Global Information Assurance Certification Paper

The Naptha Denial-of-Service Vulnerabilities by Sven Peterson

History and Discovery

On November 30, 2000, CERT released an advisory regarding a set of TCP/IP vulnerabilities found in a variety of network operating systems. The vulnerabilities were originally researched and discovered by the BindView Razor security team and publicized as a potential denial-of-service problem. The specific type of attack outlined is unique in that the resources expended by the attacker do not match the resources consumed by the victim. Thus, an attacker would not necessarily need to dedicate a high-end computer with a fast connection to bring down the target system. The attack is also not merely a flood attack in which all of the available bandwidth is saturated, nor a SYN flood attack, which exploits how TCP stacks handle large numbers of connections in the SYN RECVD state. Instead, the Naptha exploit works by keeping many connections in the ESTABLISHED or FINWAIT_1 states. Although connections in these states eventually time out on the target system, they are created in a rapid succession from the attacking system – something an ordinary network application would not do, but which the rules of TCP do not prevent.

The Razor security team notified CERT of their discovery, and sent them sample program code demonstrating how the vulnerabilities could be exploited; CERT has contacted the vendors involved. So far, the products confirmed to be affected are all versions of Windows prior to Windows 2000, Novell Netware 5.0, Compaq Tru64 UNIX, FreeBSD 4.0, HP-UX, IRIX, Solaris, and several brands of Linux. The Razor team did not release the exploitation code to the public.

Vulnerability in Microsoft's Operating Systems

Microsoft has issued a security bulletin regarding the existence of the vulnerability in Windows 95, Windows 98, Windows 98 Second Edition, and Windows NT 4.0. A patch has been developed for systems running Windows NT 4.0 and is available at http://www.microsoft.com/Downloads/Release.asp?ReleaseID=25114. An alternate workaround for Windows NT 4.0 systems is to stop the server service. Although the problem has been acknowledged to happen with Microsoft's non-NT operating systems, Microsoft has advised work-arounds rather than issuing a patch; users of those operating systems are advised to disable file and printer sharing for systems connected directly to the Internet, or to implement a firewall that blocks traffic coming in on port 139.

The effect of the vulnerability, according to Microsoft, is that network services on a targeted system would temporarily cease as long as the attack was occurring. It is also possible that the attack would cause the operating system to halt entirely, requiring a reboot to regain functionality. The flaw itself is in how the implementation of the NetBIOS over TCP/IP protocol handles certain types of data packets. These packets would not result from common networking applications; an attacker would specifically need to write a program that created such packets. The Razor team found that the program would need to create a large number of connections to the target system and keep them in the FINWAIT_1 state.

Defending Against Naptha and Similar Denial-of Service Vulnerabilities

As a standard security system-hardening practice, any services that are not essential to the operation of a server running on the Internet should be disabled. For Windows systems, this especially applies to the server service on NT systems, and the File and Printer sharing service on Win9x systems. Unfortunately, many users of non-NT Windows operating systems may not be aware that these services may be running, and may not know how to disable them. At worst, though, these people may be subject to random attacks that require a reboot to recover from. Since it is not a buffer-overrun type of exploit, it is not possible for an attacker to run code on the target machine or take any action in the context of the user.

Different server configurations may handle a large number of connections differently; for example, Linux systems by default would continue to allocate memory to new connections indefinitely – memory which could not be paged out to the hard disk. Thus once the physical memory of the server were exhausted, the operating system would halt or become too slow to be usable. Other server operating system configurations may handle the problem by simply limiting the allowed number of incoming connections. Still, whether a legitimate user's connection is refused or just times out, the attack has been effective by stopping the server's intended network service.

Since the Naptha type of attack uses open TCP connections, the IP address of the attacker can at least be logged. A program running on a server could intelligently analyze connection statistics and, if an unusual amount of connections were sensed to be coming from a particular IP address, appropriate action could be taken. On a Linux or Unix system, for example, the action taken could be to automatically add the attacker's IP address to the hosts.deny file. Unfortunately, if the attacker used IP spoofing techniques while carrying out the attack, the wrong source's IP address would be incorrectly denied access. Further, the attacker may send a stream of spoofed packets in which the apparent source IP address changes with every packet.

A serious hacker planning to exploit one of the Naptha vulnerabilities would very likely use IP spoofing to cover his tracks, so methods of prevention should involve general measures taken to prevent IP spoofing in the first place. One of the best methods of preventing IP spoofing is to install a filtering router at the edge of the network. The router should be configured to not allow packets in to the external interface if the source address appeared to be from the internal network, was a broadcast address (255.255.255.255), or was one of the reserved IP address ranges (10.0.0.0 – 10.255.255.255, 127.0.0.0 – 127.255.255.255, 172.16.0.0 – 172.31.255.255, or 192.168.0.0 – 192.168.255.255). Additionally, the router should be configured not to allow outgoing packets that have a source address not originating from the internal network address space. If all ISPs implemented these filters at the router level, spoofing IP addresses – and thus the concealment of TCP-based denial of service attacks – would be much more difficult.

With a filter implemented at the router that dropped packets from invalid addresses, it would be wise for the network administrator to log dropped packets. This would provide a means of being alerted to attackers attempting to use spoofing techniques.

As we have seen, the Naptha attacks are yet another example of how TCP stacks

not handling large numbers of connections in a particular TCP state can cause a denial of service. Naptha is unique from other recently-discovered attacks in its asymmetric requirement of resources between the attacker and the target, but the general methods of prevention and defense are similar to the methods used against many of the common denial of service attacks.

References:

[1] CERT Advisory CA-2000-21 11/30/00
http://www.cert.org/advisories/CA-2000-21.html

[2] Razor Advisory: "The NAPTHA DoS vulnerabilities" 11/30/00
http://razor.bindview.com/publish/advisories/adv_NAPTHA.html
http://razor.bindview.com/publish/advisories/adv_list_NAPTHA.html

[3] Microsoft's Frequently Asked Questions about the Vulnerability 11/30/00
http://www.microsoft.com/technet/security/bulletin/fq00-091.asp

[4] Microsoft's Security Bulletin MS00-091 11/30/00
http://www.microsoft.com/technet/security/bulletin/ms00-091.asp

[5] Security Focus.com: "Microsoft Windows 9x / NT 4.0 NetBIOS over TCP/IP Resource Exhaustion Vulnerability", 11/30/00. http://www.securityfocus.com/vdb/?id=2022

[6] Stanislav Shalunov, "Netkill - generic remote DoS attack", 4/21/2000
http://securityportal.com/list-archive/bugtraq/2000/Apr/0152.html

[7] CERT Advisory CA-1996-21 "TCP SYN Flooding and IP Spoofing Attacks" 9/19/96
http://www.cert.org/advisories/CA-1996-21.html

[8] P. Ferguson, Network Working Group RFC 2827 – "Network Ingress Filtering: Defeating Denial of Service attacks which employ IP source address spoofing." http://www.ietf.org/rfc/rfc2827.txt