Global Information Assurance Certification Paper

SANS GIAC

Practical Assignment for SANS Security DC 2000

By Jason Baeder

15 August 2000

Detect #1

```
Jun 30 13:16:55 perm-rtr1 %SEC-6-IPACCESSLOGP: list 101 denied
tcp 202.0.178.98(53) -> a.b.c.1(53), 1 packet

Jun 30 13:16:55 perm-rtr1 %SEC-6-IPACCESSLOGP: list 101 denied
tcp 202.0.178.98(53) -> a.b.c.2(53), 1 packet

Jun 30 13:16:55 perm-rtr1 %SEC-6-IPACCESSLOGP: list 101 denied
tcp 202.0.178.98(53) -> a.b.c.3(53), 1 packet

Jun 30 13:16:55 perm-rtr1 %SEC-6-IPACCESSLOGP: list 101 denied
tcp 202.0.178.98(53) -> a.b.c.4(53), 1 packet

Jun 30 13:16:55 perm-rtr1 %SEC-6-IPACCESSLOGP: list 101 denied
tcp 202.0.178.98(53) -> a.b.c.5(53), 1 packet

....to a.b.c.255; then

Jun 30 13:25:42 perm-rtr2 %SEC-6-IPACCESSLOGP: list 191 denied
tcp 202.0.178.98(53) -> a.b.d.64(53), 1 packet

Jun 30 13:25:42 perm-rtr2 %SEC-6-IPACCESSLOGP: list 191 denied
tcp 202.0.178.98(53) -> a.b.d.65(53), 1 packet

Jun 30 13:25:42 perm-rtr2 %SEC-6-IPACCESSLOGP: list 191 denied
tcp 202.0.178.98(53) -> a.b.d.66(53), 1 packet

Jun 30 13:25:42 perm-rtr2 %SEC-6-IPACCESSLOGP: list 191 denied
tcp 202.0.178.98(53) -> a.b.d.67(53), 1 packet
```

```
Jun 30 13:25:42 perm-rtr2 %SEC-6-IPACCESSLOGP: list 191 denied
tcp 202.0.178.98(53) -> a.b.d.68(53), 1 packet

Jun 30 13:25:42 perm-rtr2 %SEC-6-IPACCESSLOGP: list 191 denied
tcp 202.0.178.98(53) -> a.b.d.69(53), 1 packet

.....to a.b.d.255
```

1. Source of trace
   a.b.c.0 and a.b.d.64 are publicly visible class C networks that we own.

2. Detect was generated by
   Cisco routers that provide Internet connectivity. Log entries have been sanitized and abbreviated. Fields, from left to right, are: Date, time, router name, Cisco router log indicator, ACL that created this log entry, ACL action, protocol, source address and port, destination address and port, number of packets associated with this event.

3. Probability the source address was spoofed:
   Low probability; a spoofed source address would not return information to the attacker. Whois investigation reveals the source address belongs to block of IP's owned by a Hong Kong ISP. Traceroute shows a live host.

4. Description of attack:
   This is reconnaissance in preparation for an attack to exploit weaknesses in BIND.

5. Attack mechanism:
   This is a host-by-host scan of two class C network in search of DNS servers. Due to the limited information presented by the log entries, there is not much data beyond the obvious scanning that can be gleaned from them - except for one very interesting item. The source and destination ports of the scan are TCP/53. The attacker could gain three bits of

information by using TCP/53, as opposed to UDP/53. One, if he receives a SYN/ACK, he can assume DNS service is up on the host that returned the response. Two, because TCP/53 is used for DNS zone transfer, he could assume that zone transfers are probably allowed to/from that host because the TCP/53 was allowed through filtering routers in front of the host. By initiating a a zone transfer, the attacker would hope to gain more detailed knowledge of our network. Three, because TCP/53 as both source and destination port was used in older versions of BIND for DNS zone transfer, he can assume he has found older, more vulnerable versions of BIND that are ripe for exploits.

6. Correlation:
   See TCP/IP for Intrusion Detection and Perimeter Defense, pg. 5-20, for chart of DNS ports.
   See Cert Advisory CA-98.05 at http://www.cert.org/advisories/CA-98.05.bind_problems.html for BIND vulnerabilities.
   See CVE-1999-009, CVE-1999-0011, CVE-1999-0835, CVE-1999-0837, CVE-1999-0848, CVE-1999-0849, CVE-1999-0851 at http://cve.mitre.org
   A brief survey of network captures at http://www.sans.org/giac.htm and http://www.sans.org/y2k/analysts.htm shows similar scans of DNS servers with TCP/53 as the destination port, but not TCP/53 as the source port.

7. Evidence of active targeting:
   Since this attack was for reconnaissance, we can say the attack was targeted only insofar as the attacker chose these two publicly visible network segments (out of a group of several) and strictly defined the type of service he was looking for.

8. Severity:
   Criticality=5
   + Lethality=4

   System Countermeasures=5
   + Network Countermeasures=5
   ===============
   Severity=-1

   Since the perimeter routers stopped this traffic, and existing servers in those segments are patched to current vendor recommended levels, both system and network countermeasures were rated high.

9. Defensive recommendation:
   No additional action; the perimeter routers blocked the traffic.

10. Test Question:

    Jun 30 13:16:55 perm-rtr1 %SEC-6-IPACCESSLOGP: list 101 denied tcp 202.0.178.98(53) -> a.b.c.1(53), 1 packet
    Jun 30 13:16:55 perm-rtr1 %SEC-6-IPACCESSLOGP: list 101 denied tcp 202.0.178.98(53) -> a.b.c.2(53), 1 packet
    Jun 30 13:16:55 perm-rtr1 %SEC-6-IPACCESSLOGP: list 101 denied tcp 202.0.178.98(53) -> a.b.c.3(53), 1 packet

    a) Normal DNS zone transfer
    b) Scan for DNS servers capable of DNS zone transfer <-- answer
    c) Scan for printers
    d) Normal DNS query

Detect #2

Jul 27 19:19:04 perm-rtr1 %SEC-6-IPACCESSLOGP: list 191 denied udp x.x.x.x(137) -> e.f.g.109(137), 1 packet
Jul 27 19:24:21 perm-rtr1 %SEC-6-IPACCESSLOGP: list 191 denied udp x.x.x.x(137) -> e.f.g.109(137), 2 packets

1. Source of trace:
   e.f.g.0 is a publicly visible class C network that we own.

2. Detect was generated by:
   Cisco routers that provide internet connectivity. Log entries have been sanitized and abbreviated. Fields, from left to right, are: Date, time, router name, Cisco router log indicator, ACL that created this log entry, ACL action, protocol, source address and port, destination address and port, number of packets associated with this event.

3. Probability the source address was spoofed:
   Very likely. Source address belongs to one our internal networks (not a private address; as such, has been sanitized); it should not have been coming from the

Internet.

4. Description of attack:
   A reconnaissance pass on a server looking for Windows NETBIOS information.

5. Attack mechanism:
   Most likely a NBTSTAT command directed at the server in an attempt to solicit the server's Windows naming information, such as server NetBIOS name, domain name, master browser status. A positive response to this query would confirm for the attacker that the targeted server is likely a Windows NT server (though UNIX with Samba will answer this query as well). The information returned would give the attacker enough information to begin launching various NetBIOS-based exploits, such as file shares and null-user sessions, against this server and any other servers that can be found in the now known NT domain.

6. Correlation:
   See page Network-Based Intrusion Detection Analysis and Workshop, pg. 287-289.
   See http://www.sans.org/topten.htm

7. Evidence of active targeting:
   Absolutely. The attack is targeted at a specific, publicly visible server, with a specific protocol.

8. Severity:
   Criticality=4
   + Lethality=3

   System Countermeasures=5
   + Network Countermeasures=5
   ===============
   Severity=-3

   The server that was targeted is a web server. Not critical,

not lethal. Since it is a UNIX system (without Samba), and since the router stopped this traffic, both system and network countermeasures are rated high.

9. Defensive recommendations:

   The perimeter routers blocked the traffic; no further actions is needed.

10. Test question:

    Jul 27 19:19:04 perm-rtr1 %SEC-6-IPACCESSLOGP: list 191 denied udp x.x.x.x(137) -> e.f.g.109(137), 1 packet

    A) WINS name registation
    B) Windows NT directory replication
    C) Windows file share
    D) NBTSTAT query <---answer

Detect #3

Jun 30 21:48:09 perm-rtr1 %SEC-6-IPACCESSLOGP: list 101 denied tcp 206.105.232.14(1341) -> a.b.c.2(1243), 1 packet

Jun 30 21:48:09 perm-rtr1 %SEC-6-IPACCESSLOGP: list 101 denied tcp 206.105.232.14(1344) -> a.b.c.5(1243), 1 packet

Jun 30 21:48:09 perm-rtr1 %SEC-6-IPACCESSLOGP: list 101 denied tcp 206.105.232.14(1343) -> a.b.c.4(1243), 1 packet

Jun 30 21:48:09 perm-rtr1 %SEC-6-IPACCESSLOGP: list 101 denied tcp 206.105.232.14(1340) -> a.b.c.1(1243), 1 packet

Jun 30 21:48:09 perm-rtr1 %SEC-6-IPACCESSLOGP: list 101 denied tcp 206.105.232.14(1342) -> a.b.c.3(1243), 1 packet

....to a.b.c.255; then

Jun 30 22:22:45 perm-rtr2 %SEC-6-IPACCESSLOGP: list 191 denied

```
tcp 206.105.224.82(3661) -> a.b.d.64(1243), 1 packet

Jun 30 22:22:45 perm-rtr2 %SEC-6-IPACCESSLOGP: list 191 denied
tcp 206.105.224.82(3662) -> a.b.d.65(1243), 1 packet

Jun 30 22:22:45 perm-rtr2 %SEC-6-IPACCESSLOGP: list 191 denied
tcp 206.105.224.82(3663) -> a.b.d.66(1243), 1 packet

Jun 30 22:22:45 perm-rtr2 %SEC-6-IPACCESSLOGP: list 191 denied
tcp 206.105.224.82(3664) -> a.b.d.67(1243), 1 packet

Jun 30 22:22:45 perm-rtr2 %SEC-6-IPACCESSLOGP: list 191 denied
tcp 206.105.224.82(3665) -> a.b.d.68(1243), 1 packet

...to a.b.d.254
```

1. Source of trace:
   a.b.c.0 and a.b.d.64 are publicly visible class C networks that we own.

2. Detect was generated by:
   Cisco routers that provide internet connectivity. Log entries have been sanitized and abbreviated. Fields, from left to right, are: Date, time, router name, Cisco router log indicator, ACL that created this log entry, ACL action, protocol, source address and port, destination address and port, number of packets associated with this event.

3. Probability the source address was spoofed:
   Low probability; a spoofed source address would not return information to the attacker. Whois investigation reveals the source addresses belong to a major U.S.-based ISP; the addresses appear to be dynamic dial-up account assignments.

4. Description of attack:
   A search for resident SubSeven trojans.

5. Attack mechanism:
   This is a host-by-host scan. Note that during the scan of the first network it appears that target hosts are interleaved. This "pattern" appears in the remaining log entries for this portion of the attack. The second network segment is scanned less than 30 minutes after the previous scan ends. It is most likely the same scanner; perhaps he

hung up and dialed again.

6. Correlation
   http://www.cert.org/advisories/CA-99-02-Trojan-Horses.html
   http://www.sans.org/newlook/resources/IDFAQ/subseven.htm

7. Evidence of active targeting:
   Yes, insofar as the attacker was looking only for SubSeven trojans within specific network segments.

8. Severity:
   Criticality=1
   + Lethality=1

   System Countermeasures=5
   + Network Countermeasures=5
   ===============
   Severity=-8

   Out of context, criticality and lethality would be rated a 5, as evidence of communications between a Trojan client and server would indicate a compromised system on our network. However, in this case, there are no Windows-based hosts on the segments that were scanned (SubSeven is a Windows-based trojan). Because the perimeter routers stopped the traffic, both system and network countermeasures are rated high.

9. Defensive recommendation:
   While the perimeter routers blocked the incoming traffic, it would be prudent to 1) review the ACL's for outgoing traffic policy to be sure no Trojan traffic will escape our network, and 2) scan our internal networks for any evidence of Trojans.

10. Test Question:

    Jun 30 21:48:09 perm-rtr1 %SEC-6-IPACCESSLOGP: list 101 denied tcp 206.105.232.14(1341) ->a.b.c.2(1243), 1 packet
    Jun 30 21:48:09 perm-rtr1 %SEC-6-IPACCESSLOGP: list 101 denied tcp 206.105.232.14(1344) -> a.b.c.5(1243), 1 packet
    Jun 30 21:48:09 perm-rtr1 %SEC-6-IPACCESSLOGP: list 101 denied tcp 206.105.232.14(1343) -> a.b.c.4(1243), 1 packet

    A) normal IMAP traffic
    B) scan for trojans <----answer

C) scan for IMAP servers
D) abnormal IMAP traffic

Detect #4

```
Jul 14 07:21:57 perm-rtr1 %SEC-6-IPACCESSLOGP: list 101 denied
tcp 211.42.97.41(109) -> a.b.c.1(109), 1 packet

Jul 14 07:21:57 perm-rtr1 %SEC-6-IPACCESSLOGP: list 101 denied
tcp 211.42.97.41(109) -> a.b.c.2(109), 1 packet

Jul 14 07:21:57 perm-rtr1 %SEC-6-IPACCESSLOGP: list 101 denied
tcp 211.42.97.41(109) -> a.b.c.3(109), 1 packet

Jul 14 07:21:57 perm-rtr1 %SEC-6-IPACCESSLOGP: list 101 denied
tcp 211.42.97.41(109) -> a.b.c.4(109), 1 packet

Jul 14 07:21:57 perm-rtr1 %SEC-6-IPACCESSLOGP: list 101 denied
tcp 211.42.97.41(109) -> a.b.c.5(109), 1 packet

....to a.b.c.255; then

Jul 14 07:30:44 perm-rtr2 %SEC-6-IPACCESSLOGP: list 191 denied
tcp 211.42.97.41(109) -> a.b.d.64(109), 1 packet

Jul 14 07:30:44 perm-rtr2 %SEC-6-IPACCESSLOGP: list 191 denied
tcp 211.42.97.41(109) -> a.b.d.65(109), 1 packet

Jul 14 07:30:44 perm-rtr2 %SEC-6-IPACCESSLOGP: list 191 denied
tcp 211.42.97.41(109) -> a.b.d.66(109), 1 packet

Jul 14 07:30:44 perm-rtr2 %SEC-6-IPACCESSLOGP: list 191 denied
tcp 211.42.97.41(109) -> a.b.d.67(109), 1 packet

Jul 14 07:30:44 perm-rtr2 %SEC-6-IPACCESSLOGP: list 191 denied
tcp 211.42.97.41(109) -> a.b.d.68(109), 1 packet

Jul 14 07:30:44 perm-rtr2 %SEC-6-IPACCESSLOGP: list 191 denied
tcp 211.42.97.41(109) -> a.b.d.69(109), 1 packet

....to a.b.d.255
```

1. Source of trace:
   a.b.c.0 and a.b.d.64 are publicly visible class C networks that we own.

2. Detect was generated by:
   Cisco routers that provide internet connectivity. Log entries have been sanitized and abbreviated. Fields, from left to right, are: Date, time, router name, Cisco router log indicator, ACL that created this log entry, ACL action, protocol, source address and port, destination address and port, number of packets associated with this event.

3. Probability the source address was spoofed:
   Low probability; a spoofed source address would not return information to the attacker. Whois investigation reveals the source address belongs to block of IP's owned by a Korean ISP. Traceroute shows a live host.

4. Description of attack:
   This is reconnaissance in search of POP2 servers.

5. Attack mechanism:
   This is a host-by-host scan across two class C network subnets in search of hosts that will answer on TCP/109, POP2. As the source port is also TCP/109, it is likely these are crafted packets. This attack is unlikely to produce a positive result as POP2 is probably not used much these days (although there is a known exploit for it should it be found). On the other hand, this attack could be conceivably be used for network mapping, as those hosts that are alive and do not have an active service on TCP/109 would return a reset, while the router would return a "host not found" for addresses which are not populated with a host.

6. Correlation:
   See http://www.sans.org/y2k/062400.htm
   Also CVE-1999-0920 at http://cve.mitre.org

7. Evidence of active targeting:
   Yes, insofar as the attacker was looking only for a specific

service within specific network segments.

8. Severity

   Criticality=3
   + Lethality=2

   System Countermeasures=5
   + Network Countermeasures=5
   ===============
   Severity=-5

   The network segments that were scanned are populated with various UNIX hosts that could run a POP2 service (but they don't). The router stopped the traffic, the hosts are patched to current.

9. Defensive recommendations:
   The perimeter routers blocked the traffic; no further actions is needed.

10. Test question:

    Jul 14 07:30:44 perm-rtr2 %SEC-6-IPACCESSLOGP: list 191 denied tcp 211.42.97.41(109) -> a.b.d.66(109), 1 packet
    Jul 14 07:30:44 perm-rtr2 %SEC-6-IPACCESSLOGP: list 191 denied tcp 211.42.97.41(109) -> a.b.d.67(109), 1 packet
    Jul 14 07:30:44 perm-rtr2 %SEC-6-IPACCESSLOGP: list 191 denied tcp 211.42.97.41(109) -> a.b.d.68(109), 1 packet

    A) an aborted POP3 session
    B) a scan for POP3 servers
    C) Quick Mail transfer protocol
    D) attempted connections to POP2 servers <----answer

Detect #5

Jun 23 19:17:35 perm-rtr2 %SEC-6-IPACCESSLOGP: list 191 denied tcp 210.95.250.193(3946) -> a.b.e.3(98), 1 packet

```
Jun 23 19:17:35 perm-rtr2 %SEC-6-IPACCESSLOGP: list 191 denied
tcp 210.95.250.193(3955) -> a.b.e.12(98), 1 packet

Jun 23 19:17:35 perm-rtr2 %SEC-6-IPACCESSLOGP: list 191 denied
tcp 210.95.250.193(3950) -> a.b.e.7(98), 1 packet

Jun 23 19:17:35 perm-rtr2 %SEC-6-IPACCESSLOGP: list 191 denied
tcp 210.95.250.193(3959) -> a.b.e.16(98), 1 packet

Jun 23 19:17:35 perm-rtr2 %SEC-6-IPACCESSLOGP: list 191 denied
tcp 210.95.250.193(3963) -> a.b.e.20(98), 1 packet

Jun 23 19:17:35 perm-rtr2 %SEC-6-IPACCESSLOGP: list 191 denied
tcp 210.95.250.193(3968) -> a.b.e.25(98), 1 packet

Jun 23 19:17:35 perm-rtr2 %SEC-6-IPACCESSLOGP: list 191 denied
tcp 210.95.250.193(3972) -> a.b.e.29(98), 1 packet

Jun 23 19:17:36 perm-rtr2 %SEC-6-IPACCESSLOGP: list 191 denied
tcp 210.95.250.193(3976) -> a.b.e.33(98), 1 packet

Jun 23 19:17:36 perm-rtr2 %SEC-6-IPACCESSLOGP: list 191 denied
tcp 210.95.250.193(3990) -> a.b.e.47(98), 1 packet

Jun 23 19:17:36 perm-rtr2 %SEC-6-IPACCESSLOGP: list 191 denied
tcp 210.95.250.193(3994) -> a.b.e.51(98), 1 packet

Jun 23 19:17:36 perm-rtr2 %SEC-6-IPACCESSLOGP: list 191 denied
tcp 210.95.250.193(3999) -> a.b.e.56(98), 1 packet

Jun 23 19:17:36 perm-rtr2 %SEC-6-IPACCESSLOGP: list 191 denied
tcp 210.95.250.193(4003) -> a.b.e.60(98), 1 packet

Jun 23 19:17:36 perm-rtr2 %SEC-6-IPACCESSLOGP: list 191 denied
tcp 210.95.250.193(4400) -> a.b.e.70(98), 1 packet

Jun 23 19:17:36 perm-rtr2 %SEC-6-IPACCESSLOGP: list 191 denied
tcp 210.95.250.193(4401) -> a.b.e.71(98), 1 packet

Jun 23 19:17:36 perm-rtr2 %SEC-6-IPACCESSLOGP: list 191 denied
tcp 210.95.250.193(4402) -> a.b.e.72(98), 1 packet

Jun 23 19:17:36 perm-rtr2 %SEC-6-IPACCESSLOGP: list 191 denied
tcp 210.95.250.193(4403) -> a.b.e.73(98), 1 packet

Jun 23 19:17:36 perm-rtr2 %SEC-6-IPACCESSLOGP: list 191 denied
```

```
tcp 210.95.250.193(4404) -> a.b.e.74(98), 1 packet

Jun 23 19:17:36 perm-rtr2 %SEC-6-IPACCESSLOGP: list 191 denied
tcp 210.95.250.193(4405) -> a.b.e.75(98), 1 packet

Jun 23 19:17:36 perm-rtr2 %SEC-6-IPACCESSLOGP: list 191 denied
tcp 210.95.250.193(4409) -> a.b.e.79(98), 1 packet

Jun 23 19:17:36 perm-rtr2 %SEC-6-IPACCESSLOGP: list 191 denied
tcp 210.95.250.193(4413) -> a.b.e.83(98), 1 packet

Jun 23 19:17:36 perm-rtr2 %SEC-6-IPACCESSLOGP: list 191 denied
tcp 210.95.250.193(4417) -> a.b.e.87(98), 1 packet

Jun 23 19:17:36 perm-rtr2 %SEC-6-IPACCESSLOGP: list 191 denied
tcp 210.95.250.193(4421) -> a.b.e.91(98), 1 packet

Jun 23 19:17:36 perm-rtr2 %SEC-6-IPACCESSLOGP: list 191 denied
tcp 210.95.250.193(4425) -> a.b.e.95(98), 1 packet

Jun 23 19:17:36 perm-rtr2 %SEC-6-IPACCESSLOGP: list 191 denied
tcp 210.95.250.193(4430) -> a.b.e.100(98), 1 packet

Jun 23 19:17:36 perm-rtr2 %SEC-6-IPACCESSLOGP: list 191 denied
tcp 210.95.250.193(4434) -> a.b.e.104(98), 1 packet

Jun 23 19:17:36 perm-rtr2 %SEC-6-IPACCESSLOGP: list 191 denied
tcp 210.95.250.193(4437) -> a.b.e.107(98), 1 packet

Jun 23 19:17:36 perm-rtr2 %SEC-6-IPACCESSLOGP: list 191 denied
tcp 210.95.250.193(4442) -> a.b.e.112(98), 1 packet

Jun 23 19:17:36 perm-rtr2 %SEC-6-IPACCESSLOGP: list 191 denied
tcp 210.95.250.193(4446) -> a.b.e.116(98), 1 packet

Jun 23 19:17:36 perm-rtr2 %SEC-6-IPACCESSLOGP: list 191 denied
tcp 210.95.250.193(4450) -> a.b.e.120(98), 1 packet

Jun 23 19:17:36 perm-rtr2 %SEC-6-IPACCESSLOGP: list 191 denied
tcp 210.95.250.193(4455) -> a.b.e.125(98), 1 packet

Jun 23 19:17:36 perm-rtr2 %SEC-6-IPACCESSLOGP: list 191 denied
tcp 210.95.250.193(4475) -> a.b.e.145(98), 1 packet

Jun 23 19:17:36 perm-rtr2 %SEC-6-IPACCESSLOGP: list 191 denied
tcp 210.95.250.193(4496) -> a.b.e.166(98), 1 packet
```

```
Jun 23 19:17:36 perm-rtr2 %SEC-6-IPACCESSLOGP: list 191 denied
tcp 210.95.250.193(4513) -> a.b.e.183(98), 1 packet

Jun 23 19:17:36 perm-rtr2 %SEC-6-IPACCESSLOGP: list 191 denied
tcp 210.95.250.193(4525) -> a.b.e.195(98), 1 packet

Jun 23 19:17:36 perm-rtr2 %SEC-6-IPACCESSLOGP: list 191 denied
tcp 210.95.250.193(4537) -> a.b.e.207(98), 1 packet

Jun 23 19:17:37 perm-rtr2 %SEC-6-IPACCESSLOGP: list 191 denied
tcp 210.95.250.193(4563) -> a.b.e.233(98), 1 packet

Jun 23 19:17:38 perm-rtr2 %SEC-6-IPACCESSLOGP: list 191 denied
tcp 210.95.250.193(3946) -> a.b.e.3(98), 1 packet

Jun 23 19:17:38 perm-rtr2 %SEC-6-IPACCESSLOGP: list 191 denied
tcp 210.95.250.193(3950) -> a.b.e.7(98), 1 packet

Jun 23 19:17:38 perm-rtr2 %SEC-6-IPACCESSLOGP: list 191 denied
tcp 210.95.250.193(3954) -> a.b.e.11(98), 1 packet

Jun 23 19:17:38 perm-rtr2 %SEC-6-IPACCESSLOGP: list 191 denied
tcp 210.95.250.193(3958) -> a.b.e.15(98), 1 packet

Jun 23 19:17:38 perm-rtr2 %SEC-6-IPACCESSLOGP: list 191 denied
tcp 210.95.250.193(3970) -> a.b.e.27(98), 1 packet

Jun 23 19:17:38 perm-rtr2 %SEC-6-IPACCESSLOGP: list 191 denied
tcp 210.95.250.193(3983) -> a.b.e.40(98), 1 packet

Jun 23 19:17:38 perm-rtr2 %SEC-6-IPACCESSLOGP: list 191 denied
tcp 210.95.250.193(3987) -> a.b.e.44(98), 1 packet

Jun 23 19:17:39 perm-rtr2 %SEC-6-IPACCESSLOGP: list 191 denied
tcp 210.95.250.193(3991) -> a.b.e.48(98), 1 packet

Jun 23 19:17:39 perm-rtr2 %SEC-6-IPACCESSLOGP: list 191 denied
tcp 210.95.250.193(3994) -> a.b.e.51(98), 1 packet

Jun 23 19:17:39 perm-rtr2 %SEC-6-IPACCESSLOGP: list 191 denied
tcp 210.95.250.193(3998) -> a.b.e.55(98), 1 packet

Jun 23 19:17:39 perm-rtr2 %SEC-6-IPACCESSLOGP: list 191 denied
tcp 210.95.250.193(4005) -> a.b.e.62(98), 1 packet

Jun 23 19:17:39 perm-rtr2 %SEC-6-IPACCESSLOGP: list 191 denied
tcp 210.95.250.193(4401) -> a.b.e.71(98), 1 packet
```

```
Jun 23 19:17:39 perm-rtr2 %SEC-6-IPACCESSLOGP: list 191 denied
tcp 210.95.250.193(4402) -> a.b.e.72(98), 1 packet

Jun 23 19:17:39 perm-rtr2 %SEC-6-IPACCESSLOGP: list 191 denied
tcp 210.95.250.193(4403) -> a.b.e.73(98), 1 packet

Jun 23 19:17:39 perm-rtr2 %SEC-6-IPACCESSLOGP: list 191 denied
tcp 210.95.250.193(4404) -> a.b.e.74(98), 1 packet

Jun 23 19:17:39 perm-rtr2 %SEC-6-IPACCESSLOGP: list 191 denied
tcp 210.95.250.193(4405) -> a.b.e.75(98), 1 packet

Jun 23 19:17:39 perm-rtr2 %SEC-6-IPACCESSLOGP: list 191 denied
tcp 210.95.250.193(4406) -> a.b.e.76(98), 1 packet

Jun 23 19:17:39 perm-rtr2 %SEC-6-IPACCESSLOGP: list 191 denied
tcp 210.95.250.193(4407) -> a.b.e.77(98), 1 packet

Jun 23 19:17:39 perm-rtr2 %SEC-6-IPACCESSLOGP: list 191 denied
tcp 210.95.250.193(4408) -> a.b.e.78(98), 1 packet

Jun 23 19:17:39 perm-rtr2 %SEC-6-IPACCESSLOGP: list 191 denied
tcp 210.95.250.193(4409) -> a.b.e.79(98), 1 packet

Jun 23 19:17:39 perm-rtr2 %SEC-6-IPACCESSLOGP: list 191 denied
tcp 210.95.250.193(4410) -> a.b.e.80(98), 1 packet

Jun 23 19:17:39 perm-rtr2 %SEC-6-IPACCESSLOGP: list 191 denied
tcp 210.95.250.193(4411) -> a.b.e.81(98), 1 packet

Jun 23 19:17:39 perm-rtr2 %SEC-6-IPACCESSLOGP: list 191 denied
tcp 210.95.250.193(4412) -> a.b.e.82(98), 1 packet

Jun 23 19:17:39 perm-rtr2 %SEC-6-IPACCESSLOGP: list 191 denied
tcp 210.95.250.193(4413) -> a.b.e.83(98), 1 packet

Jun 23 19:17:39 perm-rtr2 %SEC-6-IPACCESSLOGP: list 191 denied
tcp 210.95.250.193(4414) -> a.b.e.84(98), 1 packet

Jun 23 19:17:39 perm-rtr2 %SEC-6-IPACCESSLOGP: list 191 denied
tcp 210.95.250.193(4415) -> a.b.e.85(98), 1 packet

Jun 23 19:17:39 perm-rtr2 %SEC-6-IPACCESSLOGP: list 191 denied
tcp 210.95.250.193(4416) -> a.b.e.86(98), 1 packet

Jun 23 19:17:39 perm-rtr2 %SEC-6-IPACCESSLOGP: list 191 denied
```

```
tcp 210.95.250.193(4417) -> a.b.e.87(98), 1 packet

Jun 23 19:17:39 perm-rtr2 %SEC-6-IPACCESSLOGP: list 191 denied
tcp 210.95.250.193(4418) -> a.b.e.88(98), 1 packet

Jun 23 19:17:39 perm-rtr2 %SEC-6-IPACCESSLOGP: list 191 denied
tcp 210.95.250.193(4419) -> a.b.e.89(98), 1 packet

Jun 23 19:17:39 perm-rtr2 %SEC-6-IPACCESSLOGP: list 191 denied
tcp 210.95.250.193(4420) -> a.b.e.90(98), 1 packet

Jun 23 19:17:39 perm-rtr2 %SEC-6-IPACCESSLOGP: list 191 denied
tcp 210.95.250.193(4421) -> a.b.e.91(98), 1 packet

Jun 23 19:17:39 perm-rtr2 %SEC-6-IPACCESSLOGP: list 191 denied
tcp 210.95.250.193(4422) -> a.b.e.92(98), 1 packet

Jun 23 19:17:39 perm-rtr2 %SEC-6-IPACCESSLOGP: list 191 denied
tcp 210.95.250.193(4423) -> a.b.e.93(98), 1 packet

Jun 23 19:17:39 perm-rtr2 %SEC-6-IPACCESSLOGP: list 191 denied
tcp 210.95.250.193(4424) -> a.b.e.94(98), 1 packet

Jun 23 19:17:39 perm-rtr2 %SEC-6-IPACCESSLOGP: list 191 denied
tcp 210.95.250.193(4425) -> a.b.e.95(98), 1 packet

Jun 23 19:17:39 perm-rtr2 %SEC-6-IPACCESSLOGP: list 191 denied
tcp 210.95.250.193(4426) -> a.b.e.96(98), 1 packet

Jun 23 19:17:39 perm-rtr2 %SEC-6-IPACCESSLOGP: list 191 denied
tcp 210.95.250.193(4427) -> a.b.e.97(98), 1 packet

Jun 23 19:17:39 perm-rtr2 %SEC-6-IPACCESSLOGP: list 191 denied
tcp 210.95.250.193(4430) -> a.b.e.100(98), 1 packet

Jun 23 19:17:39 perm-rtr2 %SEC-6-IPACCESSLOGP: list 191 denied
tcp 210.95.250.193(4435) -> a.b.e.105(98), 1 packet

Jun 23 19:17:39 perm-rtr2 %SEC-6-IPACCESSLOGP: list 191 denied
tcp 210.95.250.193(4439) -> a.b.e.109(98), 1 packet

Jun 23 19:17:39 perm-rtr2 %SEC-6-IPACCESSLOGP: list 191 denied
tcp 210.95.250.193(4442) -> a.b.e.112(98), 1 packet

Jun 23 19:17:39 perm-rtr2 %SEC-6-IPACCESSLOGP: list 191 denied
tcp 210.95.250.193(4447) -> a.b.e.117(98), 1 packet
```

```
Jun 23 19:17:39 perm-rtr2 %SEC-6-IPACCESSLOGP: list 191 denied
tcp 210.95.250.193(4451) -> a.b.e.121(98), 1 packet

Jun 23 19:17:39 perm-rtr2 %SEC-6-IPACCESSLOGP: list 191 denied
tcp 210.95.250.193(4458) -> a.b.e.128(98), 1 packet

Jun 23 19:17:39 perm-rtr2 %SEC-6-IPACCESSLOGP: list 191 denied
tcp 210.95.250.193(4462) -> a.b.e.132(98), 1 packet

Jun 23 19:17:39 perm-rtr2 %SEC-6-IPACCESSLOGP: list 191 denied
tcp 210.95.250.193(4466) -> a.b.e.136(98), 1 packet

Jun 23 19:17:39 perm-rtr2 %SEC-6-IPACCESSLOGP: list 191 denied
tcp 210.95.250.193(4471) -> a.b.e.141(98), 1 packet

Jun 23 19:17:39 perm-rtr2 %SEC-6-IPACCESSLOGP: list 191 denied
tcp 210.95.250.193(4474) -> a.b.e.144(98), 1 packet

Jun 23 19:17:39 perm-rtr2 %SEC-6-IPACCESSLOGP: list 191 denied
tcp 210.95.250.193(4478) -> a.b.e.148(98), 1 packet

Jun 23 19:17:39 perm-rtr2 %SEC-6-IPACCESSLOGP: list 191 denied
tcp 210.95.250.193(4483) -> a.b.e.153(98), 1 packet

Jun 23 19:17:39 perm-rtr2 %SEC-6-IPACCESSLOGP: list 191 denied
tcp 210.95.250.193(4487) -> a.b.e.157(98), 1 packet

Jun 23 19:17:39 perm-rtr2 %SEC-6-IPACCESSLOGP: list 191 denied
tcp 210.95.250.193(4491) -> a.b.e.161(98), 1 packet

Jun 23 19:17:39 perm-rtr2 %SEC-6-IPACCESSLOGP: list 191 denied
tcp 210.95.250.193(4495) -> a.b.e.165(98), 1 packet

Jun 23 19:17:39 perm-rtr2 %SEC-6-IPACCESSLOGP: list 191 denied
tcp 210.95.250.193(4499) -> a.b.e.169(98), 1 packet

Jun 23 19:17:39 perm-rtr2 %SEC-6-IPACCESSLOGP: list 191 denied
tcp 210.95.250.193(4503) -> a.b.e.173(98), 1 packet

Jun 23 19:17:40 perm-rtr2 %SEC-6-IPACCESSLOGP: list 191 denied
tcp 210.95.250.193(4507) -> a.b.e.177(98), 1 packet

Jun 23 19:17:40 perm-rtr2 %SEC-6-IPACCESSLOGP: list 191 denied
tcp 210.95.250.193(4511) -> a.b.e.181(98), 1 packet

Jun 23 19:17:40 perm-rtr2 %SEC-6-IPACCESSLOGP: list 191 denied
tcp 210.95.250.193(4522) -> a.b.e.192(98), 1 packet
```

```
Jun 23 19:17:40 perm-rtr2 %SEC-6-IPACCESSLOGP: list 191 denied
tcp 210.95.250.193(4535) -> a.b.e.205(98), 1 packet

Jun 23 19:17:40 perm-rtr2 %SEC-6-IPACCESSLOGP: list 191 denied
tcp 210.95.250.193(4548) -> a.b.e.218(98), 1 packet
```

1. Source of trace:
   a.b.e.0 is a class C network that we own.

2. Detect was generated by:
   Cisco routers that provide internet connectivity. Log entries have been sanitized and abbreviated. Fields, from left to right, are: Date, time, router name, Cisco router log indicator, ACL that created this log entry, ACL action, protocol, source address and port, destination address and port, number of packets associated with this event.

3. Probability the source address was spoofed:
   Low probability; a spoofed source address would not return information to the attacker. Whois investigation reveals the source address belongs to block of IP's owned by a Korean ISP. Traceroute shows a live host.

4. Description of attack:
   This is reconnaissance in search of hosts listening on TCP/98.

5. Attack mechanism:
   Recently TCP/98 has been become an increasingly popular port for attackers as it is now used for the GUI interface for linuxconf, a Linux system configuration utility that ships with many Linux distributions. By default, many of those systems turn this service on during installation. Misconfiguration by inexperienced administrators or users could leave a Linux system open to configuration, and thus compromise, by anyone. While no exploits for linuxconf have been published yet, such a host-by-host scan would, at the very least, reveal Linux systems (as those that respond to a connect on TCP/98).

One interesting characteristic of this attack is that the source ports from the attacking host are not in the sequence one would expect. Normally one would expect to see the source ports increment sequentially. Because of this, at first glance the source ports appear jumbled as the hosts to be scanned are interleaved. But if you look closely, it becomes apparent that each host to be scanned is associated with a source port on the attacker. Note the two lines in red: as the attacker rolls back to the beginning of the network, the same source port is used to scan host a.b.e.3 again.

Definitely a crafted packets, and definitely a scanner with a purpose.

6. Correlation:
   http://www.sans.org/y2k/practical/John_Springer.doc, detect #8
   http://www.sans.org/y2k/practical/Diane_Wood.doc, detect #8
   http://www.cert.org/current/current_activity.html#scans

7. Evidence of active targeting:
   Yes, insofar as the attacker was looking only for a specific service within specific network segments. But the interesting thing here is that the network in question, a.b.e.0. While it is public knowledge that we own this network segment - and thus the attacker knew about it - it is not currently publicly accessible. I do not have direct knowledge that this network has ever been publicly used, but I suspect that is the case as it appears a route to this network is being advertised.

8. Severity Criticality=1
   + Lethality=1

   System Countermeasures=5
   + Network Countermeasures=5
   ===
   Severity=-8

   Since the network being attacked is not even publicly visible, and since the no Linux systems are installed on any of the publicly visible networks, and since the routers stopped the offending traffic, this is not severe.

9. Defensive recommendations:
   The perimeter routers blocked the traffic. However, if a route to this non-accessible network is being publicly advertised, that advertisement should be stopped.

10. Test Question:

    ```
    Jun 23 19:17:36 perm-rtr2 %SEC-6-IPACCESSLOGP: list 191
    denied tcp 210.95.250.193(3976) -> a.b.e.33(98), 1 packet
    Jun 23 19:17:36 perm-rtr2 %SEC-6-IPACCESSLOGP: list 191
    denied tcp 210.95.250.193(3990) -> a.b.e.47(98), 1 packet
    Jun 23 19:17:36 perm-rtr2 %SEC-6-IPACCESSLOGP: list 191
    denied tcp 210.95.250.193(3994) -> a.b.e.51(98), 1 packet
    ```

    A) normal tacnews traffic
    B) a scan for tacnews servers
    C) a broken tacnews server
    D) a scan for linuxconf <---98

Analysis of an Attack

1. Source of trace:
   A network in a lab.

2. Detect was generated by:
   Tcpdump, hex format dump to a file. Data was later processed with tcpdump and tcpshow.

3. Probability the source address was spoofed:
   None.

4. Description of attack:
   Port scan, then attempted exploit of statd and mountd in a Solaris system..

5. Attack mechanism:
   This attack was obtained from http://www.ducktank.net
   The victim host is sol7.victim.com, Solaris 7 system running on a SPARC box. The attacker is linux.hacker.com, a Redhat 6.2 system running a PC.

```
22:18:55.159279 linux.hacker.com.40269 > sol7.victim.com.80:
. ack 0 win 4096 (ttl 47, id 53411)
22:18:55.160736 sol7.victim.com.80 > linux.hacker.com.40269:
R 0:0(0) win 0 (DF) (ttl 47, id 50924)
22:18:55.248304 linux.hacker.com.1731 > sol7.victim.com.171:
S 2285475294:2285475294(0) win 32120 <mss
1460,sackOK,timestamp 9537421 0,nop,wscale 0> (DF) (ttl 64,
id 6313)
22:18:55.248675 sol7.victim.com.171 > linux.hacker.com.1731:
R 0:0(0) ack 2285475295 win 0 (DF) (ttl 64, id 50925)
22:18:55.248727 linux.hacker.com.1732 > sol7.victim.com.348:
S 2283976581:2283976581(0) win 32120 <mss
1460,sackOK,timestamp 9537421 0,nop,wscale 0> (DF) (ttl 64,
id 6314)
22:18:55.249047 sol7.victim.com.348 > linux.hacker.com.1732:
R 0:0(0) ack 2283976582 win 0 (DF) (ttl 64, id 50926)
22:18:55.249473 linux.hacker.com.1733 >
sol7.victim.com.1402: S 2290589852:2290589852(0) win 32120
<mss 1460,sackOK,timestamp 9537421 0,nop,wscale 0> (DF) (ttl
64, id 6315)
22:18:55.249812 sol7.victim.com.1402 >
linux.hacker.com.1733: R 0:0(0) ack 2290589853 win 0 (DF)
(ttl 64, id 50927)
22:18:55.249865 linux.hacker.com.1734 > sol7.victim.com.21:
S 2283102270:2283102270(0) win 32120 <mss
1460,sackOK,timestamp 9537421 0,nop,wscale 0> (DF) (ttl 64,
id 6316)
22:18:55.250330 linux.hacker.com.1735 >
sol7.victim.com.7005: S 2283332548:2283332548(0) win 32120
<mss 1460,sackOK,timestamp 9537421 0,nop,wscale 0> (DF) (ttl
64, id 6317)
22:18:55.250404 sol7.victim.com.21 > linux.hacker.com.1734:
S 2286734954:2286734954(0) ack 2283102271 win 10136
<nop,nop,timestamp 122052794 9537421,nop,wscale
0,nop,nop,sackOK,mss 1460> (DF) (ttl 255, id 50928)
22:18:55.250682 sol7.victim.com.7005 >
```

```
linux.hacker.com.1735: R 0:0(0) ack 2283332549 win 0 (DF)
(ttl 64, id 50929)
22:18:55.250732 linux.hacker.com.1734 > sol7.victim.com.21:
. ack 1 win 32120 <nop,nop,timestamp 9537421 122052794> (DF)
(ttl 64, id 6318)
22:18:55.251185 linux.hacker.com.1736 > sol7.victim.com.243:
S 2294729698:2294729698(0) win 32120 <mss
1460,sackOK,timestamp 9537421 0,nop,wscale 0> (DF) (ttl 64,
id 6319)
22:18:55.251549 linux.hacker.com.1737 > sol7.victim.com.751:
S 2283976506:2283976506(0) win 32120 <mss
1460,sackOK,timestamp 9537421 0,nop,wscale 0> (DF) (ttl 64,
id 6320)
22:18:55.251596 sol7.victim.com.243 > linux.hacker.com.1736:
R 0:0(0) ack 2294729699 win 0 (DF) (ttl 64, id 50930)
22:18:55.251890 sol7.victim.com.751 > linux.hacker.com.1737:
R 0:0(0) ack 2283976507 win 0 (DF) (ttl 64, id 50931)
0,nop,wscale 0> (DF) (ttl 64, id 6321)
22:18:55.252610 linux.hacker.com.1739 >
sol7.victim.com.1479: S 2284689956:2284689956(0) win 32120
<mss 1460,sackOK,timestamp 9537421 0,nop,wscale 0> (DF) (ttl
64, id 6322)
22:18:55.252656 sol7.victim.com.760 > linux.hacker.com.1738:
R 0:0(0) ack 2285472864 win 0 (DF) (ttl 64, id 50932)
22:18:55.252957 sol7.victim.com.1479 >
linux.hacker.com.1739: R 0:0(0) ack 2284689957 win 0 (DF)
(ttl 64, id 50933)
22:18:55.253319 linux.hacker.com.1740 > sol7.victim.com.576:
S 2286486953:2286486953(0) win 32120 <mss
1460,sackOK,timestamp 9537421 0,nop,wscale 0> (DF) (ttl 64,
id 6323)
22:18:55.253697 sol7.victim.com.576 > linux.hacker.com.1740:
R 0:0(0) ack 2286486954 win 0 (DF) (ttl 64, id 50934)
22:18:55.254302 linux.hacker.com.1734 > sol7.victim.com.21:
F 1:1(0) ack 1 win 32120 <nop,nop,timestamp 9537421
122052794> (DF) (ttl 64, id 6324)
22:18:55.254611 sol7.victim.com.21 > linux.hacker.com.1734:
. ack 2 win 10136 <nop,nop,timestamp 122052795 9537421> (DF)
(ttl 255, id 50935)
22:18:55.255278 linux.hacker.com.1741 >
sol7.victim.com.1405: S 2282312904:2282312904(0) win 32120
<mss 1460,sackOK,timestamp 9537421 0,nop,wscale 0> (DF) (ttl
64, id 6325)
```

```
22:18:55.255547 linux.hacker.com.1742 > sol7.victim.com.543:
S 2278815621:2278815621(0) win 32120 <mss
1460,sackOK,timestamp 9537421 0,nop,wscale 0> (DF) (ttl 64,
id 6326)
22:18:55.255869 sol7.victim.com.1405 >
linux.hacker.com.1741: R 0:0(0) ack 2282312905 win 0 (DF)
(ttl 64, id 50936)
22:18:55.255913 sol7.victim.com.543 > linux.hacker.com.1742:
R 0:0(0) ack 2278815622 win 0 (DF) (ttl 64, id 50937)
22:18:55.256330 linux.hacker.com.1743 > sol7.victim.com.132:
S 2288452983:2288452983(0) win 32120 <mss
1460,sackOK,timestamp 9537421 0,nop,wscale 0> (DF) (ttl 64,
id 6327)
22:18:55.256785 sol7.victim.com.132 > linux.hacker.com.1743:
R 0:0(0) ack 2288452984 win 0 (DF) (ttl 64, id 50938)
22:18:55.256839 linux.hacker.com.1744 >
sol7.victim.com.1381: S 2287459534:2287459534(0) win 32120
<mss 1460,sackOK,timestamp 9537421 0,nop,wscale 0> (DF) (ttl
64, id 6328)
22:18:55.257097 linux.hacker.com.1745 > sol7.victim.com.418:
S 2291707793:2291707793(0) win 32120 <mss
1460,sackOK,timestamp 9537421 0,nop,wscale 0> (DF) (ttl 64,
id 6329)
22:18:55.257141 sol7.victim.com.1381 >
linux.hacker.com.1744: R 0:0(0) ack 2287459535 win 0 (DF)
(ttl 64, id 50939)
22:18:55.257391 sol7.victim.com.418 > linux.hacker.com.1745:
R 0:0(0) ack 2291707794 win 0 (DF) (ttl 64, id 50940)
22:18:55.257946 linux.hacker.com.1746 > sol7.victim.com.549:
S 2281225563:2281225563(0) win 32120 <mss
1460,sackOK,timestamp 9537422 0,nop,wscale 0> (DF) (ttl 64,
id 6330)
22:18:55.258205 linux.hacker.com.1747 > sol7.victim.com.95:
S 2291240297:2291240297(0) win 32120 <mss
1460,sackOK,timestamp 9537422 0,nop,wscale 0> (DF) (ttl 64,
id 6331)
```

The segment above is the beginning of a full connect port scan against sol7.victim.com. You can see where linux.hacker.com begins sending SYN packets to random ports on sol7.victim.com *from sequential source ports* on it's own system. Ports that are closed on sol7.victim.com respond with a Reset. In the case of port 21, FTP command,

sol7.victim.com responds with a SYN/ACK. A three-way handshake insues, and linux.hacker.com even politely issues a FIN to close the connection. Connection attempts continue for a total of more than 900 ports, with completed three-way handshakes taking place for each open port. Then the scan enters a new phase: OS fingerprinting.

```
22:18:55.779243 linux.hacker.com.2715 >
sol7.victim.com.2049: F 1:1(0) ack 1 win 32120
<nop,nop,timestamp 9537474 122052846> (DF) (ttl 64, id 7377)
22:18:55.779568 sol7.victim.com.2049 >
linux.hacker.com.2715: . ack 2 win 10136 <nop,nop,timestamp
122052847 9537474> (DF) (ttl 255, id 51968)
22:18:55.780917 linux.hacker.com.2749 > sol7.victim.com.514:
F 1:1(0) ack 1 win 32120 <nop,nop,timestamp 9537474
122052847> (DF) (ttl 64, id 7378)
22:18:55.781248 sol7.victim.com.514 > linux.hacker.com.2749:
. ack 2 win 10136 <nop,nop,timestamp 122052848 9537474> (DF)
(ttl 255, id 51969)
22:18:55.782754 linux.hacker.com.40258 > sol7.victim.com.7:
SFP 802394890:802394890(0) win 4096 urg 0 <wscale 10,nop,mss
265,timestamp 1061109567 0,eol> (ttl 47, id 25382)
22:18:55.783623 linux.hacker.com.40260 >
sol7.victim.com.44330: S 802394890:802394890(0) win 4096
<wscale 10,nop,mss 265,timestamp 1061109567 0,eol> (ttl 47,
id 13890)
22:18:55.783824 linux.hacker.com.40261 >
sol7.victim.com.44330: . ack 0 win 4096 <wscale 10,nop,mss
265,timestamp 1061109567 0,eol> (ttl 47, id 43612)
22:18:55.783905 sol7.victim.com.44330 >
linux.hacker.com.40260: R 0:0(0) ack 802394891 win 0 (DF)
(ttl 47, id 51972)
22:18:55.784112 sol7.victim.com.44330 >
linux.hacker.com.40261: R 0:0(0) win 0 (DF) (ttl 47, id
51973)
22:18:55.784363 linux.hacker.com.40262 >
sol7.victim.com.44330: FP 802394890:802394890(0) win 4096
urg 0 <wscale 10,nop,mss 265,timestamp 1061109567 0,eol>
(ttl 47, id 24455)
22:18:55.784935 linux.hacker.com.40249 >
sol7.victim.com.44330: udp 300 (ttl 56, id 25257)
22:18:55.784980 sol7.victim.com.44330 >
```

```
linux.hacker.com.40262: R 0:0(0) ack 802394890 win 0 (DF)
(ttl 47, id 51974)
22:18:56.310912 linux.hacker.com.40257 > sol7.victim.com.7:
. win 4096 <wscale 10,nop,mss 265,timestamp 1061109567
0,eol> (ttl 47, id 53426)
22:18:56.311108 linux.hacker.com.40258 > sol7.victim.com.7:
SFP 802394890:802394890(0) win 4096 urg 0 <wscale 10,nop,mss
265,timestamp 1061109567 0,eol> (ttl 47, id 25293)
```

The pattern above is typical of "OS fingerprinting". linux.hacker.com is connecting to open and closed ports with a series of packets that contain "impossible flags". The response to those packets can be compared to known OS response; for instance, some OS's will not follow RFC 793 whe responding to a packet with just the FIN bit set. In other cases, "xmastree" packets are sent - with the SYN/FIN/PUSH and URG flags set - to gauge the OS's reaction. The end result is that the attacker determines open ports and his can guess at the identity of the victim OS. In thi case, the attacker sees TCP/111 open: SunRPC. Not surprising considering the attacker's scanning software, nmap, identified the victim OS as either Solaris 2.6 or 2.7. One packet is a sure-giveaway what nmap is at work here: a udp signature used in nmap's OS fingerprinting:

```
22:18:55.784935 192.168.1.199.40249 > 192.168.1.11.44330:
udp 300 (ttl 56, id 25257) 4500 0148 62a9 0000 3811 9ad9
c0a8 01c7
c0a8 010b 9d39 ad2a 0134 01d2 6868 6868
6868 6868 6868 6868 6868 6868 6868 6868
6868 6868 6868 6868 6868 6868 6868 6868
6868 6868 6868 6868 6868 6868 6868 6868
6868 6868 6868 6868 6868 6868 6868 6868
6868 6868 6868 6868 6868 6868 6868 6868
6868 6868 6868 6868 6868 6868 6868 6868
6868 6868 6868 6868 6868 6868 6868 6868
6868 6868 6868 6868 6868 6868 6868 6868
6868 6868 6868 6868 6868 6868 6868 6868
6868 6868 6868 6868 6868 6868 6868 6868
6868 6868 6868 6868 6868 6868 6868 6868
```

```
6868 6868 6868 6868 6868 6868 6868 6868
6868 6868 6868 6868 6868 6868 6868 6868
6868 6868 6868 6868 6868 6868 6868 6868
6868 6868 6868 6868 6868 6868 6868 6868
6868 6868 6868 6868 6868 6868 6868 6868
6868 6868 6868 6868 6868 6868 6868 6868
6868 6868 6868 6868
```

The attacker decides it might be worthwhile to pursue an RPC exploit, specifically one aimed at rpc.statd. He runs 'rpcinfo -p sol7.victim.com' to gather information about the rpc services that are running on the victim host.

```
22:19:18.862884 linux.hacker.com.895 > sol7.victim.com.111:
S 2312973759:2312973759(0) win 32120 <mss
1460,sackOK,timestamp 9539782 0,nop,wscale 0> (DF) (ttl 64,
id 7403)
22:19:18.863448 sol7.victim.com.111 > linux.hacker.com.895:
S 2291673541:2291673541(0) ack 2312973760 win 10136
<nop,nop,timestamp 122055156 9539782,nop,wscale
0,nop,nop,sackOK,mss 1460> (DF) (ttl 255, id 51999)
22:19:18.863704 linux.hacker.com.895 > sol7.victim.com.111:
. ack 1 win 32120 <nop,nop,timestamp 9539782 122055156> (DF)
(ttl 64, id 7404)
22:19:18.864545 linux.hacker.com.895 > sol7.victim.com.111:
P 1:45(44) ack 1 win 32120 <nop,nop,timestamp 9539782
122055156> (DF) (ttl 64, id 7405)
22:19:18.864841 sol7.victim.com.111 > linux.hacker.com.895:
. ack 45 win 10092 <nop,nop,timestamp 122055156 9539782>
(DF) (ttl 255, id 52000)
22:19:18.871024 sol7.victim.com.111 > linux.hacker.com.895:
P 1:1233(1232) ack 45 win 10136 <nop,nop,timestamp 122055156
9539782> (DF) (ttl 255, id 52001)
22:19:18.871280 linux.hacker.com.895 > sol7.victim.com.111:
. ack 1233 win 31856 <nop,nop,timestamp 9539783 122055156>
(DF) (ttl 64, id 7406)
22:19:18.926681 linux.hacker.com.895 > sol7.victim.com.111:
F 45:45(0) ack 1233 win 31856 <nop,nop,timestamp 9539788
122055156> (DF) (ttl 64, id 7407)
22:19:18.926949 sol7.victim.com.111 > linux.hacker.com.895:
. ack 46 win 10136 <nop,nop,timestamp 122055162 9539788>
```

```
(DF) (ttl 255, id 52002)
22:19:18.927588 sol7.victim.com.111 > linux.hacker.com.895:
F 1233:1233(0) ack 46 win 10136 <nop,nop,timestamp 122055162
9539788> (DF) (ttl 255, id 52003)
22:19:18.927811 linux.hacker.com.895 > sol7.victim.com.111:
. ack 1234 win 31856 <nop,nop,timestamp 9539789 122055162>
(DF) (ttl 64, id 7408)
```

The packets highlighted in red is where the command 'rpcinfo -p' is run bu the attacker and the victim returns a list of rpc services that are registered with portmapper. The attacker learns that both the rstatd and mountd are running....

```
100000 4 tcp 111 portmapper
100000 3 tcp 111 portmapper
100000 2 tcp 111 portmapper
.....[snip].....
100001 2 udp 32821 rstatd
100001 3 udp 32821 rstatd
100001 4 udp 32821 rstatd
...[snip]...
100005 1 udp 41700 mountd
100005 2 udp 41700 mountd
100005 3 udp 41700 mountd
100005 1 tcp 33092 mountd
100005 2 tcp 33092 mountd
100005 3 tcp 33092 mountd
```

so the exploit is launched...

```
22:19:44.704217 linux.hacker.com.896 > sol7.victim.com.111:
udp 56 (ttl 64, id 7410)
22:19:44.706296 sol7.victim.com.111 > linux.hacker.com.896:
udp 28 (DF) (ttl 255, id 52004)
22:19:44.707157 linux.hacker.com.897 >
sol7.victim.com.32772: udp 108 (ttl 64, id 7411)
22:19:44.720908 sol7.victim.com.32772 >
linux.hacker.com.897: udp 32 (DF) (ttl 255, id 52005)
22:19:44.722339 linux.hacker.com.897 >
sol7.victim.com.32772: udp 72 (ttl 64, id 7412)
```

```
22:19:44.728729 sol7.victim.com.32772 >
linux.hacker.com.897: udp 24 (DF) (ttl 255, id 52006)
```

What happened? The attacker attempted to use a statd exploit that is supposed to allow him to send arbitrary commands to the victim and have them run in the victim OS. The attacker is attempting to emulate the Mitnick attack: he attempts to create a rhost file in the root directory with a content of "+". But the attack fails: from the victim's log file:

```
Aug 9 22:19:25 SOL7 statd[130]: statd: attempt to create
"/var/statmon/sm/; echo "+" >> /.rhosts"
```

Nmap had identified the OS of the sol7.victim.com as either Solaris 2.6 or 2.7. In this case, it is 2.7; SUN had already fixed the statd vulnerability in 2.7. The attack fails. But our hacker need not give up. During the port scan he was able to pick up some valuable information about a service running on sol7.victim.com:

```
Packet 1927
Timestamp: 22:18:55.819036
Source Ethernet Address: 08:00:20:7C:30:39
Destination Ethernet Address: 00:80:C8:67:9E:7C
Encapsulated Protocol: IP
IP Header
Version: 4
Header Length: 20 bytes
Service Type: 0x00
Datagram Length: 145 bytes
Identification: 0xCB09
Flags: MF=off, DF=on
Fragment Offset: 0
TTL: 255
Encapsulated Protocol: TCP
Header Checksum: 0x2C3A
Source IP Address: 192.168.1.11
Destination IP Address: 192.168.1.199
TCP Header
Source Port: 25 (smtp)
Destination Port: 2563 (<unknown>)
```

Sequence Number: 2287599565
Acknowledgement Number: 2294715192
Header Length: 32 bytes (data=93)
Flags: URG=off, ACK=on, PSH=on
RST=off, SYN=off, FIN=off
Window Advertisement: 10136 bytes
Checksum: 0x2EE5
Urgent Pointer: 0
<Options not displayed>
TCP Data 220 sol7.victim.com ESMTP Sendmail 8.9.3+Sun/8.9.1; Wed, 9 Aug 2000 22:18:37 -0400 (EDT).

Perhaps an exploit for Sendmail could be used for the next round?.....

6. Correlation:
   http://www.sans.org/infosecFAQ/nfs.
   http://www.sans.org/newlook/resources/IDFAQ/blocking.htm
   http://www.sans.org/newlook/resources/IDFAQ/NMAP.htm
   http://www.sans.org/newlook/resources/IDFAQ/trouble_RPCs.htm
   http://www.cert.org/advisories/CA-94.15.NFS.Vulnerabilities.html

7. Evidence of active targeting:

   Yes; a portscan on a single host, and then an exploit on a service on that host.

8. Severity

   Criticality=4
   + Lethality=4

   System Countermeasures=5
   + Network Countermeasures=5
   ===============
   Serverity=-2

9. Defensive recommendations:

   All traffic to and from Ports 111, 2049, 32771 and 32772 should be blocked at the network perimeter. In addition, all Solaris systems should be patched to current levels (see http://www.sun.com) and

other measures outlined in the CERT advisory should be followed. http://www.cert.org/advisories/CA-94.15.NFS.Vulnerabilities.html

10. Test question:

    22:19:44.707157 linux.hacker.com.897 > sol7.victim.com.32772: udp 108 (ttl 64, id 7411)

    A) traceroute
    B) port scan
    C) trojan
    D) RPC attack <---answer

Consulting Scenario

1. Executive Summary
    1. MY.NET has been exposed to many reconnaissance scans from malevolent forces on the Internet during the 30 days it was monitored.
    2. Many systems are already compromised, including MY.NET.1.3, MY.NET.253.12, MY.NET.253.52 and possibly several systems on the MY.NET.97.0 subnet.
    3. The compromised systems must be sanitized immediately, and a comprehensive policy regarding network security must be implemented.
2. Incidents
    1. Scanning

       As MY.NET is entirely exposed to the Internet, it is not surprising to find daily instances of port scanning from all over the world. Some are general scans for open services, followed by an effort to "fingerprint" the operating system being scanned. This was likely

undertaken with a program called NMAP:

```
May 27 23:44:43 MY.NET.253.12:43746 -> MY.NET.14.1:251
SYN **S*****
May 27 23:44:43 MY.NET.253.12:43746 -> MY.NET.14.1:229
SYN **S*****
May 27 23:44:45 MY.NET.253.12:22108 -> MY.NET.14.1:13
SYN **S*****
May 27 23:44:45 MY.NET.253.12:45815 -> MY.NET.14.1:19
SYN **S*****
May 27 23:44:45 MY.NET.253.12:12158 -> MY.NET.14.1:23
SYN **S*****
May 27 23:44:45 MY.NET.253.12:12045 -> MY.NET.14.1:79
SYN **S*****
May 27 23:44:47 MY.NET.253.12:48810 -> MY.NET.14.1:2001
SYN **S*****
May 27 23:44:47 MY.NET.253.12:47851 -> MY.NET.14.1:6001
SYN **S*****
May 27 23:44:47 MY.NET.253.12:43753 -> MY.NET.14.1:7 SYN
2*S***** RESERVEDBITS
May 27 23:44:47 MY.NET.253.12:43754 -> MY.NET.14.1:7
NULL ********
May 27 23:44:47 MY.NET.253.12:43755 -> MY.NET.14.1:7
NMAPID **SF*P*U
May 27 23:44:47 MY.NET.253.12:43757 -> MY.NET.14.1:1 SYN
**S*****
May 27 23:44:47 MY.NET.253.12:43759 -> MY.NET.14.1:1
XMAS ***F*P*U
May 27 23:44:47 MY.NET.253.12:43746 -> MY.NET.14.1:1 UDP
May 27 23:44:47 MY.NET.253.12:43752 -> MY.NET.14.1:7 SYN
**S*****
```

Various scan methodologies such as tiny fragments, UDP, FTP-bounce and impossible packets were also observed.

Other scans were looking for specific services, such as DNS, FTP or Telnet:

```
Jun 5 01:37:24 208.220.120.13:53 -> MY.NET.1.16:53
SYNFIN **SF****
Jun 6 09:45:54 213.188.8.45:2508 -> MY.NET.205.94:21 SYN
**S*****
Jun 16 14:33:31 207.107.55.209:3075 -> MY.NET.60.20:23
```

SYN **S*****

Others were looking for resident Trojans, such as this:

```
Jun 6 19:06:41 194.154.153.201:3354 ->
MY.NET.97.148:1243 SYN **S***** <--SubSeven
Jun 6 19:06:41 194.154.153.201:3355 ->
MY.NET.97.148:21554 SYN **S***** <--Girlfriend
Jun 6 19:06:41 194.154.153.201:3356 ->
MY.NET.97.148:1080 SYN **S****** <--WinHole
Jun 6 19:06:41 194.154.153.201:3357 ->
MY.NET.97.148:20034 SYN **S***** <--NetBus 2 Pro
Jun 17 22:23:03 202.235.50.12:65535 -> MY.NET.1.12:8080
SYN **S***** <--RingZero
```

This type of activity is "normal" and to be expected in the Internet. This type of activity should not be reaching into your network, however. The actions required to minimize this type of activity within MY.NET are included in our recommendations later in this report.

2. Compromised Systems

Early on in the timeframe during which data was collected were able to discern evidence of systems internal to MY.NET that had been compromised and were now under hacker control. From these systems, they launched reconnaissance into MY.NET. (We did not undertake any network scans in MY.NET during the assessment period, and your staff was informed to avoid this type of activity. As such, we are left with the conclusion that these scans originated from users, either internal or external, and were undertaken with malicious intent. Based on the activity we noted coming from outside of MY.NET, it is most likely that the internally originated scans are being undertaken by persons situated outside of MY.NET.) They appear as follows:

```
05/24-17:24:45.410430 [**] spp_portscan: PORTSCAN
DETECTED from MY.NET.1.3 (THRESHOLD 7 connections in 2
```

seconds) [**]
05/24-17:24:46.772964 [**] spp_portscan: portscan status from MY.NET.1.3: 10 connections across 2 hosts: TCP(0), UDP(10) [**]

Scans launched from MY.NET.1.3 into the rest of MY.NET continued throughout the assessment period. After MY.NET.253.12 was captured, those in control also launched scans within MY.NET, searching for other Trojan systems and systems vulnerable to RPC exploits. Those in control of MY.NET.101.160 sought to gain Windows NT domain and workgroup information from one particular system, MY.NET.101.192. Perhaps this a PDC? This needs to be confirmed.

We saw the clearest of evidence of the presence of a Trojan server in MY.NET.253.52. It is evident that this system has been overtaken by the Trinoo trojan, and that various entities around the world have been in contact with this system. It appears that MY.NET.101.89 has likely been infected as well. Finally, we saw a lot of SNMP activity in the MY.NET.97.0 subnet. We believe that MY.NET.101.92, potentially an HP OpenView server, was under a denial-of-service attack from several compromised hosts on the MY.NET.97.0 subnet. It will take further investigation in concert with your network operations staff to confirm this.

3. Reconstruction and Remediation

    1. Compromised Systems

    Computer forensics must be performed on the compromised systems immediately. All evidence of intrusion should be copied to non-networked computers for further study and possible submission to law enforcement agencies. The systems should then be restored from backup to a state prior to their compromise. After the restore, the systems should be patched to the current vendor-recommended levels.

    2. Network audit

    An audit of all networked hosts will be conducted to

determine

1. if unnecessary, or necessary but exploitable, services are running

2. unnecessary services will be stopped; exploitable services will be patched

3. all operating systems will be patched to current levels

3. Network perimeter

The network perimeter will be secured in the following manner

1. A DMZ will be defined by choke routers and a firewall. The DMZ will be occupied by servers hosting services to be visible to the outside world: HTTP, FTP, DNS, SMTP.

2. Only necessary traffic will be allowed in and out of MY.NET and the DMZ: HTTP, HTTPS, DNS, SMTP, FTP. Limited ICMP traffic will be allowed ('ICMP echo reply' will be allowed in; 'ICMP recho request' will not be allowed in; all outgoing traceroutes will initiated from the DMZ only).

3. All hosts on MY.NET will be assigned "private" IP address (per RFC 1597). All connectivity between hosts on MY.NET's internal segments (segments behind the firewall) to the DMZ or the Internet will be via proxy and will be NAT'd to a virtual IP address. A simple schematic is included in Appendix II.

Appendix I: References

http://www.sans.org/newlook/resources/IDFAQ/port_137.htm
http://www.sans.org/newlook/resources/IDFAQ/IRC.htm
http://www.sans.org/newlook/resources/IDFAQ/ring_zero.htm
See http://www.sans.org/topten.htm
http://www.cert.org/advisories/CA-99-02-Trojan-Horses.html

Appendix II: Network Diagram