is asymmetrical when it requires a small amount of resources to cause a great amount of damage. This ratio is desired by an attacker, which makes successful asymmetrical attacks very dangerous. NAPTHA is an asymmetrical attack.

DoS attacks are designed around known weaknesses in certain applications or protocols. Examples of DoS attacks are Mail Bombs, Ping of Death, and SYN flood. A Mail Bomb sends more email to a mail server than the application can handle. This eventually causes the application to crash. Ping of Death exploits the ability of Internet Protocol to fragment packets. Pings are crafted to be greater than the allowed 65,535 bytes. Internet Protocol chops up the ping into several packets that are re-assembled into one packet at

the receiving end. When the machine attempts to handle the large packet it usually crashes (10). A SYN flood exploits weaknesses in the TCP protocol and will be discussed in more detail below. NAPTHA is designed to exploit TCP protocol weaknesses.

An overview of TCP Protocol

A TCP connection normally progresses through a series of states: LISTEN, SYN_SENT, SYN_RECEIVED, ESTABLISHED, FIN_WAIT-1, FIN_WAIT-2, CLOSE_WAIT, CLOSING, LAST-ACK, TIME-WAIT, and CLOSED (4). If a conversation between two machines is initiated by a client, it sends a host a SYN packet and changes its state to SYN_SENT. This SYN packet includes:

- the client's IP address,
- desired port for connection on the host and the client,
- maximum segment size the network will allow,
- maximum buffer size for the client,
- initial sequence number and ending sequence number indicating how many bytes were sent,
- and a flag indicating it is a SNY packet.

The host responds to the SYN by sending a SYN,ACK and changes its state to SYN_RECEIVED. This packet includes the same information listed above, but this information pertains to the host. The SYN,ACK also includes the initial sequence number sent by the client. This number is used as an acknowledgement number. The client responds to the SYN,ACK with an ACK, acknowledging the SNY,ACK. This process essentially opens two connections: one between the client and the host and another between the host and the client. Both connections are now in the ESTABLISHED state.

Eventually, the connection will be closed by one of the parties. TCP has a set of states to close a connection gracefully that can be initiated by the client or the host. If the client terminates the session, it will send a FIN to the host and the client's state will change to FIN_WAIT 1. When the host receives the FIN, it will respond with an ACK and change its state to CLOSE_WAIT. When the client receives this ACK its state will change to FIN_WAIT 2. The client is now waiting for the host to close its connection. The host will close its connection by sending a FIN to the client. The state of the client will change to TIME_WAIT, essentially closing its connection to the host, and it will respond to the host will an ACK. The host will then close its connection to the client and change to the CLOSED state. It is also possible to abruptly close the two connections. One machine may send a RESET to the other which will cause both connections to be closed.

These states enable TCP to establish reliable connections between two machines. On each machine, the kernel of the operating system keeps track of each TCP state. An excessive amount of TCP states not being handled in the normal fashion will cause the machine to exhaust its CPU and RAM. Eventually the machine will run out facilities to handle all the connections. This situation is also known as a TCP state exploit (3).

Exploiting Vulnerabilities in the TCP Protocol

Some Denial of Service attacks exploit TCP states. A well-known example is a SYN flood attack. This attack exploits the way TCP handles a large number of connections that establish a SYN_RECVD state. The victim's machine is continually flooded with SYN packets. These packets contain the information discussed above including IP addresses that do not belong to the machine sending the SYN. Consequently, when the victim attempts to respond to the SYN, sending a SYN,ACK, the fake client does not reply. Eventually, the combination of having many SYN_RECVD states open and sending SYN,ACK replies to fake clients exhausts the resources of the victim. The origin of this type of DoS is difficult to detect because of the fake IP addresses. If this attack is launched as a DDoS, the effect can ripple across several locations as the fake IP addresses the victim sends SYN, ACK replies to can theoretically be live hosts.

Similar to a SYN flood attack, NAPTHA has introduced weaknesses in the way TCP handles a large number of connections in ESTABLISHED and FIN_WAIT-1 states. NAPTHA acts as a client, participating in the normal exchange of SNY, SYN,ACK, and ACK resulting in an ESTABLISHED state on the host. The host then waits for further data from the client. At the same time, the client is sending additional SYNs to create more ESTABLISHED states on the host. All of these SYNs are answered in the normal manner by NAPTHA until the ESTABLISHED state is reached.

NAPTHA can also exploit the way some applications handle the FIN_WAIT-1 state. If the host initiates closing the session, it will send a FIN to the client and change to the FIN_WAIT-1 state. If the client does not respond with an ACK, the host will remain in the FIN_WAIT-1 state, essentially keeping the connection open until it times out.

Razor's NAPTHA does not utilize a traditional TCP/IP stack. This enables it to proceed though the normal connection steps of TCP without the overhead of tracking all TCP states in the kernel of the operating system. According to the Razor team, it "responds to a packet sent to it based on the flags in that packet alone"(1). It has the ability to establish and respond to several thousand TCP connections without consuming a large amount of resources on the attacker's machine. NAPTHA can also be used in a DDoS attack working in concert with several machines allowing for anonymity of the attacker.

What can be done to prevent NAPTHA?

The Razor team has not released NAPTHA to the general public. If the NAPTHA tool is leaked, it carries a footprint inside its packets (a line of a B52's song) so it can be identified.
It is assumed that another party could design a program similar to NAPTHA so the vulnerabilities it introduces should be considered security threats. NAPTHA or an attack using similar methods cannot be easily detected because it may look like normal traffic occurring across a system. If there is an unusually large increase of ESTABLISHED or

FIN_WAIT-1 state connections on a machine, it may be an indication of this type of attack and should be investigated.

The Razor team has successfully documented vulnerabilities in Compaq's Tru64 Unix, FreeBSD, Linux 2.0 and 2.1 kernel based systems (including Red Hat 6.1 and Slackware Linux 4.0), HP-Unix, Windows 95,98,98SE and NT, Novel's Netware, SGI's IRX 6.5.7, and Sun's Solaris 7 and 8 (1). The only system free of vulnerabilities is Windows 2000. Each of these systems has different thresholds for the number of connections that can be established before the victim's resources are depleted. The Razor team tested several ports and applications across different operating systems. Results range from the system needing a complete reboot to the system refusing to accept new connections for a period of time.

Currently there are only a few specific solutions offered by vendors to combat this problem. Microsoft has released a patch for the NT server and a solution for Window's 95, 98 and 98E to combat the vulnerability in NetBOIS uncovered by NAPTHA. NetBOIS is a networking service used for PC networking. The vulnerability lies in the way NBT, the protocol standard for NetBOIS, handles the packets NAPTHA produces. Any attacker that has access to port 139 can exploit this vulnerability. The patch for Windows NT "eliminates the flaw in NBT"(7).

Users of the Windows Operating Systems effected by NAPTHA are instructed to disable the File and Print sharing services on their computer. This service usually runs on port 139, which may already be blocked if the machine in question lives behind a firewall. Users of Windows 98 may already be aware of security issues with this port, as Microsoft has included a default message that notifies the user "File and printer sharing is running on the TCP/IP connection you will use to access the internet. Other users on the Internet might be able to access your files"(8). Refer to Microsoft article Q199346 for more information on these settings.

FreeBSD, IBM, and SUN are working on fixes as of this writing. Compaq instructed users of Tru64 UNIX to implement tuning steps to increase the size of the queuing resources to cause timeouts on incomplete connections (6). These guidelines are based on the guidelines developed to prevent a SYN flood attack. They are available from Compaq's web site: http://tru64unix.compaq.com/.

The Razor security team recommends the following general steps to reduce the risk on your system:(3)

1. Limit the amount of services running on systems. Disable services that are not needed.
2. Limit access to systems where applicable.
3. Check firewall and router configuration to ensure ingress and egress filtering which may prevent spoofing.
4. Use inetd or tcpserver to limit spawned daemon processes. This may prevent

daemons from crashing a server and may allow it to recover. Processes running under inetd were less vulnerable on some systems (9).

5. Adjust TCP timeouts and keepalives settings, which will reduce the number of processes running at one time, and will keep processes recycling.

Further updates and information will continue to be documented on the BindView's website: http://razor.bindview.com/. Vendor specific comments, workarounds, and links to patches are also available on this site. NAPTHA has been assigned BUGTRAQ ID 2022, which can be used to track new information. Refer to Security Focus's website for more information on BUGTRAQ: http://www.securityfocus.com/bid/2022

References:

1. "The NAPTHA DoS Vulnerabilities 'Tested Products'". 2 December 2000. URL: Http://razor.bindview.com/publish/advisories/adv_list_NAPTHA.html (2 December 2000).

2. CERT Coordination Center. "Denial of Service Attacks". 12 February 1999. URL: http://www.cert.org/tech_tips/denial_of_service.html. (2 December 2000).

3. "The NAPTHA DoS vulnerabilities". 30 November 2000. URL: http://razor.bindview.com/publish/advisories/adv_NAPTHA.html. (2 December 2000).

4. NTA Monitor. "Denial of Service Attacks - Yahoo/ Amazon" 16 February 2000. URL: http://www.nta-monitor.com/news/yahoo.htm. (3 December 2000).

5. RFC: 793. "TRANSMISSION CONTROL PROTOCOL DARPA INTERNET PROGRAM PROTOCOL SPECIFICATION" September 1981. URL: http://www.faqs.org/rfcs/rfc793.html. (2 December 2000).

6. CERT Advisory CA-2000-21. "Denial-of-Service Vulnerabilities in TCP/IP Stacks". 30 November 2000. URL: http://www.cert.org/advisories/CA-2000-21.html. (2 December 2000).

7. Microsoft. "Microsoft Security Bulletin (MS00-091): Frequently Asked Questions". 29 November 2000. URL: http://www.mocrosoft.com/technet/security/bulletin/fq00-091.asp. (2 December 2000).

8. Microsoft. "Disable File and Printer Sharing for Additional Security". 20 October 2000. URL: http://support.microsoft.com/support/kb/articles/q199/3/46.asp". (3 December 2000).

9. "The NAPTHA DoS vulnerabilities 'Tested Products'". 2 December 2000 URL: http://razor.bindview.com/publish/advisories/adv_list_NAPTHA.html. (2 December 2000).

10. Computer Incident Advisory Capability. "H-12: IBM AIX® 'SYN Flood' and 'Ping of Death' Vulnerabilities" 10 December 1996. URL: http://ciac.llnl.gov/ciac/bulletins/h-12.shtml. (6 December 2000).