Global Information Assurance Certification Paper

Stop Port Scans with LaBrea

Jim McClurg
GSEC Practical version 1.2F
Octover 19th, 2001

Malevolent port scanning is one of the most common activities on networks today. The constant barrage of blind probes to our networks seems to never end. Is there any way we can stop them? Thanks to some clever work by Tom Liston and a group of IT security professionals, the answer is a definite yes!

Common Scanning Techniques

Scans encompass a wide range of purposes and techniques. Here are some of the more common ones.

Ping Sweeps

In order for an attacker to know his targets, he must acquire information about potential victims. When starting "cold", an attacker may perform an ICMP ping sweep. Using one of a variety of tools (such as fping, nmap, Pinger, Ping Sweep, WS_Ping ProPack, and Netscan tools), the attacker builds a table of IP addresses, along with information about those addresses – which were reachable? Which returned hostnames?

TCP/UDP port scans

Many times it is possible for an organization to block ICMP ping sweeps at the firewall. However, this may not stop the attacker. With or without the information from ICMP ping sweeps, TCP and UDP port scanning are possible if the IP range of a potential victim can be found. One common attacker technique is to send a crafted TCP/IP SYN/ACK packet to a range of IP addresses. By default, any computer receiving these unsolicited packets will immediately reply with a TCP/IP reset (RST). Bingo! This tells the attacker that there is a computer at that IP address! Now he knows the IP range of a potential victim. From that range of addresses, he knows which addresses are active and which are inactive. Of the active addresses, he knows which ports are listening on those computers.

Stack Fingerprinting

If an IP address is known, many port scanners are available that will send queries to TCP and UDP ports at that address. Some scanners query all common ports, and then determine the operating system. Knowing the IP address and the operating system will give an attacker a distinct advantage in future efforts.

Nmap is considered one of the best overall scanners ever. Another scanner is Queso, which sends various malformed packets to a single port on a server. From the replies, Queso can tell what operating system is running on the server. (Note that Queso is

especially effective because only *one* port is needed to discover the operating system!)

Scanning in Bulk – the Worms

Port scanning is a level of noise that many sites must endure. The tools are readily available and interest runs high in so-called "script kiddies" who run the tools for reasons ranging from curiosity to malice. However, in July 2001, scanning reached a new level. The Code Red worm quickly traversed the Internet, infecting hundreds of thousands of systems. The worm would scan random IP addresses for port 80. On successful location of a server, the worm would try to exploit the .IDA buffer overflow weakness of Microsoft's IIS web servers. Upon successful exploitation, the worm would start 100 threads on that victim server, looking for additional victims. The infection mushroomed quickly, as administrators scrambled to apply the Microsoft fix which had been announced almost a month before the worm appeared.

What Can an Attacker Do With the Information From Scans?

Worm Traversal

The most devastating results from worm scans are the widespread damage to information networks. Worm propagation can move at an exponential rate. But there are other uses for scanned information.

Vulnerability Detection

After some basic scanning, the attacker has the IP address and the operating system of the potential victim. This information is **the jackpot** to an attacker. According to John Pescatore, a Gartner analyst speaking at Symposium/Itxpo2001 [1], 90% of security breaches occur because the victim systems have not been patched in a timely manner, or the administrator has incorrectly configured the machine.

Now, if an attacker knows the IP address and the operating system, he can search various vulnerability databases for common weaknesses, and now he has a list of possible vulnerabilities on that server. Incidentally, the best information often comes from vendors and security sites that are trying to prevent attacks – www.microsoft.com, www.cert.org, www.vulnwatch.com, www.sans.org are some examples. Taken a step further, new announcements of vulnerabilities may be the *best* source of information, since it may take administrators some time to implement patches for these vulnerabilities.

IP Hijacking

Victim IP addresses can also be used for IP hijacking – altering existing TCP connections so that one or both ends are no longer communicating with their intended recipient. Generally more sophisticated attackers use IP hijacking. However, they must find valid IP addresses first, and port scanning may be used for that purpose. Worms or the more common scanning tools simply look for information on targets.

Spoofing

"Spoofing" is sending data to a recipient from a falsified IP address. It can be devastating when used in distributed denial of service attacks. An example is a "smurf" attack. A computer sends a ping to a network address containing many computers, using a crafted source IP address. This forged source IP address is actually the address of the intended victim. All computers (that received the original ping) reply to the apparent sender. The result is a massive wave of pings directed at the victim's IP address. The victim's server and network can be dragged to a standstill by the high traffic.

Denial of Service

Along the lines of the smurf attack, there are many denial-of-service attacks. These can be directed at specific ports, or simply at any IP address. A common one is called SYN flooding. Attacking computers send high numbers of TCP/IP SYN packets to victims. The victims then reply to these SYN packets, but never get an acknowledgement reply from the attacker. The victim will hold the half-connected TCP connection request in its buffer until a timer expires. If enough half-open TCP connections are established in a short amount of time, the victim will not be able to communicate any more!

What Can Be Done to Stop the Scans?

If the scans can be prevented or delayed, then the attacker loses time in his attack. For automated (worm) attacks, a stopped scan will stall or significantly slow the worm's progress.

How can you possibly avoid scans, or at least slow them down? There are standard firewall settings that can help, such as blocking ICMP pings or allowing very limited port traffic. Retaliatory port scans or other offensive moves can be illegal, landing a well-meaning administrator in severe trouble. (Many times retaliation is fruitless anyway. As was demonstrated with Code Red, many scanning machines were hapless victims of the worm.) "Honeypots" are fake systems designed to lure crackers, track their activities and thereby protect production systems. They can capture attackers as they interactively troll through systems, but port scans are usually of a wider breadth and frequency of attack.

Introducing LaBrea

A group of experts in network security developed an ingenious solution in the summer of 2001. As the Internet was being pounded by the Code Red Worm, Tom Liston wondered if he could come up with some way to slow these port scans. Since retaliation was not really an option, he *could* answer any of the scans that hit him *first*, and try to slow them down. He wrote some software that evolved over a few iterations, until he reached the current product, called LaBrea 2.0.

Normal TCP/IP and ARP Operation

A valid TCP/IP request will include a source IP address, source port, destination IP address, and destination port. The key point is that the destination IP address is known *before* a valid connection. Thus, there should never be a reason for an external computer to attempt a connection to an inactive IP address on a private network. Liston used this principle to determine that any computer trying to access unused IP addresses on his network was "fair game" to his form of passive retaliation.

Normally, when a TCP (SYN) connection request from the Internet arrives on a LAN, the router for the destination subnet will issue an ARP (broadcast to the entire subnet) to find the target computer. The target computer receives the ARP, and replies to the router. Now the router can store a pair of addresses in its ARP cache – it knows the target computer's IP and MAC addresses, so that any future packets sent to the target's IP address can be routed directly to the target's MAC address.

Once the target's MAC address is known, the router passes the SYN packet to the target. Upon receipt, the target computer replies to the initiator with a SYN/ACK packet. The initiator sends a final ACK (in reply to the target's SYN/ACK) to complete the TCP three-way handshake, and communication begins in earnest.

However, things change when the attacking computer sends a TCP connection request for an *unused* IP address. The router broadcasts an ARP, just like normal operation. However, if there is no reply to the ARP, the router will re-send the ARP broadcast multiple times. Eventually, when there is no reply, the router sends an ICMP "destination unreachable" message to the initiator. Eureka! The attacker has just discovered an unused IP address! The attacking scanner can quickly continue on, discovering all computers on the network very quickly.

The Mechanics of LaBrea

LaBrea provides the "tar pit" that stops these scans cold. The LaBrea software runs on a low-horsepower computer (Linux, a Trinux boot disk, or Windows NT), and listens for ARPS. When a router receives a request for an unused IP address, it broadcasts a series of ARPS to the subnet, waiting for a reply. LaBrea sees that there are multiple ARPS for an IP address, which means that there is no response from that target IP address. So, LaBrea assumes the identity of that IP address and answers for it! Now, the router has a MAC address for the target IP address, so it passes the packet to the LaBrea server.

Depending on the packet, one of several things will happen:

- If the packet is an ICMP ping, LaBrea will reply. This obviously negates the effectiveness of a ping sweep. All IP addresses reply!
- If the packet is a SYN/ACK packet, LaBrea will reply with a reset (RST) packet. This negates the SYN/ACK sweep – from the attacker's perspective, the subnet has no unused IP addresses – they all reply!

- Finally, if the packet is a SYN packet, LaBrea really becomes effective with its "tarpit":

    1. Originally (and still available), LaBrea would reply to a SYN packet with a SYN/ACK, and then ignore everything after sending that packet. This causes the attacker's connection to reply. In fact, this is a retaliatory SYN attack! Instead of the attacker issuing a single SYN and causing LaBrea to time out trying to complete the connection, LaBrea replies with a SYN/ACK, and forces the attacker into a timeout while it tries to finish the three-way handshake.
    2. With LaBrea 2.0, Liston incorporated the "persist" mode of TCP/IP. LaBrea replies to the initial SYN packet with a SYN/ACK, but it also sets the TCP window size very small (5-bytes). When the attacking computer replies with the first 5-byte packet, LaBrea replies with a TCP Window size set to 0, which is a wait status. The attacking computer has no choice but to send "probe" packets at steadily increasing intervals (up to four minutes for Microsoft Windows NT), waiting for the LaBrea server's TCP Window to increase. LaBrea dutifully continues to reply to these probes with a TCP Window of 0. The end result: The quick-hit scan is mired in the tarpit, with no theoretical exit.

Any attackers that are trapped in the "tarpit" are logged. This provides many benefits:

- Displays those nodes that are infected by worms. This would allow administrators to contact those infected sites so infected servers could be cleaned or removed. (In an ideal world, these logs could be consolidated centrally like www.dshield.org.)
- Allows historical information for incident handling.
- Lists information that may show holes in firewalls which can be repaired.

For more details on the available switches on the product, and for information on the latest releases, check www.hackbusters.net.

How effective could LaBrea be in the real world? According to Tom Liston [2]:

> At its peak, the Code Red worm infected approximately 300,000 servers, yet a quick 'back of the envelope' calculation (note 5) indicates that 1000 sites connected to T1 lines and dedicating only 5% of their total bandwidth to LaBrea's "-p" option would have been able to capture and hold all Code Red scanning threads at once.
>
> And by capturing these scanning threads, LaBrea makes it possible to contact compromised system owners while keeping their systems from compromising other systems.

What does the future hold in store for LaBrea?
According to Liston, here are some possible expansions on LaBrea that may come in the future:

- A "live" PC version. A similar "tarpitting" capability might exist for unused ports on existing machines. This becomes very handy for IP addresses that are in use – current LaBrea 2.0 code can only assume the identity of *unused* addresses. If there is an agent running on live computers, they can "tarpit" any connection attempt for ports that are not in use on that computer. This would be customizable – the user determines how much bandwidth to allocate to LaBrea, and then the program would adjust automatically.
- Fake "Profiles" or "machine types" for the LaBrea server. This would allow the software to answer more selectively, thereby tricking the port scanner into thinking that the unused IP address is a specific type of server. For instance, LaBrea could answer as a Windows NT server, or a web server, for any given IP address. This creates confusion for the port scanner, which will not be able to tell if the server is real or not. This feature will be important in the future. If the security community begins widespread use of LaBrea, it is inevitable that "LaBrea detectors" will appear, which will be able to use stack fingerprinting techniques to determine if an address is a real address or a "LaBrea" address.

Conclusion

LaBrea is one of the best ideas in security retaliation. By only replying to unsolicited communication, the product avoids some of the ethical issues of other forms of retaliation. With its techniques in TCP windowing, the product is effective at stopping quick-spreading worms and unintelligent port scans. As the product evolves, it will hopefully provide more confusion and headaches for the attacker community. If this product can gain wide acceptance, evolve to keep up with attack techniques, and centralize logging, it can be a powerful weapon the information security defense. Visit www.hackbusters.net now to learn more!

[1] Perez, Juan Carlos. "Most IT Security Problems are self-inflicted." NetworkWorldFusion News. IDG News service. October 9th, 2001. URL: http://www.nwfusion.com/news/2001/1009itsec.html

[2] Liston, Tom. "Welcome to My Tarpit. The Tactical and Strategic Use of Labrea".
URL: http://www.hackbusters.net/LaBrea/LaBrea.txt

List of References

Scambray, Joel;McClure, Stuart;Kurtz, George. Hacking Exposed, 2nd Edition. Berkeley, CA. Osborn/McGraw-Hill, 2001. 34-69.

Fyodor@insecure.org "Nmap Stealth Port Scanner Intro." Last modified June 21st, 2001.
URL: http://www.insecure.org/nmap/index.html#intro

Fyodor@insecure.org. "Remote OS Detection via TCP/IP Stack Fingerprinting." Last Modified April 10th, 1999.
URL: http://www.insecure.org/nmap/nmap-fingerprinting-article.html

Northcutt, Stephen. "Intrusion Detection Overview and Trends in Internet Attacks." SANS Internet Threat Brief. Last re-write April 22nd, 2000. 20-28.

CERT.org. "CERT® Advisory CA-2001-19 'Code Red' Worm Exploiting Buffer Overflow In IIS Indexing Service DLL." Last Revised August 23rd, 2001.
URL: http://www.cert.org/advisories/CA-2001-19.html

Permeh, Ryan; Maiffret, Mark. ".ida 'Code Red' Worm." July 17th, 2001.
URL: http://www.eeye.com/html/Research/Advisories/AL20010717.html

Microsoft. "Unchecked Buffer in Index Server ISAPI Extension Could Enable Web Server Compromise." Last Revised August 21st, 2001.
URL:
http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/bulletin/MS01-033.asp

CERT.org "CERT® Advisory CA-2001-13 Buffer Overflow In IIS Indexing Service DLL." Last revised August 16th, 2001.
URL: http://www.cert.org/advisories/CA-2001-13.html

Perez, Juan Carlos. "Most IT Security Problems are self-inflicted." NetworkWorldFusion News. IDG News service. October 9th, 2001.
URL: http://www.nwfusion.com/news/2001/1009itsec.html

Stein, Lincoln; Stewart, John. "The World Wide Web Security FAQ." Question #10. World-Wide Web Consortium. Last revised July 28th, 2001.
URL: http://www.w3.org/Security/Faq/wwwsf6.html#DOS-Q10

Honeynet.org. "Frequently Asked Questions."

URL: http://project.honeynet.org/faq.html#faq1

Liston, Tom. "Labrea."
URL: http://www.hackbusters.net/LaBrea/

Liston, Tom. "Welcome to My Tarpit. The Tactical and Strategic Use of Labrea".
URL: http://www.hackbusters.net/LaBrea/LaBrea.txt

Pine Mountain Group. "The Internet Protocol (IP)" Course workbook. Last revised 2000.

Northcutt, Stephen. Network Intrusion Detection. An Analyst's Handbook.
Indianapolis, IN. New Riders, 1999. 9-11.