Global Information Assurance Certification Paper

GIAC Certified Intrusion Analyst
Practical Exam Version 2.8

Jeff Nieusma

New Orleans SANS Conference
January 2001

Table of Contents

Assignment 1: Network Detects (40 points)

*Submit **five** network detects, with analysis. Each of the detects must be different; do NOT submit two of the same attack. Please use the analysis format shown below so that we can grade your submission as fairly as possible.*

Detect 1: RPC Scan

record	router	date	time	action	proto	src_ip	src_p	dst_ip	dst_p
1	routerC	04/02/01	10:19:33	denied	tcp	212.103.165.11	4906	10.206.136.7	111
2	routerC	04/02/01	10:19:34	denied	tcp	212.103.165.11	4953	10.206.136.49	111
3	routerC	04/02/01	10:19:35	denied	tcp	212.103.165.11	1113	10.206.136.81	111
4	routerC	04/02/01	10:19:36	denied	tcp	212.103.165.11	1195	10.206.136.120	111
5	routerC	04/02/01	10:19:38	denied	tcp	212.103.165.11	1074	10.206.136.73	111
6	routerC	04/02/01	10:19:39	denied	tcp	212.103.165.11	1153	10.206.136.116	111
7	routerC	04/02/01	10:19:40	denied	tcp	212.103.165.11	1487	10.206.137.17	111
8	routerC	04/02/01	10:19:41	denied	tcp	212.103.165.11	1285	10.206.136.174	111
9	routerC	04/02/01	10:19:42	denied	tcp	212.103.165.11	1412	10.206.136.227	111
10	routerC	04/02/01	10:19:43	denied	tcp	212.103.165.11	1521	10.206.137.20	111
11	routerC	04/02/01	10:19:44	denied	tcp	212.103.165.11	1577	10.206.137.69	111
12	routerC	04/02/01	10:19:46	denied	tcp	212.103.165.11	1748	10.206.137.137	111
13	routerC	04/02/01	10:19:47	denied	tcp	212.103.165.11	2064	10.206.138.35	111
14	routerC	04/02/01	10:19:48	denied	tcp	212.103.165.11	1858	10.206.137.188	111
15	routerC	04/02/01	10:19:49	denied	tcp	212.103.165.11	1943	10.206.137.240	111
16	routerC	04/02/01	10:19:50	denied	tcp	212.103.165.11	2068	10.206.138.38	111
17	routerC	04/02/01	10:19:51	denied	tcp	212.103.165.11	2416	10.206.138.219	111
18	routerC	04/02/01	10:19:52	denied	tcp	212.103.165.11	2487	10.206.138.226	111
19	routerC	04/02/01	10:19:53	denied	tcp	212.103.165.11	2516	10.206.138.249	111
20	routerC	04/02/01	10:19:54	denied	tcp	212.103.165.11	2420	10.206.138.223	111
21	routerC	04/02/01	10:19:55	denied	tcp	212.103.165.11	2487	10.206.138.226	111
22	routerV	04/02/01	10:19:55	denied	tcp	212.103.165.11	2915	10.206.140.0	111
23	routerV	04/02/01	10:19:56	denied	tcp	212.103.165.11	3060	10.206.140.106	111
24	routerV	04/02/01	10:19:58	denied	tcp	212.103.165.11	3141	10.206.140.153	111
25	routerC	04/02/01	10:19:59	denied	tcp	212.103.165.11	3225	10.206.140.160	111
26	routerV	04/02/01	10:19:59	denied	tcp	212.103.165.11	2964	10.206.140.47	111
27	routerV	04/02/01	10:20:00	denied	tcp	212.103.165.11	3349	10.206.140.249	111
28	routerV	04/02/01	10:20:01	denied	tcp	212.103.165.11	3144	10.206.140.154	111
29	routerV	04/02/01	10:20:02	denied	tcp	212.103.165.11	3223	10.206.140.158	111
30	routerC	04/02/01	10:20:02	denied	tcp	212.103.165.11	3226	10.206.140.161	111
31	routerV	04/02/01	10:20:03	denied	tcp	212.103.165.11	3356	10.206.141.1	111
32	routerV	04/02/01	10:20:04	denied	tcp	212.103.165.11	3706	10.206.141.152	111
33	routerV	04/02/01	10:20:05	denied	tcp	212.103.165.11	3810	10.206.141.224	111
34	routerV	04/02/01	10:20:06	denied	tcp	212.103.165.11	3892	10.206.141.229	111
35	routerV	04/02/01	10:20:08	denied	tcp	212.103.165.11	3751	10.206.141.193	111
36	routerV	04/02/01	10:20:09	denied	tcp	212.103.165.11	3817	10.206.141.226	111
37	routerV	04/02/01	10:20:10	denied	tcp	212.103.165.11	3901	10.206.141.235	111
38	routerV	04/02/01	10:20:12	denied	tcp	212.103.165.11	4400	10.206.143.0	111
39	routerV	04/02/01	10:20:13	denied	tcp	212.103.165.11	4475	10.206.143.40	111
40	routerV	04/02/01	10:20:14	denied	tcp	212.103.165.11	4554	10.206.143.49	111
41	routerV	04/02/01	10:20:16	denied	tcp	212.103.165.11	4616	10.206.143.84	111
42	routerV	04/02/01	10:20:17	denied	tcp	212.103.165.11	4730	10.206.143.165	111
43	routerV	04/02/01	10:20:18	denied	tcp	212.103.165.11	4791	10.206.143.197	111
44	routerV	04/02/01	10:20:19	denied	tcp	212.103.165.11	4874	10.206.143.206	111
45	routerV	04/02/01	10:20:20	denied	tcp	212.103.165.11	4761	10.206.143.167	111
46	routerV	04/02/01	10:20:21	denied	tcp	212.103.165.11	4863	10.206.143.199	111
47	routerV	04/02/01	10:20:22	denied	tcp	212.103.165.11	4910	10.206.143.242	111

Source of trace

This log segment came from four days of ISP border router logs

Detect was generated by

Cisco access list logs slightly reformatted and put into MS Excel for easier sorting on multiple fields.

Probability the source address was spoofed

Low. There are no decoy addresses in the log file scanning the RPC port. All of the other 26 RPC port scans logged during these four days happen at independent times and do not overlap significantly.

Description of attack

This is a fast port scan looking for portmapper or rpcbind daemons. Possible explanations are to exploit CVE-1999-0168, in which the portmapper acts as a proxy to send commands to the local host, or to exploit CVE-2001-0236, a buffer overflow in the SNMP to DMI mapper, which registers with the portmapper. Since this log is from a border router, which doesn't permit the traffic, we can only speculate as to what the attacker would do after finding a listening portmapper.

Attack mechanism

Since it doesn't hit every IP address in the range, and since sessions are separated by a second or two, we assume that a ping sweep is happening and only systems that answer get probed on port 111. If the attacker finds a listening portmapper, it might just add the address to a list for future exploitation, or the next step might be to ask for all the registered services for immediate or future vulnerability probes. Since most RPC services run at high ports (usually > 32000) most networks that only use static packet filtering don't block access to these processes. As each new exploit is announced, the hacker community gets busy looking for vulnerabilities.

Correlation

The portmapper scan is one of the oldest and most common scans on the Internet today. In examining four days of log files from our border routers, we found 27 scans from separate attackers aimed at our network ranges. There are many vulnerabilities and exploits published for various RPC services.
CERT has issued Advisory CA-2001-05 about the exploitation of the snmpXdmid, and many others. See http://www.cert.org for details.

Evidence of active targeting

This is clearly a general network scan

Severity

(critical system + lethal attack) – (system + net countermeasures) = severity
(2 + 1) – (3 + 3) = -3

Defensive recommendation

This attack is already blocked for the border routers

Multiple choice test question

Since RPC services are generally shared among workgroup members and not usually secure enough to share with anonymous Internet users, what actions should be recommended to system administrators?

a. block inbound tcp port 111 at border routers
b. block outbound tcp port 111 at border routers

c. replace portmapper or rpcbind with a tcp_wrappers enabled version from http://ftp.porcupine.org/pub/security/index.html
d. all of the above

Answer: d

Detect 2: POP2 + RPC Scan

```
Apr 24 08:44:16 MDT: denied tcp 210.119.103.190(109) -> 10.37.128.1(109)
...                                                              .1(109)
Apr 24 08:44:26 MDT: denied tcp 210.119.103.190(109) -> 10.37.130.1(109)
Apr 24 08:44:36 MDT: denied tcp 210.119.103.190(111) -> 10.37.128.1(111)
...                                                          ....1
Apr 24 08:45:53 MDT: denied tcp 210.119.103.190(111) -> 10.37.143.1(111)
Apr 24 08:48:17 MDT: denied tcp 210.119.103.190(109) -> 10.7.175.1(109)
Apr 24 08:48:37 MDT: denied tcp 210.119.103.190(111) -> 10.7.175.1(111)
Apr 24 09:06:18 MDT: denied tcp 210.119.103.190(109) -> 10.37.128.2(109)
...                                                          ....2
Apr 24 09:07:25 MDT: denied tcp 210.119.103.190(111) -> 10.37.139.2(111)
Apr 24 09:10:09 MDT: denied tcp 210.119.103.190(109) -> 10.7.175.2(109)
Apr 24 09:10:29 MDT: denied tcp 210.119.103.190(111) -> 10.7.175.2(111)
Apr 24 09:28:18 MDT: denied tcp 210.119.103.190(111) -> 10.37.128.3(111)
...                                                          ....3
Apr 24 09:29:35 MDT: denied tcp 210.119.103.190(111) -> 10.37.143.3(111)
Apr 24 09:31:59 MDT: denied tcp 210.119.103.190(109) -> 10.7.175.3(109)
... .4 .5 .6 .7 .8 --- .86 .87 ...
Apr 25 16:13:20 MDT: denied tcp 210.119.103.190(109) -> 10.37.128.88(109)
...                                                          ....88
Apr 25 16:14:56 MDT: denied tcp 210.119.103.190(111) -> 10.37.143.88(111)
Apr 25 16:17:39 MDT: denied tcp 210.119.103.190(111) -> 10.7.175.88(111)
... he's not finished, but I didn't want to delay this any more. :-)
```

Source of trace

This log segment came from four days of ISP border router logs

Detect was generated by

Cisco access list logs slightly reformatted and sanitized

Probability the source address was spoofed

Low. There are no decoys and the attacker would need to receive the replies in order to be successful.

Description of attack

The attacker is looking for POP2 and portmapper daemons. This scan could be looking for CVE-1999-0920, CVE-1999-0168, CVE-1999-0190, any of the many RPC vulnerabilities, or, most likely, this scan is looking for honey pots. Since very few systems are still running POP2, it is a fairly safe bet that if it is running, it is meant to be broken; likewise with port 111. If a system has both ports open and available to the Internet, either the system administrator needs lots of education, or it is a trap.

Attack mechanism

The attacker is hitting all the addresses in a Class B range by changing the third octet in the inside loop, and the last octet in the outside loop. This is a brute force scan hitting all the addresses in the range sequentially. There doesn't seem to be any reconnaissance or intelligence in the attack. Since there is no NIDS on this

network, and (hopefully) no openings through the firewall for these queries, we can only speculate about what the attacker was hoping to gain from this attack.

Correlation

There have been a number of recent scans looking for ports 111 and 109. Since no one runs POP2 any more, the most likely reasons for port 109 to be open on any given machine is to allow hackers to get in. (I.e. a honeypot) The speculation is that the underground community is attempting to make a list of all the honeypots.

Evidence of active targeting

This is clearly a general network scan

Severity

(critical system + lethal attack) – (system + net countermeasures) = severity
(2 + 1) – (3 + 3) = -3

Defensive recommendation

These ports are blocked by the firewall already

Multiple choice test question

The above log segment is an example of
a. buffer overflow
b. targeted host port scan
c. network scan for specific service(s)
d. SYN attack

Answer: c

Detect 3: Looking for 1i0n

```
Apr 21 08:39:02 MDT denied tcp 209.112.47.7 (4879) -> 10.37.138.100 (1008)
Apr 21 08:54:23 MDT denied tcp 209.112.47.7 (2318) -> 10.37.139.33 (1008)
Apr 21 10:54:34 MDT denied tcp 209.112.47.7 (1593) -> 10.37.137.217 (1008)
Apr 21 11:58:57 MDT denied tcp 209.112.47.7 (4529) -> 10.37.138.217 (1008)
Apr 21 12:04:05 MDT denied tcp 209.112.47.7 (4529) -> 10.37.138.217 (1008)
Apr 21 16:45:57 MDT denied tcp 209.112.47.7 (1725) -> 10.37.135.203 (1008)
Apr 22 00:21:44 MDT denied tcp 209.112.47.7 (2784) -> 10.37.141.213 (1008)
Apr 22 00:49:22 MDT denied tcp 209.112.47.7 (4058) -> 10.168.141.168 (1008)
Apr 23 13:45:30 MDT denied tcp 209.112.47.7 (4034) -> 10.37.142.206 (1008)
Apr 23 21:31:27 MDT denied tcp 209.112.47.7 (3595) -> 10.7.175.157 (1008)
Apr 24 05:13:11 MDT denied tcp 209.112.47.7 (2524) -> 10.174.184.197 (1008)
Apr 24 11:41:16 MDT denied tcp 209.112.47.7 (1855) -> 10.37.143.77 (1008)
Apr 24 12:55:35 MDT denied tcp 209.112.47.7 (4297) -> 10.37.141.12 (1008)
Apr 24 12:59:52 MDT denied tcp 209.112.47.7 (4297) -> 10.37.141.12 (1008)
Apr 24 14:55:50 MDT denied tcp 209.112.47.7 (4083) -> 10.168.143.79 (1008)
Apr 24 15:15:53 MDT denied tcp 209.112.47.7 (4587) -> 10.168.141.221 (1008)
Apr 24 16:59:16 MDT denied tcp 209.112.47.7 (3311) -> 10.174.184.180 (1008)
Apr 24 20:57:56 MDT denied tcp 209.112.47.7 (2941) -> 10.174.134.222 (1008)
Apr 25 12:27:42 MDT denied tcp 209.112.47.7 (4934) -> 10.174.205.171 (1008)
Apr 25 14:56:32 MDT denied tcp 209.112.47.7 (3290) -> 10.174.204.43 (1008)
Apr 25 21:10:40 MDT denied tcp 209.112.47.7 (1940) -> 10.7.175.149 (1008)
Apr 26 03:32:34 MDT denied tcp 209.112.47.7 (2136) -> 10.168.140.188 (1008)
Apr 26 05:18:03 MDT denied tcp 209.112.47.7 (4252) -> 10.174.184.202 (1008)
```

Source of trace

This log segment came from ISP border router logs and represent addresses in four

different class A network ranges

Detect was generated by

Cisco access list logs slightly reformatted and sanitized

Probability the source address was spoofed

Not very likely. This attack would need to get responses in order to be successful.

Description of attack

This scan is looking for a root shell left behind by the 1i0n worm. The lion worm exploits the recently announced BIND vulnerability (CAN-2001-0010) and leaves a root shell running at port 1008 by adding a line to /etc/inetd.conf.

Attack mechanism

By looking at the times and source port information, this seems to be either a manual attack or it is coming from an extremely busy machine. Either the source ports are randomized, or there is a huge amount of traffic coming from this attacker. Unfortunately, we only have the packet filter logs and cannot look at the packet traces to get a better understanding of the methodology.

Correlation

Information about the lion worm can be found at http://www.whitehats.com/library/worms/lion/index.html

Evidence of active targeting

While it is readily apparent that the attacker is looking for root shells, the target addresses seem to be fairly random.

Severity

(critical system + lethal attack) – (system + net countermeasures) = severity
(3 + 5) – (4 + 4) = 0

Defensive recommendation

This port is already blocked by the packet filters

Multiple choice test question

An attacker attempting to connect to a specific TCP port on a number of machines is trying to:

a. break into a specific service daemon on the machine
b. find back doors on already compromised systems
c. map a network
d. both a and b

Answer: d

Detect 4: possible millennium worm

```
frame    Apr 21 19:17:52 MDT denied tcp 216.102.153.120 (16637) -> 10.174.203.2 (53)
frame    Apr 21 19:18:37 MDT denied tcp 216.102.153.120 (30770) -> 10.174.203.3 (53)
[snip]
frame    Apr 21 19:55:31 MDT denied tcp 216.102.153.120 (8759) -> 10.174.203.63 (53)
frame    Apr 21 20:01:24 MDT denied tcp 216.102.153.120 (10390) -> 10.174.204.72 (53)
frame    Apr 21 20:02:03 MDT denied tcp 216.102.153.120 (24176) -> 10.174.204.73 (53)
```

```
frame    Apr 21 20:04:41 MDT denied tcp 216.102.153.120 (18602) -> 10.174.204.77 (53)
frame    Apr 21 20:05:56 MDT denied tcp 216.102.153.120 (16840) -> 10.174.204.79 (53)

216.102.153.120 appeared 36 time(s) above and 1581 time(s) below:
routerV  Apr 21 19:16:47 MDT denied tcp 216.102.153.120 (25453) -> 10.174.134.1 (143)
routerV  Apr 21 19:16:55 MDT denied tcp 216.102.153.120 (27840) -> 10.174.183.1 (635)
routerV  Apr 21 19:16:58 MDT denied tcp 216.102.153.120 (28812) -> 10.174.183.1 (143)
routerV  Apr 21 19:17:05 MDT denied tcp 216.102.153.120 (29738) -> 10.174.204.1 (143)
routerV  Apr 21 19:17:08 MDT denied tcp 216.102.153.120 (23945) -> 10.174.134.1 (110)
routerV  Apr 21 19:17:32 MDT denied tcp 216.102.153.120 (25453) -> 10.174.134.1 (143)
routerV  Apr 21 19:17:35 MDT denied tcp 216.102.153.120 (27907) -> 10.174.183.1 (110)
routerV  Apr 21 19:17:43 MDT denied tcp 216.102.153.120 (27908) -> 10.174.184.1 (635)
routerV  Apr 21 19:17:46 MDT denied tcp 216.102.153.120 (28936) -> 10.174.204.1 (635)
routerV  Apr 21 19:17:49 MDT denied tcp 216.102.153.120 (29302) -> 10.174.183.1 (109)
frame    Apr 21 19:17:52 MDT denied tcp 216.102.153.120 (16637) -> 10.174.203.2 (53)
routerV  Apr 21 19:17:53 MDT denied tcp 216.102.153.120 (29738) -> 10.174.204.1 (143)
routerV  Apr 21 19:18:15 MDT denied tcp 216.102.153.120 (30987) -> 10.174.204.1 (109)
routerV  Apr 21 19:18:18 MDT denied tcp 216.102.153.120 (9301) -> 10.174.134.2 (110)
routerV  Apr 21 19:18:25 MDT denied tcp 216.102.153.120 (10334) -> 10.174.134.2 (143)
routerV  Apr 21 19:18:28 MDT denied tcp 216.102.153.120 (28479) -> 10.174.203.3 (635)
routerV  Apr 21 19:18:31 MDT denied tcp 216.102.153.120 (15224) -> 10.174.204.2 (635)
routerV  Apr 21 19:18:32 MDT denied tcp 216.102.153.120 (15911) -> 10.174.203.2 (143)
routerV  Apr 21 19:18:34 MDT denied tcp 216.102.153.120 (16820) -> 10.174.205.2 (109)
frame    Apr 21 19:18:37 MDT denied tcp 216.102.153.120 (30770) -> 10.174.203.3 (53)
routerV  Apr 21 19:19:00 MDT denied tcp 216.102.153.120 (25156) -> 10.174.134.3 (143)
[snip ... this scan hits every address in 5 class C's ]
routerV  Apr 21 21:51:31 MDT denied tcp 216.102.153.120 (10988) -> 10.174.134.254 (143)
routerV  Apr 21 21:51:39 MDT denied tcp 216.102.153.120 (14361) -> 10.174.203.254 (635)
routerV  Apr 21 21:52:03 MDT denied tcp 216.102.153.120 (22774) -> 10.174.134.255 (635)
routerV  Apr 21 21:52:06 MDT denied tcp 216.102.153.120 (9556) -> 10.174.134.254 (110)
routerV  Apr 21 21:52:10 MDT denied tcp 216.102.153.120 (10988) -> 10.174.134.254 (143)
routerV  Apr 21 21:52:13 MDT denied tcp 216.102.153.120 (12642) -> 10.174.184.254 (635)
routerV  Apr 21 21:52:17 MDT denied tcp 216.102.153.120 (14412) -> 10.174.184.254 (110)
routerV  Apr 21 21:52:20 MDT denied tcp 216.102.153.120 (15182) -> 10.174.204.254 (110)
routerV  Apr 21 21:52:21 MDT denied tcp 216.102.153.120 (15396) -> 10.174.184.254 (143)
routerV  Apr 21 21:53:06 MDT denied tcp 216.102.153.120 (16970) -> 10.174.204.254 (143)
routerV  Apr 21 21:55:59 MDT denied tcp 216.102.153.120 (22837) -> 10.174.134.255 (110)
routerV  Apr 21 21:57:20 MDT denied tcp 216.102.153.120 (25794) -> 10.174.204.255 (110)
```

Source of trace

This log segment came from ISP border router logs

Detect was generated by

Cisco access list logs slightly reformatted and sanitized

Probability the source address was spoofed

Low. There are no decoys and the attacker would need to receive the replies in order to be successful.

Description of attack

This could be the Millennium Worm attempting to spread itself, primarily to Linux machines, via the BIND IQUERY (CVE-1999-0009) vulnerability, the Qpopper Overflow (CVE-1999-0006) vulnerability, the imapd overflow (CVE-1999-0005) vulnerability, and the rpc.mountd (CVE-1999-0002) vulnerability. The anomaly here is the attempt to hit TCP port 109, which is not part of the “normal” signature.

Attack mechanism

The worm has already infected dsl-216-102-153-120.dsl.lsan03.pacbell.net. It is now trying to find more hosts to infect by scanning all addresses in class B address

ranges. This is obviously a brute force attack. Luckily it targets older Linux systems with holes that have been published for over a year. Hopefully, people have upgraded their systems already and this worm poses little risk.

Correlation

Much information about these Linux attacks came from http://www.robertgraham.com/pubs/firewall-seen.html and http://www.whitehats.com/library/worms/mworm/index.html

Evidence of active targeting

This worm is hitting sequential addresses looking for any host to attack. This is not aimed at any particular host.

Severity

(critical system + lethal attack) – (system + net countermeasures) = severity
(3 + 5) – (5 + 4) = -1

Defensive recommendation

Upgrade any existing older systems and block all external traffic to these ports. In this case the border router is already blocking this traffic.

Multiple choice test question

Any system administrator who runs Linux (or windows) without keeping current security patches installed needs:

a. a good firewall to block traffic to all but port 22
b. to get a different hobby
c. good backups for recovering after each break in
d. better education
e. all of the above except b

Answer: e

Detect 5: printer port scan

```
routerI Apr 21 07:02:01 MDT denied tcp 255.255.255.255 (31337) -> 10.7.175.133 (515)
routerI Apr 22 01:27:39 MDT denied tcp 255.255.255.255 (31337) -> 10.7.175.72 (515)
routerI Apr 22 02:23:06 MDT denied tcp 207.18.175.10 (31337) -> 10.37.133.91 (515)
routerV Apr 22 07:41:41 MDT denied tcp 207.18.175.10 (31337) -> 10.174.134.192 (515)
routerI Apr 22 13:11:02 MDT denied tcp 207.18.175.10 (31337) -> 10.37.131.183 (515)
routerI Apr 23 06:01:22 MDT denied tcp 207.18.175.10 (31337) -> 10.37.130.206 (515)
routerI Apr 23 11:15:01 MDT denied tcp 207.18.175.10 (31337) -> 10.37.135.53 (515)
routerI Apr 24 02:34:41 MDT denied tcp 255.255.255.255 (31337) -> 10.37.131.67 (515)
routerI Apr 24 02:44:10 MDT denied tcp 255.255.255.255 (31337) -> 10.37.134.71 (515)
routerI Apr 24 05:30:53 MDT denied tcp 207.18.175.10 (31337) -> 10.144.168.180 (515)
routerI Apr 24 06:35:02 MDT denied tcp 255.255.255.255 (31337) -> 10.37.131.12 (515)
routerV Apr 24 11:42:28 MDT denied tcp 255.255.255.255 (31337) -> 10.174.184.18 (515)
routerI Apr 24 12:56:58 MDT denied tcp 255.255.255.255 (31337) -> 10.144.168.145 (515)
routerI Apr 24 15:41:16 MDT denied tcp 255.255.255.255 (31337) -> 10.37.133.11 (515)
routerI Apr 24 15:48:17 MDT denied tcp 207.18.175.10 (31337) -> 10.168.141.190 (515)
routerI Apr 24 16:35:42 MDT denied tcp 255.255.255.255 (31337) -> 10.37.128.160 (515)
routerI Apr 24 20:43:16 MDT denied tcp 207.18.175.10 (31337) -> 10.144.168.206 (515)
routerV Apr 24 23:14:25 MDT denied tcp 207.18.175.10 (31337) -> 10.174.134.210 (515)
routerI Apr 24 23:44:29 MDT denied tcp 255.255.255.255 (31337) -> 10.37.136.121 (515)
routerI Apr 25 01:57:36 MDT denied tcp 255.255.255.255 (31337) -> 10.168.140.95 (515)
routerI Apr 25 04:25:41 MDT denied tcp 255.255.255.255 (31337) -> 10.144.168.189 (515)
```

```
routerI Apr 25 04:45:13 MDT denied tcp 255.255.255.255 (31337) -> 10.37.141.194 (515)
routerV Apr 25 06:29:35 MDT denied tcp 255.255.255.255 (31337) -> 10.174.203.175 (515)
routerV Apr 25 06:50:48 MDT denied tcp 255.255.255.255 (31337) -> 10.174.134.61 (515)
routerI Apr 25 08:17:24 MDT denied tcp 255.255.255.255 (31337) -> 10.168.143.108 (515)
routerI Apr 25 13:43:00 MDT denied tcp 255.255.255.255 (31337) -> 10.37.135.132 (515)
routerV Apr 25 16:58:10 MDT denied tcp 207.18.175.10 (31337) -> 10.174.134.152 (515)
routerV Apr 25 18:00:42 MDT denied tcp 207.18.175.10 (31337) -> 10.174.184.147 (515)
routerI Apr 25 21:26:36 MDT denied tcp 255.255.255.255 (31337) -> 10.168.140.69 (515)
routerI Apr 25 21:45:06 MDT denied tcp 255.255.255.255 (31337) -> 10.168.143.5 (515)
routerI Apr 25 21:59:30 MDT denied tcp 255.255.255.255 (31337) -> 10.37.128.163 (515)
routerI Apr 25 22:23:12 MDT denied tcp 255.255.255.255 (31337) -> 10.37.137.158 (515)
routerV Apr 26 00:27:35 MDT denied tcp 255.255.255.255 (31337) -> 10.174.183.173 (515)
routerI Apr 26 01:27:18 MDT denied tcp 207.18.175.10 (31337) -> 10.144.168.222 (515)
routerI Apr 26 07:04:40 MDT denied tcp 207.18.175.10 (31337) -> 10.37.134.115 (515)
routerI Apr 26 08:38:59 MDT denied tcp 255.255.255.255 (31337) -> 10.168.143.147 (515)
routerI Apr 26 10:07:24 MDT denied tcp 207.18.175.10 (31337) -> 10.168.141.99 (515)
routerI Apr 26 10:19:08 MDT denied tcp 255.255.255.255 (31337) -> 10.37.134.9 (515)
routerI Apr 26 11:02:28 MDT denied tcp 255.255.255.255 (31337) -> 10.37.138.33 (515)
routerI Apr 26 15:38:33 MDT denied tcp 207.18.175.10 (31337) -> 10.37.133.152 (515)
routerV Apr 26 20:53:49 MDT denied tcp 207.18.175.10 (31337) -> 10.174.134.3 (515)
routerI Apr 27 03:35:08 MDT denied tcp 255.255.255.255 (31337) -> 10.7.175.13 (515)
routerV Apr 27 10:50:57 MDT denied tcp 255.255.255.255 (31337) -> 10.174.134.75 (515)


--- Other traffic grouped by attackers ---

207.18.175.10 appeared 16 time(s) above and 16 time(s) below:
routerI Apr 23 06:01:22 MDT denied tcp 207.18.175.10 (31337) -> 10.37.130.206 (515)
routerI Apr 22 13:11:02 MDT denied tcp 207.18.175.10 (31337) -> 10.37.131.183 (515)
routerI Apr 26 15:38:33 MDT denied tcp 207.18.175.10 (31337) -> 10.37.133.152 (515)
routerI Apr 22 02:23:06 MDT denied tcp 207.18.175.10 (31337) -> 10.37.133.91 (515)
routerI Apr 26 07:04:40 MDT denied tcp 207.18.175.10 (31337) -> 10.37.134.115 (515)
routerI Apr 23 11:15:01 MDT denied tcp 207.18.175.10 (31337) -> 10.37.135.53 (515)
routerI Apr 24 05:30:53 MDT denied tcp 207.18.175.10 (31337) -> 10.144.168.180 (515)
routerI Apr 24 20:43:16 MDT denied tcp 207.18.175.10 (31337) -> 10.144.168.206 (515)
routerI Apr 26 01:27:18 MDT denied tcp 207.18.175.10 (31337) -> 10.144.168.222 (515)
routerI Apr 24 15:48:17 MDT denied tcp 207.18.175.10 (31337) -> 10.168.141.190 (515)
routerI Apr 26 10:07:24 MDT denied tcp 207.18.175.10 (31337) -> 10.168.141.99 (515)
routerV Apr 25 16:58:10 MDT denied tcp 207.18.175.10 (31337) -> 10.174.134.152 (515)
routerV Apr 22 07:41:41 MDT denied tcp 207.18.175.10 (31337) -> 10.174.134.192 (515)
routerV Apr 24 23:14:25 MDT denied tcp 207.18.175.10 (31337) -> 10.174.134.210 (515)
routerV Apr 26 20:53:49 MDT denied tcp 207.18.175.10 (31337) -> 10.174.134.3 (515)
routerV Apr 25 18:00:42 MDT denied tcp 207.18.175.10 (31337) -> 10.174.184.147 (515)

255.255.255.255 appeared 27 time(s) above and 1951 time(s) below:
routerI Apr 27 03:35:08 MDT denied tcp 255.255.255.255 (31337) -> 10.7.175.13 (515)
routerI Apr 21 07:02:01 MDT denied tcp 255.255.255.255 (31337) -> 10.7.175.133 (515)
routerI Apr 22 01:27:39 MDT denied tcp 255.255.255.255 (31337) -> 10.7.175.72 (515)
routerI Apr 24 16:35:42 MDT denied tcp 255.255.255.255 (31337) -> 10.37.128.160 (515)
routerI Apr 25 21:59:30 MDT denied tcp 255.255.255.255 (31337) -> 10.37.128.163 (515)
routerI Apr 24 06:35:02 MDT denied tcp 255.255.255.255 (31337) -> 10.37.131.12 (515)
routerI Apr 24 02:34:41 MDT denied tcp 255.255.255.255 (31337) -> 10.37.131.67 (515)
routerI Apr 24 15:41:16 MDT denied tcp 255.255.255.255 (31337) -> 10.37.133.11 (515)
routerI Apr 24 02:44:10 MDT denied tcp 255.255.255.255 (31337) -> 10.37.134.71 (515)
routerI Apr 26 10:19:08 MDT denied tcp 255.255.255.255 (31337) -> 10.37.134.9 (515)
routerI Apr 25 13:43:00 MDT denied tcp 255.255.255.255 (31337) -> 10.37.135.132 (515)
routerI Apr 24 23:44:29 MDT denied tcp 255.255.255.255 (31337) -> 10.37.136.121 (515)
routerI Apr 25 22:23:12 MDT denied tcp 255.255.255.255 (31337) -> 10.37.137.158 (515)
routerI Apr 26 11:02:28 MDT denied tcp 255.255.255.255 (31337) -> 10.37.138.33 (515)
routerI Apr 25 04:45:13 MDT denied tcp 255.255.255.255 (31337) -> 10.37.141.194 (515)
routerI Apr 24 12:56:58 MDT denied tcp 255.255.255.255 (31337) -> 10.144.168.145 (515)
routerI Apr 25 04:25:41 MDT denied tcp 255.255.255.255 (31337) -> 10.144.168.189 (515)
routerI Apr 25 21:26:36 MDT denied tcp 255.255.255.255 (31337) -> 10.168.140.69 (515)
routerI Apr 25 01:57:36 MDT denied tcp 255.255.255.255 (31337) -> 10.168.140.95 (515)
```

```
routerI Apr 25 08:17:24 MDT denied tcp 255.255.255.255 (31337) -> 10.168.143.108 (515)
routerI Apr 26 08:38:59 MDT denied tcp 255.255.255.255 (31337) -> 10.168.143.147 (515)
routerI Apr 25 21:45:06 MDT denied tcp 255.255.255.255 (31337) -> 10.168.143.5 (515)
routerV Apr 25 06:50:48 MDT denied tcp 255.255.255.255 (31337) -> 10.174.134.61 (515)
routerV Apr 27 10:50:57 MDT denied tcp 255.255.255.255 (31337) -> 10.174.134.75 (515)
routerV Apr 26 00:27:35 MDT denied tcp 255.255.255.255 (31337) -> 10.174.183.173 (515)
routerV Apr 24 11:42:28 MDT denied tcp 255.255.255.255 (31337) -> 10.174.184.18 (515)
routerV Apr 25 06:29:35 MDT denied tcp 255.255.255.255 (31337) -> 10.174.203.175 (515)
```

Source of trace

This log segment came from ISP border router logs

Detect was generated by

Cisco access list logs slightly reformatted and sanitized

Probability the source address was spoofed

Obviously the 255.255.255.255 addresses are spoofed. Since the packets coming from the 207.18.175.10 host are using the same source port and coming at the same time, and interspersed, I'd say there is a reasonable possibility it is the hackers real address.

Description of attack

This attacker is looking for systems running LPRng, which are susceptible to the format string vulnerability (CAN-2000-0917). It appears to be a somewhat random spattering of address that doesn't seem to be following a pattern, however the time between hits is large enough to indicate that this could be an enormous network scan including more than the 5 class B address ranges indicated above.

Attack mechanism

These packets are obviously crafted with the source port hard coded to 31337. Some of these packets come from a (poorly) crafted source address of 255.255.255.255 and are therefore wasted packets, unless the attack comes from a machine on the local LAN. Since these log entries are from the border router, this attack is coming from the Internet. Since not all the addresses in a range are hit, this scan might be combined with a ping scan to only look for port 515 on machines that answer ping. When it works, this attack is designed to overflow a buffer and allow a remote user to execute arbitrary code, or crash the printing subsystem.

Correlation

CERT has issued advisory CA-2000-22 regarding Input Validation problems in LPRng. Details can be found at http://www.cert.org/advisories/CA-2000-22.html

Evidence of active targeting

This seems to be a random, or massive, attack.

Severity

(critical system + lethal attack) – (system + net countermeasures) = severity
(3 + 4) – (2 + 5) = 0

Defensive recommendation

The firewall already blocks this traffic

Multiple choice test question

If you don't know what a software package is or does, you should:

a. install it
b. learn what it does before making this decision
c. leave it off and wait until something breaks
d. turn it on and see if anyone breaks into it

Answer: b

Detect 6: SNMP scan

```
Apr 21 11:37:31 MDT denied udp 256.7.155.3 (45145) -> 10.7.174.6 (161) 1 packet
Apr 21 15:52:54 MDT denied udp 256.7.155.10 (44542) -> 10.7.174.6 (161) 1 packet
Apr 21 15:58:16 MDT denied udp 256.7.155.10 (44542) -> 10.7.174.6 (161) 7 packets
Apr 22 00:48:19 MDT denied udp 1.213.212.20 (42405) -> 10.7.174.6 (161) 1 packet
Apr 22 00:53:41 MDT denied udp 1.213.212.20 (42405) -> 10.7.174.6 (161) 7 packets
Apr 23 18:11:51 MDT denied udp 256.7.155.10 (60885) -> 10.7.174.6 (161) 7 packets
Apr 23 19:53:46 MDT denied udp 1.213.212.20 (57177) -> 10.7.174.6 (161) 7 packets
Apr 23 23:30:27 MDT denied udp 256.7.155.3 (63447) -> 10.7.174.6 (161) 1 packet
Apr 23 23:35:56 MDT denied udp 256.7.155.3 (63447) -> 10.7.174.6 (161) 7 packets
Apr 24 11:52:34 MDT denied udp 256.7.155.12 (35375) -> 10.7.174.6 (161) 1 packet
Apr 24 11:58:25 MDT denied udp 256.7.155.12 (35375) -> 10.7.174.6 (161) 7 packets
Apr 24 16:15:44 MDT denied udp 256.7.155.10 (53491) -> 10.7.174.6 (161) 7 packets
Apr 24 17:04:47 MDT denied udp 256.7.155.10 (37767) -> 10.7.174.6 (161) 7 packets
Apr 25 00:38:59 MDT denied udp 256.7.155.3 (59373) -> 10.7.174.6 (161) 1 packet
Apr 25 09:28:54 MDT denied udp 256.7.155.10 (53538) -> 10.7.174.6 (161) 7 packets
Apr 25 09:46:25 MDT denied udp 256.7.155.10 (61346) -> 10.7.174.6 (161) 1 packet
Apr 25 09:51:29 MDT denied udp 256.7.155.10 (61346) -> 10.7.174.6 (161) 7 packets
Apr 25 10:22:04 MDT denied udp 256.7.155.10 (39427) -> 10.7.174.6 (161) 7 packets
Apr 25 11:57:39 MDT denied udp 256.7.155.3 (46757) -> 10.7.174.6 (161) 1 packet
Apr 25 12:02:16 MDT denied udp 256.7.155.3 (46757) -> 10.7.174.6 (161) 7 packets
Apr 25 14:42:26 MDT denied udp 1.213.212.20 (33702) -> 10.7.174.6 (161) 7 packets
Apr 26 15:01:50 MDT denied udp 256.7.155.10 (64506) -> 10.7.174.6 (161) 1 packet
Apr 27 01:01:09 MDT denied udp 1.213.212.20 (47545) -> 10.7.174.6 (161) 1 packet
```

Source of trace

This log segment came from ISP border router logs

Detect was generated by

Cisco access list logs slightly reformatted and sanitized

Probability the source address was spoofed

Not likely. These are all routable addresses and they answer pings.

Description of attack

Obviously there are several machines trying to get information to or from the SNMP daemon running on the same target host. A possible motive is to exploit one of the many SNMP vulnerabilities: CAN-2000-0955, CAN-1999-517, etc.

Attack mechanism

Four remote hosts are trying to gain access to the SNMP port on the target machine. These attempts are continuing throughout the week at a fairly long period or manually. After a few minutes of speculation and DNS queries, it was determined that all of the remote hosts listed belonged to the same organization: the upstream bandwidth provider. It seems the upstream provider is attempting to

pull SNMP stats off this ISP's router without (I assume) prior arrangements.

Correlation

Initially, I thought this might be an attempt to use a default SNMP community name or exploit some other SNMP bug. Unfortunately, or fortunately depending on how you look at things, I was wrong.

Evidence of active targeting

It's pretty clear from the log the attack was directed at a certain host.

Severity

(critical system + lethal attack) – (system + net countermeasures) = severity
(5 + 5) – (5 + 5) = 0

Defensive recommendation

The firewall ACLs already block this port from the outside. All machines running SNMP should have changed the community strings and implemented ACLs where possible.

Multiple choice test question

SNMP is

a. a very bad thing – avoid it at all costs
b. a convenient tool for monitoring and managing enterprise equipment
c. full of bugs and not to be trusted
d. being replaced by XML

Answer: b

Detect 7: snmpXdmid exploit

```
[Note: 1.1.1.138 is the attacker, 2.2.2.130 is the victim]
19:45:34.871506 1.1.1.138.56224 > 2.2.2.130.111:  udp 56 (DF)
                        4500 0054 e619 4000 ff11 ab50 0101 018a
                        0202 0282 dba0 006f 0040 468b 3ae1 a979
                        0000 0000 0000 0002 0001 86a0 0000 0002
                        0000 0003 0000 0000 0000 0000 0000 0000
                        0000 0000 0001 8799 0000 0001 0000 0006
                        0000 0000
19:45:34.873383 2.2.2.130.111 > 1.1.1.138.56224:  udp 28 (DF)
                        4500 0038 e973 4000 ff11 a812 0101 0182
                        0101 018a 006f dba0 0024 cd5a 3ae1 a979
                        0000 0001 0000 0000 0000 0000 0000 0000
                        0000 0000 0000 87b1
19:45:34.875814 1.1.1.138.35292 > 2.2.2.130.34737: S 494413953:494413953(0) win 8760 <mss 1460>
(DF)
                        4500 002c e61a 4000 ff06 ab82 0101 018a
                        0202 0282 89dc 87b1 1d78 2881 0000 0000
                        6002 2238 3439 0000 0204 05b4 5555
19:45:34.875909 2.2.2.130.34737 > 1.1.1.138.35292: S 2274683232:2274683232(0) ack 494413954 win
8760 <mss 1460> (DF)
                        4500 002c e974 4000 ff06 a828 0101 0182
                        0101 018a 87b1 89dc 8794 e960 1d78 2882
                        6012 2238 c332 0000 0204 05b4
19:45:34.876230 1.1.1.138.35292 > 2.2.2.130.34737: . ack 1 win 8760 (DF)
                        4500 0028 e61b 4000 ff06 ab85 0101 018a
                        0202 0282 89dc 87b1 1d78 2882 8794 e961
                        5010 2238 daef 0000 5555 5555 5555
```

```
19:45:34.958869 1.1.1.138.35292 > 2.2.2.130.34737: P 1:1461(1460) ack 1 win 8760 (DF)
                        4500 05dc e61c 4000 ff06 a5d0 0101 018a
                        0202 0282 89dc 87b1 1d78 2882 8794 e961
                        5018 2238 b746 0000 0000 2324 3ae1 b286
                        0000 0000 0000 0002 0001 8799 0000 0001
                        0000 0101 0000 0001 0000 0020 3aea 20be
                        0000 0009 6c6f 6361 6c68 6f73 7400 0000
                        0000 0000 0000 0000 0000 0000 0000 0000
                        0000 0000 0000 0000 0000 0000 0000 0001
                        0000 0000 0000 0001 0000 0644 0000 0000
                        0000
19:45:34.959099 1.1.1.138.35292 > 2.2.2.130.34737: P 1461:2921(1460) ack 1 win 8760 (DF)
19:45:34.959217 2.2.2.130.34737 > 1.1.1.138.35292: . ack 1461 win 7300 (DF)
19:45:34.959260 2.2.2.130.34737 > 1.1.1.138.35292: . ack 2921 win 5840 (DF)
19:45:34.959889 1.1.1.138.35292 > 2.2.2.130.34737: . 2921:4381(1460) ack 1 win 8760 (DF)
19:45:34.960127 1.1.1.138.35292 > 2.2.2.130.34737: P 4381:5841(1460) ack 1 win 8760 (DF)
19:45:34.960294 1.1.1.138.35292 > 2.2.2.130.34737: . 5841:7301(1460) ack 1 win 8760 (DF)
[…snip… deleted 200 lines of pushing 1460 byte payloads, and the acks coming back]
19:47:13.278507 1.1.1.138.35292 > 2.2.2.130.34737: . 514461:515921(1460) ack 1 win 8760 (DF)
19:47:13.278732 1.1.1.138.35292 > 2.2.2.130.34737: . 515921:517381(1460) ack 1 win 8760 (DF)
19:47:13.278882 1.1.1.138.35292 > 2.2.2.130.34737: P 517381:518777(1396) ack 1 win 8760 (DF)
19:47:13.279022 2.2.2.130.34737 > 1.1.1.138.35292: . ack 517381 win 8760 (DF)
19:47:13.334029 2.2.2.130.34737 > 1.1.1.138.35292: . ack 518777 win 8760 (DF)
19:47:23.207061 1.1.1.138.35292 > 2.2.2.130.34737: P 518777:518791(14) ack 1 win 8760 (DF)
                        4500 0036 24b5 4000 ff06 6cde 0101 018a
                        0202 0282 89dc 87b1 1d80 12fa 8794 e961
                        5018 2238 cb15 0000 2f62 696e 2f75 6e61    uname -a
                        6d65 202d 610a
19:47:23.207341 1.1.1.138.35292 > 2.2.2.130.34737: P 518791:518792(1) ack 1 win 8760 (DF)
                        4500 0029 24b6 4000 ff06 6cea 0101 018a
                        0202 0282 89dc 87b1 1d80 1308 8794 e961
                        5018 2238 e658 0000 0a55 5555 5555
19:47:23.207461 1.1.1.138.35292 > 2.2.2.130.34737: P 518792:518795(3) ack 1 win 8760 (DF)
                        4500 002b 24b7 4000 ff06 6ce7 0101 018a
                        0202 0282 89dc 87b1 1d80 1309 8794 e961
                        5018 2238 7cf1 0000 6964 0a55 5555
19:47:23.207606 1.1.1.138.35292 > 2.2.2.130.34737: P 518795:518797(2) ack 1 win 8760 (DF)
                        4500 002a 24b8 4000 ff06 6ce7 0101 018a
                        0202 0282 89dc 87b1 1d80 130c 8794 e961
                        5018 2238 7949 0000 770a 5555 5555
19:47:23.207754 2.2.2.130.34737 > 1.1.1.138.35292: . ack 518792 win 8760 (DF)
                        4500 0028 ea24 4000 ff06 a77c 0202 0282
                        0101 018a 87b1 89dc 8794 e961 1d80 1309
                        5010 2238 f060 0000
19:47:23.207794 2.2.2.130.34737 > 1.1.1.138.35292: . ack 518797 win 8760 (DF)
                        4500 0028 ea25 4000 ff06 a77b 0202 0282
                        0101 018a 87b1 89dc 8794 e961 1d80 130e
                        5010 2238 f05b 0000
19:47:23.244657 2.2.2.130.34737 > 1.1.1.138.35292: P 1:58(57) ack 518797 win 8760 (DF)
                        4500 0061 ea26 4000 ff06 a741 0202 0282
                        0101 018a 87b1 89dc 8794 e961 1d80 130e
                        5018 2238 d39a 0000 5375 6e4f 5320 736c    SunOS sl
                        6963 6520 352e 3720 4765 6e65 7269 6320    ice 5.7 Generic
                        7375 6e34 6d20 7370 6172 6320 5355 4e57    sun4m sparc SUNW,
                        2c53 5041 5243 7374 6174 696f 6e2d 3230    SPARCstation-20
                        0a
19:47:23.245016 1.1.1.138.35292 > 2.2.2.130.34737: . ack 58 win 8760 (DF)
                        4500 0028 24b9 4000 ff06 6ce8 0101 018a
                        0202 0282 89dc 87b1 1d80 130e 8794 e99a
                        5010 2238 f022 0000 5555 5555 5555
19:47:33.757986 1.1.1.138.35292 > 2.2.2.130.34737: P 518797:518800(3) ack 58 win 8760 (DF)
                        4500 002b 24ba 4000 ff06 6ce4 0101 018a
                        0202 0282 89dc 87b1 1d80 130e 8794 e99a
```

```
                        5018 2238 7cb3 0000 6964 0a55 5555          id
19:47:33.795691 2.2.2.130.34737 > 1.1.1.138.35292: P 58:82(24) ack 518800 win 8760 (DF)
                        4500 0040 ea27 4000 ff06 a761 0202 0282
                        0101 018a 87b1 89dc 8794 e99a 1d80 1311
                        5018 2238 945a 0000 7569 643d 3028 726f    uid=0(ro
                        6f74 2920 6769 643d 3128 6f74 6865 7229    ot) gid=1(other)
19:47:33.836920 1.1.1.138.35292 > 2.2.2.130.34737: . ack 82 win 8760 (DF)
                        4500 0028 24bb 4000 ff06 6ce6 0101 018a
                        0202 0282 89dc 87b1 1d80 1311 8794 e9b2
                        5010 2238 f007 0000 5555 5555 5555
19:47:33.837019 2.2.2.130.34737 > 1.1.1.138.35292: P 82:83(1) ack 518800 win 8760 (DF)
                        4500 0029 ea28 4000 ff06 a777 0202 0282
                        0101 018a 87b1 89dc 8794 e9b2 1d80 1311
                        5018 2238 e5fe 0000 0a
19:47:33.886712 1.1.1.138.35292 > 2.2.2.130.34737: . ack 83 win 8760 (DF)
                        4500 0028 24bc 4000 ff06 6ce5 0101 018a
                        0202 0282 89dc 87b1 1d80 1311 8794 e9b3
                        5010 2238 f006 0000 5555 5555 5555
19:47:35.243024 1.1.1.138.35292 > 2.2.2.130.34737: P 518800:518802(2) ack 83 win 8760 (DF)
                        4500 002a 24bd 4000 ff06 6ce2 0101 018a
                        0202 0282 89dc 87b1 1d80 1311 8794 e9b3
                        5018 2238 78f2 0000 770a 5555 5555          w
19:47:35.287768 2.2.2.130.34737 > 1.1.1.138.35292: . ack 518802 win 8760 (DF)
                        4500 0028 ea29 4000 ff06 a777 0202 0282
                        0101 018a 87b1 89dc 8794 e9b3 1d80 1313
                        5010 2238 f004 0000
19:47:35.309072 2.2.2.130.34737 > 1.1.1.138.35292: P 83:212(129) ack 518802 win 8760 (DF)
                        4500 00a9 ea2a 4000 ff06 a6f5 0202 0282
                        0101 018a 87b1 89dc 8794 e9b3 1d80 1313
                        5018 2238 6c22 0000 2020 373a 3437 706d
                        2020 7570 2036 2064 6179 2873 292c 2031
                        363a 3530 2c20 2032 3020 7573 6572 732c
                        2020 6c6f 6164 2061 7665 7261 6765 3a20
                        302e 3134 2c20 302e 3138 2c20 302e 3139
                        0a55 7365 7220 2020 2020 7474 7920 2020
                        2020 2020 2020 2020 6c6f 6769 6e40 2020
                        6964
```

Source of trace

This log segment came from my test network

Detect was generated by

tcpdump –nlfxs 160

I added the bold characters at the right of the hex dumps

Probability the source address was spoofed

None. Authenticity of packets verified with phone call

Description of attack

This is the exploit for the Solaris snmpXdmid vulnerability (CAN-2001-0236) that I happened to catch running on our test network.

Attack mechanism

This attack is a buffer overflow that kills the running snmpXdmid process, creating a /core file, and leaves a /bin/sh running in it's place. The attacker gets interactive access to a root shell without a prompt and pretty much owns the system.

Correlation

According to http://www.securityfocus.com/bid/2417 and

http://www.cert.org/advisories/CA-2001-05.html this exploit usually comes with a root kit for covering the attacker's tracks.

Evidence of active targeting

In this case, a machine in the test lab ran rpcinfo –p against another machine in the lab. This trace did NOT catch that, but did catch the ensuing buffer overflow attack.

Severity

(critical system + lethal attack) – (system + net countermeasures) = severity
(1 + 5) – (3 + 5) = -2

Defensive recommendation

The border routers are already blocking 111, but it is possible to guess the port for snmpXdmid, if it is running. Since we don't run dmi, our site is not at risk for this.

Multiple choice test question

The above packet trace is an example of

a. buffer overflow
b. back door password usage
c. network scan for victims
d. ping sweep

Answer: a

Assignment 2: (30 points)
Describe the State of Intrusion Detection

Write a white paper on any single intrusion detection technology or challenge.

Who needs security?

A quick look at issues outside the office LAN

"…There is nothing on my home PC that a hacker would want…"

The facts

In a typical month, a small to medium ISP uses about 8 terabits of bandwidth, receives about 200,000 unsolicited, invalid, or unwanted port scans and/or intrusion attempts, and about 45,000 SPAM e-mail messages. As you might expect, all these port scans, attacks, and SPAM messages are not directed entirely at the ISP's server infrastructure; they are directed at the downstream address space, i.e. the ISP's customers (dialup, DSL, broadband, dedicated, everyone). The most common intrusion attempts are looking for NetBIOS, RPC, DNS, NFS, telnet, FTP, IMAP, and POP. The most common automated attacks are the SubSeven windows remote access Trojan, the ramen, and the lion worms.

Most ISP customers pay for connectivity and/or e-mail access. Something to note is that when an ISP customer can get to the Internet, other users on the Internet can get to them. So, as more people convert from the traditional dialup access method to DSL and broadband, they increase the window of opportunity for hackers to attack their homes.

A few details

These ports are not being scanned by accident or because some kid wants to test a new program. Hackers know how to remotely access (i.e. break into) computers using these services.

NetBIOS (ports 135 – 139) is the protocol that Microsoft systems use to communicate. It can be used to obtain machine names, user names, and shared system resources. It is used to look at and modify files on remote machines. It can also be used to look at and modify the registry on a remote machine. In short, Microsoft designed the networking functionality of Windows to allow for ease of use and workgroup functionality. Basically, it is very easy to get Windows machines to talk to each other. Security was not (and still is not) a big concern to the Microsoft way. In fact, the official response from Microsoft about how to deal with security is to install a firewall at your network border.

RPC (port 111 for UNIX and 135 for Microsoft), short for remote procedure call, is used to allow a network of computers to be used together in a "team" approach. It was designed as a convenient way to access resources, such as hard drives, CPUs, etc., on other computers. The basic idea is that when they start up, the sharable network processes register with the "portmapper." It works like an office building with no permanent cubicle assignments. As each occupant arrives for work, they grab a cubicle, then call the receptionist (the portmapper) and say, "I'm sitting in cubicle number 32775 today." Then as clients call the receptionist, they can be directed to the right phone. The theory is that if the receptionist doesn't answer the phone, no calls will be delivered. In practice, however, it doesn't always work that way. There are quite a few reported bugs and ways to remotely exploit these unpatched RPC services.

DNS (port 53), Domain Name System, is used to map names and URLs on the Internet to their respective IP addresses, which the computers need to communicate. Generally, DNS uses UDP datagrams for "quick" communications (i.e. getting IP addresses for names). TCP is only used for zone transfers (of entire domains), or when

the data is too large to fit in a UDP datagram. There are many published ways to break into all but the most recently released version of the BIND DNS server software.

NFS (usually port 2049), Network File System, is the standard method for sharing files between UNIX machines. NFS, like most other RPC services, has very little security built in, and is therefore very susceptible to remote file manipulation.

Telnet (23) and FTP (21) services are easy to run brute force attacks against. Hackers can just guess at passwords until finding something that works. Of course, the telnet and FTP protocols send username and password information across the network unencrypted. So, a hacker can run a network sniffer program to grab usernames and passwords from users who run telnet and FTP. Once a hacker has this information, it becomes much easier to target particular users' machines.

IMAP (143), POP3 (110), and POP2 (109) are protocols for remote access to e-mail. These protocols use simple username/password authentication, and can therefore be used to facilitate brute force attacks, or can be monitored by sniffers for password harvesting. There are also published ways to remotely exploit each of these services.

SubSeven is the most common trojan tromping around the Internet today. It is one of the few hacker software packages that has an ongoing development effort, it is very easy to use and extremely powerful. SubSeven allows a hacker to remotely control a victim's machine, talk through the speaker, flip the screen over, etc. It also has propagation software, such as port scanners, and can be configured in a master/slave hierarchy where the master machine can tell slaves what to do.

Worms typically spend their life cycle breaking into machines. Once they break in, most worms use all the resources available to them to break into as many other machines as they can. In some cases, worms have destructive payloads, and in some cases, all they do is cover their tracks and attempt to propagate. The most common worms on the Internet today, lion and ramen, break into UNIX machines through documented and long published holes. Other worms take advantage of the convenience features of Microsoft operating systems and the gullibility of users. Typically, these propagate via e-mail attachments, and some have destructive payloads.

But, what does a hacker want with my home PC?

Although it would probably be pretty easy to pull your tax return off your computer, most hackers are not interested in reading it. Some would be interested in helping you with your on-line banking, however. Since most on-line banking systems only require simple password authentication, and because Microsoft software is very helpful in "remembering" your passwords, access to your accounts would be pretty easy to obtain. Also, since most people use the same password for everything, hackers have been targeting home PCs as an avenue for accessing corporate networks. It is becoming more common for people to have VPN access from home to their networks at work. Since most home computers have very little security, it is much easier for hackers to get into corporate networks through these "back doors."

Most users can detect when they are accessing the Internet. If they are sitting in front of their machine, pulling up pages, they are access the Internet. We say "most users"

because many people don't realize that if they enable the auto-update feature of their anti-virus software (you do have anti virus software, don't you?) then their computer randomly accesses the Internet to get these updates, sometimes when the user is asleep. The big questions here are: How does a user know when a hacker attacks? How does a user know if their machine is a SubSeven slave? How does a user know if their machine is being used as a drone in a smurf attack?

Intrusion Detection Systems, either network based or host based, can keep track of this activity, and in some cases, curtail it. For a home network with one or two Microsoft systems, it is highly recommended to install personal firewall software, such as BlackICE Defender, or ZoneAlarm, or something similar. These packages are a combination of IDS and firewall software and are a very cost effective way to protect your systems from unauthorized use.

If your home or small office has a network, a small firewall appliance, may be a better option. One option is to setup a low-cost dual-NIC machine, (e.g. Linux), strip out all the network services, setup ipchains (or ipfw, etc), and use it as a firewall and network intrusion detection system. Even though this is a small capital investment, it may be well worth the investment to pay an expert to configure this environment, and provide enough training to be able to review the logs and recognize the warning signs of impending doom. The big question is, how much time and money will it cost to lose some or all of the network?

What about NAT?

Network Address Translation (NAT) allows a network to use "non-routable" addresses instead of more expensive, harder to obtain, routable address space. Please don't be fooled. NAT does not provide security, it is simply a way to avoid paying for expensive routable address space. It is trivial for a hacker to get to your machines in an insecure non-routable address environment.

So, now what?

It is time to start asking ISPs to provide more security options to their customers. An ISP can block known malicious network traffic at their border and save the headache of routing it, as well as saving the bandwidth of sending attack traffic to all their customers. It is also more cost effect for the ISP to build a large virus scanning system for e-mail, than for every down-stream customer to implement this functionality. Basically, how much is a customer willing to pay for a more secure upstream connection?

Bibliography

http://www.sans.org/ is a very helpful general source of information about security
http://www.securityfocus.com/ is another general source of security information
http://www.whitehats.com/library/worms/ is Max Vision's recent Internet worm library
http://www.simovits.com/nyheter9902.html is Simovits Consulting's list of trojan ports
http://www.linux-firewall-tools.com/linux/ports.html is a list of commonly probed ports
http://www.robertgraham.com/pubs/firewall-seen.html is a list of answers to commonly asked questions regarding firewalls
http://www.networkice.com/ is the company website for BlackICE Defender

http://www.zonelabs.com/ is the company website for ZoneAlarm

Assignment 3: (30 points)
”Analyze This” Scenario

Background Information

GIAC Enterprises, an e-business startup that sells electronic fortune cookies, henceforth to be referred to as the client, has asked me, henceforth known as the consultant, to provide a bid for security services based on an initial analysis of network traffic. The client provided various Network Intrusion Detection System (NIDS) data files to the consultant which were dated in February, 2000, March, 2000, January, 2001, February, 2001, and March, 2001 Since, the data provided by the client is missing several days of logs (for various reasons) the consultant is making a best effort to estimate the scope of this job and retains the right to modify terms if damage is significantly greater than expected. It should also be noted that the client did not provide a network diagram, or any other information about their network infrastructure. The estimates for work and hourly rates will be sent under separate cover and will not be attached to this document, so that this document can be used in the event you choose to solicit other bids for the work.

Executive Summary

The integrity of the network has been compromised. Several systems are running virus, worm, or trojan programs and are actively spreading to other machines, internal and external. These machines need to be addressed immediately and the rest of the systems and network infrastructure should be checked for malware. There is a Network Intrusion Detection System (NIDS), but the log files have been ignored. Systems have been down, broken, or dysfunctional for many days and have not been fixed. An enterprise monitoring system should be implemented that can actively notify the system administrator when error conditions exist. The system administrator also needs to periodically review the log files and act on issues that need attention. The network border(s) require current, verified, and validated dynamic packet filtering firewall appliance(s) with up-to-date packet filtering strategies implemented. Mail, FTP, web, and DNS servers should be appropriately shielded by the firewall and should be running current software and security configurations. Internal networks should be compartmentalized as much as possible to allow work groups to share data among themselves while preventing inappropriate access from other departments and users.

Methodology

The consultant reviewed the data provided from the client’s network. The basic methodology used was to parse through the alert files, and when necessary, use the other files for supporting evidence of malicious activities. The client provided about 140MB of snort data files to the consultant. These data files are dated in February, 2000, March, 2000, January, 2001, February, 2001, and March, 2001, as follows:

oos.2000-02-10	scan.2000-02-11	oos.2000-02-12	scan.2000-02-21
oos.2000-02-11	alert.2000-02-12	alert.2000-02-21	oos.2000-03-08

alert.2000-03-09	oos.2001-02-06	scan.2001-02-25	scan.2001-03-04
scan.2000-03-09	scan.2001-02-06	alert.2001-02-26	oos.2001-03-05
oos.2001-01-20	alert.2001-02-07	scan.2001-02-26	scan.2001-03-05
scan.2001-01-22	scan.2001-02-07	oos.2001-02-27	oos.2001-03-06
oos.2001-01-23	oos.2001-02-08	scan.2001-02-27	scan.2001-03-06
alert.2001-01-31	scan.2001-02-08	alert.2001-02-28	alert.2001-03-07
oos.2001-01-31	oos.2001-02-09	scan.2001-02-28	scan.2001-03-07
scan.2001-01-31	scan.2001-02-10	alert.2001-03-01	alert.2001-03-08
oos.2001-02-01	scan.2001-02-22	oos.2001-03-01	scan.2001-03-08
oos.2001-02-02	alert.2001-02-23	scan.2001-03-01	alert.2001-03-11
scan.2001-02-02	scan.2001-02-23	oos.2001-03-02	scan.2001-03-11
alert.2001-02-04	alert.2001-02-24	scan.2001-03-02	scan.2001-03-13
oos.2001-02-04	oos.2001-02-24	oos.2001-03-03	
alert.2001-02-05	scan.2001-02-24	scan.2001-03-03	
scan.2001-02-05	alert.2001-02-25	oos.2001-03-04	

The alert files were concatenated, sorted, then reviewed sequentially. Since there were 597,677 alerts to review and the consultant was not provided a network diagram or any information about server and network infrastructure, the consultant was forced to make several assumptions. Some of the assumptions include, but are not limited to:

- There is a firewall, or host based software, which prevents access to (at least) some systems based on known vulnerabilities
- There are systems running Microsoft Windows
- There are systems running Linux which are vulnerable to known and published exploits
- This is a very large network for an e-business startup selling fortune cookies

Several trends were identified in the alert logs, then investigated further in the findings section. In some instances, internal systems are exhibiting obvious behavior patterns consistent with machines infected with various viruses, worms, and Trojans.

Findings

The lion's share of the analyst's time focused on the trends and a thorough analysis of every single alert has been postponed until such time as the major problems can be dealt with and the consultant has a better understanding of the infrastructure. The primary trends found during the review of the alerts are listed below:

1. 36042 externally initiated port scans to 37260 TCP ports and 12341 UDP ports
2. 338461 Internally initiated port scans to 41553 TCP ports and 1979819 UDP ports
3. 1517 RPC targeted scans at port 111, 543 scans for the ghost portmapper at port 32771, and 112 **successful** connections to port 32771

```
02/11-00:08:00.575906  205.188.153.97:4000 -> MY.NET.221.246:32771
... 132 lines deleted ...
02/11-23:50:23.661861  205.188.153.97:4000 -> MY.NET.221.246:32771
02/20-03:41:17.557159  MY.NET.70.38:36338 -> MY.NET.103.112:32771
02/20-03:41:17.557209  MY.NET.70.38:36339 -> MY.NET.103.112:32771
02/20-03:41:17.557261  MY.NET.70.38:36340 -> MY.NET.103.112:32771
02/20-09:52:50.620251  24.9.158.233:22 -> MY.NET.163.17:32771
... 94 lines deleted ...
```

```
02/20-17:27:29.484838  24.9.158.233:22 -> MY.NET.163.17:32771
02/20-19:34:43.274146  129.105.107.190:1400 -> MY.NET.1.117:111
02/20-19:34:43.274210  129.105.107.190:1405 -> MY.NET.1.122:111
... 240 lines deleted ...
02/20-19:37:13.215976  129.105.107.190:3740 -> MY.NET.71.220:111
02/20-19:37:13.216140  129.105.107.190:3753 -> MY.NET.71.233:111
02/20-19:37:13.216191  129.105.107.190:3755 -> MY.NET.71.235:111
02/20-19:41:05.730067  171.65.61.201:1464 -> MY.NET.1.15:111
02/20-19:41:05.731385  171.65.61.201:1462 -> MY.NET.1.13:111
02/20-19:41:06.172737  171.65.61.201:1455 -> MY.NET.1.6:111
02/20-19:41:07.758966  171.65.61.201:2214 -> MY.NET.4.0:111
02/20-19:41:07.759014  171.65.61.201:2215 -> MY.NET.4.1:111
... 1257 lines deleted ...
02/20-19:50:27.762190  171.65.61.201:3566 -> MY.NET.253.125:111
02/20-19:50:27.801089  171.65.61.201:3827 -> MY.NET.254.131:111
02/20-19:50:27.802801  171.65.61.201:3837 -> MY.NET.254.141:111
02/20-19:50:27.841864  171.65.61.201:3558 -> MY.NET.253.117:111
02/20-19:50:27.841918  171.65.61.201:3559 -> MY.NET.253.118:111
01/30-14:00:10.320844  64.244.10.40:7777 -> MY.NET.223.254:32771
01/30-14:00:13.264842  64.244.10.40:7777 -> MY.NET.223.254:32771
... 358 lines deleted ...
01/30-14:12:15.135405  64.244.10.40:7777 -> MY.NET.223.254:32771
01/30-14:12:19.383302  64.244.10.40:7777 -> MY.NET.223.254:32771
01/30-14:34:29.280204  200.233.81.13:13765 -> MY.NET.60.17:32771
01/30-16:34:54.990563  205.188.153.108:4000 -> MY.NET.105.115:32771
... 4 lines deleted ...
01/30-18:24:20.548499  205.188.153.108:4000 -> MY.NET.105.115:32771
01/30-19:19:16.387947  24.9.203.188:61207 -> MY.NET.165.129:32771
01/30-22:42:37.801366  205.188.153.107:4000 -> MY.NET.97.217:32771
... 4 lines deleted ...
01/30-23:21:36.641709  205.188.153.107:4000 -> MY.NET.97.217:32771
02/03-22:17:09.957552  205.188.5.157:5190 -> MY.NET.98.227:32771
02/03-22:17:10.679807  205.188.5.157:5190 -> MY.NET.98.227:32771
02/22-07:53:23.593135  24.9.158.233:22 -> MY.NET.163.17:32771
... 3 lines deleted ...
02/22-14:53:26.320388  24.9.158.233:22 -> MY.NET.163.17:32771
02/25-16:47:04.317011  205.188.153.98:4000 -> MY.NET.224.230:32771
02/25-16:58:36.348916  205.188.153.98:4000 -> MY.NET.224.230:32771
02/25-17:10:45.435276  205.188.153.98:4000 -> MY.NET.224.230:32771
02/25-17:13:37.040887  205.188.153.109:4000 -> MY.NET.97.207:32771
02/25-17:29:25.730577  205.188.153.109:4000 -> MY.NET.97.207:32771
02/25-17:46:00.660027  205.188.153.98:4000 -> MY.NET.224.230:32771
02/25-17:48:10.498810  205.188.153.109:4000 -> MY.NET.97.207:32771
02/25-17:50:56.541091  205.188.153.98:4000 -> MY.NET.224.230:32771
02/25-21:44:50.037904  205.188.153.98:4000 -> MY.NET.224.230:32771
03/06-00:48:13.503963  209.88.124.3:4257 -> MY.NET.133.170:111
03/06-00:48:17.029343  209.88.124.3:4615 -> MY.NET.135.18:111
03/06-00:48:18.012440  209.88.124.3:4789 -> MY.NET.135.192:111
03/06-00:48:18.055797  209.88.124.3:4794 -> MY.NET.135.197:111
03/06-01:53:39.846281  216.136.171.195:1501 -> MY.NET.100.225:32771
... 14 lines deleted ...
03/06-01:53:39.923576  216.136.171.195:1501 -> MY.NET.100.225:32771
03/06-13:28:25.915564  205.188.153.105:4000 -> MY.NET.223.70:32771
... 11 lines deleted ...
03/06-20:59:57.694464  205.188.153.105:4000 -> MY.NET.223.70:32771
03/07-17:16:44.648225  199.174.56.66:3278 -> MY.NET.135.178:111
03/10-20:54:17.215127  152.163.241.90:5190 -> MY.NET.98.122:32771
03/10-20:54:17.919511  152.163.241.90:5190 -> MY.NET.98.122:32771
03/10-20:54:26.705542  152.163.241.90:5190 -> MY.NET.98.122:32771
```

4. 25 probes for the back orifice Trojan

```
02/24-17:04:09.754841  63.10.224.59:2382 -> MY.NET.97.3:31337
02/24-17:04:16.714295  63.10.224.59:2382 -> MY.NET.97.119:31337
02/24-17:04:19.102521  63.10.224.59:2382 -> MY.NET.97.162:31337
02/24-17:04:22.457194  63.10.224.59:2382 -> MY.NET.97.225:31337
02/24-17:04:24.335687  63.10.224.59:2382 -> MY.NET.98.3:31337
02/24-17:04:25.359418  63.10.224.59:2382 -> MY.NET.98.28:31337
```

```
02/24-17:04:27.815284  63.10.224.59:2382 -> MY.NET.98.75:31337
02/24-17:04:30.711389  63.10.224.59:2382 -> MY.NET.98.123:31337
02/24-17:04:36.800828  63.10.224.59:2382 -> MY.NET.98.238:31337
03/07-08:49:31.283316  203.170.152.87:31338 -> MY.NET.98.23:31337
03/07-08:49:31.349034  203.170.152.87:31338 -> MY.NET.98.35:31337
03/07-08:49:31.859244  203.170.152.87:31338 -> MY.NET.98.142:31337
03/07-08:49:31.876076  203.170.152.87:31338 -> MY.NET.98.144:31337
03/07-08:49:31.907963  203.170.152.87:31338 -> MY.NET.98.149:31337
03/07-08:49:31.970693  203.170.152.87:31338 -> MY.NET.98.157:31337
03/07-08:49:32.034901  203.170.152.87:31338 -> MY.NET.98.161:31337
03/07-08:49:32.246613  203.170.152.87:31338 -> MY.NET.98.188:31337
03/07-08:49:32.252468  203.170.152.87:31338 -> MY.NET.98.189:31337
03/07-08:49:32.252661  203.170.152.87:31338 -> MY.NET.98.190:31337
03/07-08:49:32.284515  203.170.152.87:31338 -> MY.NET.98.192:31337
03/07-08:49:32.284778  203.170.152.87:31338 -> MY.NET.98.193:31337
03/07-08:49:32.358145  203.170.152.87:31338 -> MY.NET.98.201:31337
03/07-08:49:32.358197  203.170.152.87:31338 -> MY.NET.98.203:31337
03/07-08:49:32.372500  203.170.152.87:31338 -> MY.NET.98.205:31337
03/07-08:49:32.385565  203.170.152.87:31338 -> MY.NET.98.207:31337
```

5. 469 system fingerprinted by queso
6. 1138 TCP connection attempts to DNS and lots of port scans for UDP DNS

```
02/06-17:13:10.643919  211.248.112.67:53 -> MY.NET.170.161:53
02/06-17:13:10.944278  211.248.112.67:53 -> MY.NET.170.176:53
02/06-17:13:11.044255  211.248.112.67:53 -> MY.NET.170.181:53
02/06-17:13:11.464773  211.248.112.67:53 -> MY.NET.170.202:53
02/06-17:13:11.524729  211.248.112.67:53 -> MY.NET.170.205:53
02/06-17:13:11.664704  211.248.112.67:53 -> MY.NET.170.212:53
02/06-17:13:11.703600  211.248.112.67:53 -> MY.NET.170.214:53
02/06-17:13:12.085368  211.248.112.67:53 -> MY.NET.170.233:53

Feb 10 01:08:09 MY.NET.100.230:32782 -> 205.188.185.18:53 UDP
Feb 10 01:08:09 MY.NET.100.230:32782 -> 216.200.206.140:53 UDP
Feb 10 01:08:10 MY.NET.100.230:32782 -> 147.9.1.9:53 UDP
... 4677 lines deleted ...
Feb  6 08:06:12 MY.NET.100.230:32782 -> 206.112.192.104:53 UDP
Feb  6 08:06:12 MY.NET.100.230:32782 -> 38.8.50.2:53 UDP
Feb  6 08:06:13 MY.NET.100.230:32782 -> 198.59.166.10:53 UDP
02/06-17:13:12.345610  211.248.112.67:53 -> MY.NET.170.246:53
```

7. 1155 “public” sessions to SNMP from two external and two internal machines to eight internal systems

```
02/22-12:01:08 SNMP public access [**] 128.46.156.197:1251 -> MY.NET.100.143:161
... 265 lines deleted
02/28-08:08:42 SNMP public access [**] 128.46.156.197:3843 -> MY.NET.100.206:161
02/27-10:29:28 SNMP public access [**] 128.46.156.197:1160 -> MY.NET.100.45:161
... 871 lines deleted
02/28-08:08:55 SNMP public access [**] 128.46.156.197:3855 -> MY.NET.100.99:161
02/20-10:33:55 SNMP public access [**] 128.183.38.30:1030 -> MY.NET.154.26:161
... 8 lines deleted ..
02/27-16:52:32 SNMP public access [**] 128.183.38.30:1030 -> MY.NET.154.26:161
01/30-00:01:03 SNMP public access [**] MY.NET.70.42:2155 -> MY.NET.50.154:161
02/03-00:01:04 SNMP public access [**] MY.NET.70.42:1156 -> MY.NET.50.154:161
02/03-00:01:05 SNMP public access [**] MY.NET.70.42:1156 -> MY.NET.50.154:161
02/03-00:04:29 SNMP public access [**] MY.NET.111.156:1737 -> MY.NET.50.154:161
02/03-00:04:30 SNMP public access [**] MY.NET.111.156:1737 -> MY.NET.50.154:161
```

8. 229 tiny fragments were detected aimed at twelve internal systems

```
01/30-00:35:05.719753  [**] Tiny Fragments 61.140.75.5 -> MY.NET.1.10
01/30-00:35:05.719854  [**] Tiny Fragments 61.140.75.5 -> MY.NET.1.10
01/30-00:46:35.731948  [**] Tiny Fragments 202.205.5.10 -> MY.NET.1.8
01/30-00:46:35.732041  [**] Tiny Fragments 202.205.5.10 -> MY.NET.1.8
01/30-04:00:03.304401  [**] Tiny Fragments 202.205.5.10 -> MY.NET.1.8
01/30-04:11:18.990423  [**] Tiny Fragments 202.205.5.10 -> MY.NET.1.8
01/30-07:26:05.596053  [**] Tiny Fragments 202.205.5.10 -> MY.NET.1.8
```

```
01/30-08:14:16.252161  [**] Tiny Fragments 202.96.96.3 -> MY.NET.1.10
01/30-08:14:16.252251  [**] Tiny Fragments 202.96.96.3 -> MY.NET.1.10
01/30-09:18:01.359282  [**] Tiny Fragments 202.101.43.220 -> MY.NET.1.10
01/30-09:18:01.359380  [**] Tiny Fragments 202.101.43.220 -> MY.NET.1.10
01/30-09:43:32.186863  [**] Tiny Fragments 61.155.13.3 -> MY.NET.1.10
01/30-10:24:28.285082  [**] Tiny Fragments 202.205.5.10 -> MY.NET.1.8
01/30-12:50:37.582483  [**] Tiny Fragments 111.111.111.111 -> MY.NET.20.10
01/30-12:52:01.851287  [**] Tiny Fragments 111.111.111.111 -> MY.NET.20.10
01/30-12:52:02.018028  [**] Tiny Fragments 127.0.0.1 -> MY.NET.20.10
01/30-14:59:36.822934  [**] Tiny Fragments 61.134.9.134 -> MY.NET.1.8
01/30-15:02:27.758724  [**] Tiny Fragments 61.140.75.3 -> MY.NET.1.8
01/30-15:18:57.560320  [**] Tiny Fragments 61.136.61.68 -> MY.NET.1.8
01/30-15:18:57.560365  [**] Tiny Fragments 61.136.61.68 -> MY.NET.1.8
01/30-16:37:37.001193  [**] Tiny Fragments 210.12.160.130 -> MY.NET.1.8
01/30-16:53:16.741168  [**] Tiny Fragments 202.96.96.3 -> MY.NET.1.8
01/30-17:01:53.791047  [**] Tiny Fragments 61.134.9.133 -> MY.NET.1.8
01/30-19:24:55.281169  [**] Tiny Fragments 202.96.96.3 -> MY.NET.1.8
01/30-19:24:55.281217  [**] Tiny Fragments 202.96.96.3 -> MY.NET.1.8
01/30-20:22:33.581963  [**] Tiny Fragments 61.134.9.133 -> MY.NET.1.8
02/04-02:50:46.103142  [**] Tiny Fragments 64.80.88.99 -> MY.NET.206.254
02/04-02:50:47.476166  [**] Tiny Fragments 64.80.88.99 -> MY.NET.206.254
02/04-02:50:48.097434  [**] Tiny Fragments 64.80.88.99 -> MY.NET.206.254
02/04-02:50:48.097484  [**] Tiny Fragments 64.80.88.99 -> MY.NET.206.254
02/04-02:50:48.295871  [**] Tiny Fragments 64.80.88.99 -> MY.NET.206.254
02/04-10:08:53.753512  [**] Tiny Fragments 64.80.90.84 -> MY.NET.160.109
02/04-10:21:24.148255  [**] Tiny Fragments 64.80.90.84 -> MY.NET.160.109
02/04-10:21:24.294591  [**] Tiny Fragments 64.80.90.84 -> MY.NET.160.109
02/04-11:44:08.012376  [**] Tiny Fragments 64.80.89.149 -> MY.NET.206.58
02/04-15:51:40.820197  [**] Tiny Fragments 64.80.90.55 -> MY.NET.160.109
02/04-15:51:40.960162  [**] Tiny Fragments 64.80.90.55 -> MY.NET.160.109
02/04-18:12:53.213115  [**] Tiny Fragments 64.80.90.36 -> MY.NET.98.117
... 51 lines deleted ...
02/04-18:13:57.641968  [**] Tiny Fragments 64.80.90.36 -> MY.NET.98.117
02/04-18:31:21.633706  [**] Tiny Fragments 64.80.90.36 -> MY.NET.97.231
... 18 lines deleted ...
02/04-18:31:44.909859  [**] Tiny Fragments 64.80.90.36 -> MY.NET.97.231
02/06-09:10:32.707874  [**] Tiny Fragments 64.80.89.149 -> MY.NET.228.10
02/22-21:25:23.575121  [**] Tiny Fragments 204.71.200.75 -> MY.NET.98.119
02/28-05:05:47.375953  [**] Tiny Fragments 206.207.108.116 -> MY.NET.205.242
03/06-01:35:45.983271  [**] Tiny Fragments 212.89.165.5 -> MY.NET.223.42
... 118 lines deleted ...
03/06-01:39:16.106940  [**] Tiny Fragments 212.89.165.5 -> MY.NET.223.42
```

9. a possible wu-ftp exploit on MY.NET.219.22
10. 591 connections from six internal machines to external printer ports

```
   1 MY.NET.162.71:2878 -> 209.249.182.79:515
   1 MY.NET.179.78:4036 -> 24.13.123.8:515
   1 MY.NET.201.170:2697 -> 209.50.66.2:515
  15 MY.NET.7.20:22 -> 216.88.97.58:515
  59 MY.NET.97.88:1025 -> 216.181.129.185:515
 514 MY.NET.98.190:1025 -> 216.181.129.185:515
```

11. more than 1000 hosts answered a single port scan at port 27374, which indicates an enormous infestation of the SubSeven trojan and/or the Ramen worm
12. 9914 alerts to possible RAMEN activity, and 4021 entries in the scan logs to or from port 27374
13. a number of internal machines are running the ramen worm and launching attacks at internal and external systems, e.g. .70.38, .207.250, .253.43, .223.42
14. 82777 scans detected with protocol errors in the packets
15. napster and gnutella are in wide spread use on the network

16. there are, or were, problems with one, or more, DHCP server(s) because there are 5303 alerts about machine(s) using 169.254/16 IP addresses
17. several machines are, or were, connected to internal LANs with misconfigured IP addresses
18. 376088 multicast packets to 8 different multicast addresses

Recommendations

The integrity of your network is clearly compromised. There are many systems behaving as if they are infected with viruses, worms, or trojans. If possible, immediate drastic action should be taken to prevent the further spread of these malicious software agents. Lots of work will be required to ensure all internal systems are cleaned, upgraded, and free of malware, and that appropriate network counter measures are in place to prevent future re-infestations.

1. Install and/or audit dynamic packet filtering firewall appliance(s) at network border(s). Configure packet filters at borders to block incoming and outgoing port scans, known trojan activity, DDoS activity, outbound internal source addresses, inbound internal source addresses, inbound SMTP (except to the mail relay), and other unused privileged ports. Block all RCP1918 non-routable address space, and perhaps a few commonly used, but unassigned addresses (e.g. 1.0.0.0/8). Also, block commonly exploited ports such as 2049 (NFS) and 6000 (X). Periodically run port scans against your own network and investigate any services that aren't expected. Also, if you block unassigned address space, make sure you periodically verify that it has not been assigned.
2. Configure internal routers with packet filters to ensure compartmentalization and prevent trojan, worm, and virus activities.
3. Isolate RPC services (port 111, and 32770 – 32900) with packet filters and replace portmapper and/or rpcbind with a tcp_wrappers enabled version from http://ftp.porcupine.org/pub/security/index.html.
4. Block all traffic to or from port 31337.
5. To prevent queso OS fingerprinting, block unused privileged ports and source ports between 1 and 5.
6. Ensure your DNS servers are running current software, and that they have appropriate access lists setup for zone transfers. All resource records should be small enough to fit in UDP packets. Block TCP connections on port 53 except between master and slave DNS servers.
7. Block SNMP at your network border and configure all systems running SNMP with appropriate community strings for read and write access. If you don't use SNMP, turn it off. Do not leave any community strings set to public or private.
8. Make sure the NIDS detects tiny fragments and notifies someone who can investigate. A NIDS is worthless if no one ever looks at the logs and alerts.
9. Block FTP traffic to all but sanctioned FTP servers that are running current software and security patches.
10. Block inbound and outbound printer subsystem traffic at the network border.
11. To prevent "trolling for trojans" such as SubSeven and Ramen, block inbound and outbound traffic to port 27374.

12. There are a number of machines that are definitely infected with the ramen worm, which must be cleaned off, or re-installed. All linux machines should have the current security patches installed to prevent another large-scale infestation of these worms, viruses, and trojans.
13. The following machines need to be shut down immediately and inspected for the ramen worm: .70.38, .207.250, .253.43, .223.42.
14. There were several machines (383) using IP addresses in the 169.254/16 range. This happens when Microsoft windows machines autoconfigure themselves after an unsuccessful attempt to obtain an address from a DHCP server.