Global Information Assurance Certification Paper

Reuben Rubio
GCIA Practical Assignment Version 2.9

Table of Contents

1.0 Network Detects

Trace 1 Bind RECON/Attack

1.1 Version Query

Bind is a well-known service that is known for a lot of vulnerabilities. DNS Servers provide hostname to ip address mapping for hosts connected to the Internet. When used correctly, DNS provides attackers precious information about organizations registration information, name servers for the domain, and even the administrative contact for social engineering.

Source of Trace: My administered NETWORK

SNORT

```
[**] MISC-DNS-version-query [**]
06/26-08:18:27.974657 200.207.120.46:3897-> my.net.work.187:53
UDP TTL:49 TOS:0x0 ID:7070 IpLen:20 DgmLen:58
Len: 38
```

TCPDUMP

08:18:27.974657 < 200.207.120.46.3897 > my.net.work.187.domain: 4660 [b2&3=0x80] TXT CHAOS)? version.bind. (30) (ttl 49, id 7070)
08:18:27.975803 < my.net.work.129.domain > 200.207.120.46.3897: 4660 ServFail 0/0/0 (30) (DF) (ttl 255, id 52035)

Trace 1.2 Inverse Query

Gauntlet Firewall

Jun 26 13:38:25 myfirewall.com unix: securityalert: no match found in local screen: TCP if=qe3 srcaddr=210.107.197.183 srcport=4552 dstaddr=my.net.work.163 dstport=53

SNORT

```
[**] IDS277 - NAMED Iquery Probe [**]
06/26-13:39:04.756157 210.107.197.183:4538->
my.net.work.113:53
UDP TTL:49 TOS:0x0 ID:32221 IpLen:20 DgmLen:51
Len: 31
```

```
[**] IDS277 - NAMED Iquery Probe [**]
06/26-13:39:04.794578 210.107.197.183:4539->
my.net.work.120:53
UDP TTL:49 TOS:0x0 ID:32223 IpLen:20 DgmLen:51
Len: 31
```

```
[**] IDS277 - NAMED Iquery Probe [**]
06/26-13:39:04.807129 210.107.197.183:4540->
```

```
my.net.work.122:53
UDP TTL:49 TOS:0x0 ID:32225 IpLen:20 DgmLen:51
Len: 31
[**] IDS277 - NAMED Iquery Probe [**]
06/26-13:39:04.818433 210.107.197.183:4541->
my.net.work.100:53
UDP TTL:49 TOS:0x0 ID:32227 IpLen:20 DgmLen:51
Len: 31
[**] IDS277 - NAMED Iquery Probe [**]
06/26-13:39:04.850386 210.107.197.183:4542->
my.net.work.106:53
UDP TTL:49 TOS:0x0 ID:32230 IpLen:20 DgmLen:51
Len: 31
[**] IDS277 - NAMED Iquery Probe [**]
06/26-13:39:04.870812 210.107.197.183:4543->
my.net.work.109:53
UDP TTL:49 TOS:0x0 ID:32233 IpLen:20 DgmLen:51
Len: 31
```

Probability the Address was spoofed:

Address was probably not spoofed because the attacker needs to see the response of the target system to see the expected result. Once the target is compromised, attacker needs to establish a session with the victim.

Description of Attack: Bind Inverse Query Vulnerability

Trace 1.1 shows that an attacker is querying a particular IP address for the Bind Version that the server is running. By doing so, the attacker is able to determine if there is a possibility of compromising the server by exploiting a particular bind vulnerability like the inverse query vulnerability or the zone transfer vulnerability.

Trace 1.2 shows that the attacker is trying to exploit the inverse query vulnerability by launching it against a range of IP Addresses hoping to actually hit a vulnerable name server.

SecurityFocus defines this vulnerability as "A buffer overflow exists in certain versions of BIND, the nameserver daemon currently maintained by the Internet Software Consortium (ISC). BIND fails to properly bound the data recieved when processing an inverse query. Upon a memory copy, portions of the program can be overwritten, and arbitrary commands run on the affected host."[1]

Trace 1.2 also indicates that the impatient attacker does not even need to query bind versions, he just launches the attack anyway hoping to hit a server that is vulnerable.

The CVE reference number for this vulnerability is CVE-1999-0009.

[1] http://www.securityfocus.com/vdb/bottom.html?section=discussion&vid=134

Name	CVE-1999-0009
Description	Inverse query buffer overflow in BIND 4.9 and BIND 8 Releases.

Correlation:
DNS attacks are one of the 10 most popular attacks launched in the Internet today. Candidates taking the Intrusion Detection Practical have time and time again encountered this attack and a lot of information can be found in the posted GCIA practicum's.

Evidence of Active Targeting:
The attacker in both traces seems to indicate that they are simply launching attacks blindly. If they really wanted to attack the server that is running bind, they could have done their homework first by checking the registry database (arin, ripe, or apnic). But then again, it may be an attempt to lead the analyst into thinking that it is a harmless dns probe in order to hide what is really happening in the background.

Severity:
Severity = (Criticality + Lethality) - (System Countermeasures + Network Countermeasures)
= (5 + 2) - (3 + 5)
= -1
Criticality = 5 because DNS Servers provide valuable service and information
Lethality = 2 because attack seems to indicate that it was more of a reconnaissance than an actual attempt against the actual DNS Server
System Countermeasures = 3 because system is not running latest version of bind
Network Countermeasures = 4 because network is protected by a firewall
Defensive Recommendation
Upgrade to Bind 9.1.3
Enforce ACLs on the border router to limit access to port 53. Since DNS server on the network is just querying, port 53/udp should only be outbound. Port 53/tcp should be blocked.
Multiple Choice Question:
When can port 53/tcp be blocked on the border routers?
a. Never
b. DNS Zone Transfers are not required
c. DNS is a caching server
d. DNS Server forwarding server

ANSWER: B - Port 53/tcp is only used for Zone Transfers

Trace 2 RPCBIND

Most applications talk to each other via RPC or remote procedure call. RPC uses a program called portmapper to listen and assign ports to running applications. RPCINFO, or it's Windows

counterpart rpcdump, queries port 111 to determine if the target machine is running services that can be exploited like rpc.cmsd, or rpc.statd, rpc.ttdbserverd.

Source of Trace: MY Administered NETWORK

Detect was generated by:

Gauntlet Firewall

Jun 27 23:48:01 my.network.com unix: securityalert: no match found in local screen: TCP if=qe3 srcaddr=209.235.8.94 srcport=2188 dstaddr=my.net.work.97 dstport=111
Jun 27 23:48:01 my.network.com unix: securityalert: no match found in local screen: TCP if=qe3 srcaddr=209.235.8.94 srcport=2188 dstaddr=my.net.work.97 dstport=111

SNORT

Two different sources are included in the table shown below. This table was generated by using Snortsnarf to process the alert file. This IP Addresses stands out because of the frequency of the attempts to connect to rpcbind. A closer look at the IP Address reveals that they are coming from the 209 IP Address belongs to a web hosting facility located in New York, www.dumbnews.com. The 200.54.185.51 IP Address belongs to a University in Chile, Universidad Autonoma Del Sur.

```
[**] RPC portmap request rstatd [**]
06/27-06:26:26.128684 200.54.185.51:872->
my.net.work.100:111
UDP TTL:48 TOS:0x0 ID:25352 IpLen:20 DgmLen:84
Len: 64
```

```
[**] RPC portmap request rstatd [**]
06/27-06:26:26.340134 200.54.185.51:873->
my.net.work.114:111
UDP TTL:48 TOS:0x0 ID:25361 IpLen:20 DgmLen:84
Len: 64
```

```
[**] RPC portmap request rstatd [**]
06/27-06:26:26.551136 200.54.185.51:874->
my.net.work.116:111
UDP TTL:48 TOS:0x0 ID:25365 IpLen:20 DgmLen:84
Len: 64
```

```
[**] RPC portmap request rstatd [**]
06/27-06:26:26.762137 200.54.185.51:875->
my.net.work.118:111
UDP TTL:48 TOS:0x0 ID:25368 IpLen:20 DgmLen:84
Len: 64
```

```
[**] RPC portmap request rstatd [**]
06/27-06:26:26.973636 200.54.185.51:876->
my.net.work.120:111
UDP TTL:48 TOS:0x0 ID:25373 IpLen:20 DgmLen:84
```

```
Len: 64
```

```
[**] RPC portmap request rstatd [**]
06/27-06:26:27.184667 200.54.185.51:877->
my.net.work.122:111
UDP TTL:48 TOS:0x0 ID:25377 IpLen:20 DgmLen:84
Len: 64
```

```
[**] RPC portmap request rstatd [**]
06/27-23:48:46.170747 209.235.8.94:875->
my.net.work.99:111
UDP TTL:55 TOS:0x0 ID:52189 IpLen:20 DgmLen:84
Len: 64
```

```
[**] RPC portmap request rstatd [**]
06/27-23:48:46.217385 209.235.8.94:876->
my.net.work.101:111
UDP TTL:55 TOS:0x0 ID:52199 IpLen:20 DgmLen:84
Len: 64
```

```
[**] RPC portmap request rstatd [**]
06/27-23:48:46.257963 209.235.8.94:877->
my.net.work.103:111
UDP TTL:55 TOS:0x0 ID:52202 IpLen:20 DgmLen:84
Len: 64
```

```
[**] RPC portmap request rstatd [**]
06/27-23:48:46.298243 209.235.8.94:878->
my.net.work.105:111
UDP TTL:55 TOS:0x0 ID:52205 IpLen:20 DgmLen:84
Len: 64
```

```
[**] RPC portmap request rstatd [**]
06/27-23:48:46.338735 209.235.8.94:879->
my.net.work.107:111
UDP TTL:55 TOS:0x0 ID:52208 IpLen:20 DgmLen:84
Len: 64
```

```
[**] RPC portmap request rstatd [**]
06/27-23:48:46.379150 209.235.8.94:880->
my.net.work.109:111
UDP TTL:55 TOS:0x0 ID:52211 IpLen:20 DgmLen:84
Len: 64
```

TCPDUMP

At this point, we need to highlight the differences between the Snort Alert and the Gauntlet Alert. The Gauntlet Alert is port 111/tcp, Snort is port 111/udp, both ports are

used by rpcbind. By using tcpdump to playback the traffic during this time period, we are able to determine that both alerts are correct. Probes to the udp port and the tcp port are indeed happening.

23:48:46.036215 209.235.8.94.2188 > my.net.work.97.sunrpc: S 2191294704:2191294704(0) win 32120 <mss 1460,sackOK,timestamp 51801432 0,nop,wscale 0> (DF)
23:48:46.036807 209.235.8.94.2190 > my.net.work.99.sunrpc: S 2198563057:2198563057(0) win 32120 <mss 1460,sackOK,timestamp 51801432 0,nop,wscale 0> (DF)
23:48:46.037574 my.net.work.99.sunrpc > 209.235.8.94.2190: S 3389103386:3389103386(0) ack 2198563058 win 10136 <nop,nop,timestamp 121974283 51801432,nop,wscale 0,mss 1460> (DF)
..... cut for brevity
23:48:46.202672 209.235.8.94.2190 > my.net.work.99.sunrpc: F 1:1(0) ack 1 win 32120 <nop,nop,timestamp 0 121974283> (DF)
23:48:46.203189 my.net.work.99.sunrpc > 209.235.8.94.2190: . ack 1 win 10136 <nop,nop,timestamp 121974299 51801446> (DF)
23:48:49.202513 209.235.8.94.2190 > my.net.work.99.sunrpc: F 1:1(0) ack 1 win 32120 <nop,nop,timestamp 51801751 121974299> (DF)
23:48:49.202887 my.net.work.99.sunrpc > 209.235.8.94.2190: . ack 2 win 10136 <nop,nop,timestamp 121974599 51801751> (DF)
23:48:49.203606 my.net.work.99.sunrpc > 209.235.8.94.2190: F 1:1(0) ack 2 win 10136 <nop,nop,timestamp 121974599 51801751> (DF)
23:48:49.226982 209.235.8.94.2190 > my.net.work.99.sunrpc: . ack 2 win 32120 <nop,nop,timestamp 51801753 121974599> (DF)

Probability the Address was spoofed:

Address is not spoofed because attacker needs to see the reply of the victim machine. What I find interesting though is the ephemeral ports of the two attackers are coming almost from the same range, 870-890. Could they be using the same tool? Could they be the same attacker? Probably. Or could they just be using the same tool? Probably.

Description of Attack:

RPCBIND can reveal a treasure chest of vulnerabilities. From buffer overflows, remote command execution, to actual attacker access that eventually leads to root user compromise.

Listed below are several references to such vulnerabilities related to RPCBIND or Portmapper.

www.cert.org/advisories/CA-1999-05.html

(CVE-1999-0018, CVE-1999-0019, CVE-1999-0493, CVE-2000-0666

Attack Mechanism:

Attacker queries rpcbind to identify rpc services that are running on target machine. Attacker goes back to exploit vulnerabilities depending on what what revealed by rpcinfo query on the target machine.

Correlation:

RPCBIND is also listed on in the SANS top 10. Please refer to http://www.sans.org/topten.htm for more information.

Evidence of Active Targeting:

There is no evidence of active targeting. However, the attacker is maliciously trying to break-in to machines in the network. These attackers do not care if they know your organization or not, as long as they see that you are vulnerable, my network is a likely target.

Severity:

Severity = (Criticality + Lethality) - (System Countermeasures + Network Countermeasures)

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

Criticality = 2 because it was more of a sweep than an actual attack to a live server

Lethality = 4 because the exploit can lead to a system compromise.

System Countermeasures = 4 because unnecessary services have been disable. Latest patches are installed on the system.

Network Countermeasures = 4 because hosts are protected by a firewall.

Defensive Recommendation

Block 111 on the border router. Make sure that latest and greatest patches are installed in the system. Replace RPC with secure rpc whenever possible or disable it if the applications on the server does not require rpc.

Multiple Choice Question:

23:48:46.036807 209.235.8.94.2190 > my.net.work.99.sunrpc: S 2198563057:2198563057(0) win 32120 <mss 1460,sackOK,timestamp 51801432 0,nop,wscale 0> (DF)
23:48:46.037574 my.net.work.99.sunrpc > 209.235.8.94.2190: S 3389103386:3389103386(0) ack 2198563058 win 10136 <nop,nop,timestamp 121974283 51801432,nop,wscale 0,mss 1460> (DF)
23:48:46.153526 209.235.8.94.2190 > my.net.work.99.sunrpc: . ack 1 win 32120 <nop,nop,timestamp 51801446 121974283> (DF)
23:48:46.170747 209.235.8.94.875 > my.net.work.99.sunrpc: udp 56
23:48:46.202672 209.235.8.94.2190 > my.net.work.99.sunrpc: F 1:1(0) ack 1 win 32120 <nop,nop,timestamp 0 121974283> (DF)
23:48:46.203189 my.net.work.99.sunrpc > 209.235.8.94.2190: . ack 1 win 10136 <nop,nop,timestamp 121974299 51801446> (DF)
23:48:49.202513 209.235.8.94.2190 > my.net.work.99.sunrpc: F 1:1(0) ack 1 win 32120 <nop,nop,timestamp 51801751 121974299> (DF)
23:48:49.202887 my.net.work.99.sunrpc > 209.235.8.94.2190: . ack 2 win 10136 <nop,nop,timestamp 121974599 51801751> (DF)
23:48:49.203606 my.net.work.99.sunrpc > 209.235.8.94.2190: F 1:1(0) ack 2 win 10136 <nop,nop,timestamp 121974599 51801751> (DF)

23:48:49.226982 209.235.8.94.2190 > my.net.work.99.sunrpc: . ack 2 win 32120 <nop,nop,timestamp 51801753 121974599> (DF)

What is wrong in the trace shown above?

a. udp 56

b. ack 2 - no such thing

c. too many Syn's

d. One too many Fin's

ANSWER: D - It normally takes four segments to terminate a tcp connection. A FIN from the server, and ACK from the Client, a FIN from the client, and finally, and ACK from the server. What could cause this session to have one extra FIN? What can this be used for? Or is this just a retransmit?

Trace 3 Scanning for Port 8080 - Proxy Scans

Source of Trace: MY Administered NETWORK

Detect was generated by:

Gauntlet Firewall

This is scary, firewall logs does not indicate any security violation. It's either the Firewall Security Policies were not violated, meaning packet was accepted and delivered to destination or logs were simply not written on /var/log/messages.
Firewall policies reveal that there should have been a security violation. Not sure why logs were not created for this event.

SNORT

```
[**] SCAN Proxy attempt [**]
06/26-22:19:38.266178 64.20.213.222:1064->
my.net.work.98:8080
TCP TTL:116 TOS:0x0 ID:17666 IpLen:20 DgmLen:48 DF
******S* Seq: 0x1A7F03 Ack: 0x0 Win: 0x16D0 TcpLen: 28
TCP Options (4) => MSS: 536 NOP NOP SackOK
```

```
[**] SCAN Proxy attempt [**]
06/26-22:19:38.342312 64.20.213.222:1065->
my.net.work.98:8080
TCP TTL:116 TOS:0x0 ID:18178 IpLen:20 DgmLen:48 DF
******S* Seq: 0x1A7F2C Ack: 0x0 Win: 0x16D0 TcpLen: 28
```

```
TCP Options (4) => MSS: 536 NOP NOP SackOK
```

```
[**] SCAN Proxy attempt [**]
06/26-22:19:39.594965 64.20.213.222:1064->
my.net.work.98:8080
TCP TTL:116 TOS:0x0 ID:20482 IpLen:20 DgmLen:48 DF
******S* Seq: 0x1A7F03 Ack: 0x0 Win: 0x16D0 TcpLen: 28
TCP Options (4) => MSS: 536 NOP NOP SackOK
```

```
[**] SCAN Proxy attempt [**]
06/26-22:19:39.693077 64.20.213.222:1065->
my.net.work.98:8080
TCP TTL:116 TOS:0x0 ID:20738 IpLen:20 DgmLen:48 DF
******S* Seq: 0x1A7F2C Ack: 0x0 Win: 0x16D0 TcpLen: 28
TCP Options (4) => MSS: 536 NOP NOP SackOK
```

```
[**] SCAN Proxy attempt [**]
06/26-22:19:40.987951 64.20.213.222:1064->
my.net.work.98:8080
TCP TTL:116 TOS:0x0 ID:23554 IpLen:20 DgmLen:48 DF
******S* Seq: 0x1A7F03 Ack: 0x0 Win: 0x16D0 TcpLen: 28
TCP Options (4) => MSS: 536 NOP NOP SackOK
```

```
[**] SCAN Proxy attempt [**]
06/26-22:19:41.057942 64.20.213.222:1065->
my.net.work.98:8080
TCP TTL:116 TOS:0x0 ID:23810 IpLen:20 DgmLen:48 DF
******S* Seq: 0x1A7F2C Ack: 0x0 Win: 0x16D0 TcpLen: 28
TCP Options (4) => MSS: 536 NOP NOP SackOK
```

```
[**] SCAN Proxy attempt [**]
06/26-22:19:42.394917 64.20.213.222:1064->
my.net.work.98:8080
TCP TTL:116 TOS:0x0 ID:26114 IpLen:20 DgmLen:48 DF
******S* Seq: 0x1A7F03 Ack: 0x0 Win: 0x16D0 TcpLen: 28
TCP Options (4) => MSS: 536 NOP NOP SackOK
```

```
[**] SCAN Proxy attempt [**]
06/26-22:19:42.401905 64.20.213.222:1065->
my.net.work.98:8080
TCP TTL:116 TOS:0x0 ID:26370 IpLen:20 DgmLen:48 DF
******S* Seq: 0x1A7F2C Ack: 0x0 Win: 0x16D0 TcpLen: 28
TCP Options (4) => MSS: 536 NOP NOP SackOK
```

TCPDUMP

```
22:19:38.266178 64.20.213.222.1064 > my.net.work.98.8080: S
1736451:1736451(0) win 5840 <mss 536,nop,nop,sackOK> (DF) (ttl
116, id 17666)
```

```
22:19:38.266798 my.net.work.98.8080 > 64.20.213.222.1064: R
0:0(0) ack 1736452 win 0 (DF) (ttl 116, id 57955)
22:19:38.342312 64.20.213.222.1065 > my.net.work.98.8080: S
1736492:1736492(0) win 5840 <mss 536,nop,nop,sackOK> (DF) (ttl
116, id 18178)
22:19:38.342687 my.net.work.98.8080 > 64.20.213.222.1065: R
0:0(0) ack 1736493 win 0 (DF) (ttl 116, id 57956)
22:19:39.594965 64.20.213.222.1064 > my.net.work.98.8080: S
1736451:1736451(0) win 5840 <mss 536,nop,nop,sackOK> (DF) (ttl
116, id 20482)
22:19:39.595488 my.net.work.98.8080 > 64.20.213.222.1064: R
0:0(0) ack 1 win 0 (DF) (ttl 116, id 57957)
22:19:39.693077 64.20.213.222.1065 > my.net.work.98.8080: S
1736492:1736492(0) win 5840 <mss 536,nop,nop,sackOK> (DF) (ttl
116, id 20738)
22:19:39.693434 my.net.work.98.8080 > 64.20.213.222.1065: R
0:0(0) ack 1 win 0 (DF) (ttl 116, id 57958)
22:19:40.987951 64.20.213.222.1064 > my.net.work.98.8080: S
1736451:1736451(0) win 5840 <mss 536,nop,nop,sackOK> (DF) (ttl
116, id 23554)
22:19:40.988505 my.net.work.98.8080 > 64.20.213.222.1064: R
0:0(0) ack 1 win 0 (DF) (ttl 116, id 57959)
22:19:41.057942 64.20.213.222.1065 > my.net.work.98.8080: S
1736492:1736492(0) win 5840 <mss 536,nop,nop,sackOK> (DF) (ttl
116, id 23810)
22:19:41.058254 my.net.work.98.8080 > 64.20.213.222.1065: R
0:0(0) ack 1 win 0 (DF) (ttl 116, id 57960)
22:19:42.394917 64.20.213.222.1064 > my.net.work.98.8080: S
1736451:1736451(0) win 5840 <mss 536,nop,nop,sackOK> (DF) (ttl
116, id 26114)
22:19:42.395310 my.net.work.98.8080 > 64.20.213.222.1064: R
0:0(0) ack 1 win 0 (DF) (ttl 116, id 57961)
22:19:42.401905 64.20.213.222.1065 > my.net.work.98.8080: S
1736492:1736492(0) win 5840 <mss 536,nop,nop,sackOK> (DF) (ttl
116, id 26370)
22:19:42.402360 my.net.work.98.8080 > 64.20.213.222.1065: R
0:0(0) ack 1 win 0 (DF) (ttl 116, id 57962)
```

Probability the Address was spoofed:

Zero, attacker needs to see the response in order to determine if the target machine is functioning as a proxy server.

Description of Attack:

This was not an attack, it was more of a reconnaissance effort done by that attacker to see if port 8080 was open. Port 8080 seems to be assiociated with Proxy Services. It is very likely that the

attacker was looking for open proxies that can be used to anonymize their requests. In doing so, they are hoping to cover their tracks.

Attempts to connect to port 8080 may also be an indication of an attempt to exploit vulnerabilities against the Microsoft Proxy Server, and other proxy servers mentioned below.

Attack Mechanism:

Name	CAN-2001-0239 (under review)
Description	Microsoft Internet Security and Acceleration (ISA) Server 2000 Web Proxy allows remote attackers to cause a denial of service, and possibly execute arbitrary commands, via a long web request with a specific type.
References	BUGTRAQ:20010416 [SX-20010320-2] - Microsoft ISA Server Denial of Service URL:http://www.securityfocus.com/archive/1/176912 BUGTRAQ:20010427 Microsoft ISA Server Vulnerability URL:http://www.securityfocus.com/archive/1/179986 BUGTRAQ:20010417 [SX-20010320-2b] - Followup re. Microsoft ISA Server Denial of Service URL:http://www.securityfocus.com/archive/1/177160 MS:MS01-021 URL:http://www.microsoft.com/technet/security/bulletin/MS01-021.asp BID:2600 URL:http://www.securityfocus.com/bid/2600

bugtraq id	2600
class	Boundary Condition Error
cve	CAN-2001-0239

Name	CVE-2001-0129
Description	Buffer overflow in Tinyproxy HTTP proxy 1.3.3 and earlier allows remote attackers to cause a denial of service and possibly execute arbitrary commands via a long connect request.

Name	CVE-2000-0416
Description	NTMail 5.x allows network users to bypass the NTMail proxy restrictions by redirecting their requests to NTMail's web configuration server.

Name	CVE-2000-0165
Description	The Delegate application proxy has several buffer overflows which allow a remote attacker to execute commands.

Correlation:

As of July 18, 2001, attacks on Port 8080 is the to 14th attack in www.incidents.org.

Evidence of Active Targeting:

The attacker seemed to know that my.net.work.98 was a proxy server. DNS information shows that the server is an anonymous ftp server hosted by navipath.net. This is definitely active targeting.

Severity:

Severity = (Criticality + Lethality) - (System Countermeasures + Network Countermeasures)

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

= -3

Criticality = 2 because it was more of a sweep than an actual attack to a live server

Lethality = 4 because the exploit can lead to a system compromise.

System Countermeasures = 4 because unnecessary services have been disabled. Latest patches are installed on the system. Port 8080 is only accepting requests from the internal network and is enforcing authentication.

Network Countermeasures = 4 because hosts are protected by a firewall.

Defensive Recommendation

Block 8080 in the border router. Review Firewall configuration to determine why logs were not generated by the possible security violation. Upgrade to the latest and greatest version of the Firewall because we are running three versions behind already,

Multiple Choice Question:

What does 8080/tcp and 3128/tcp have in common?

a. TCP

b. Used for Proxy Services

c. Can be used as an anonymizer

d. All of the above

e. Nothing

ANSWER: D - 8080 and 3128 are both using the TCP protocol, both are used for http proxy services, and if left open and configured incorrectly, they can be used to anonymize the requests.

Trace 4 Anonymous FTP Attempt

Source of Trace: MY NETWORK

Detect was generated by:

Gauntlet Firewall

```
Jun 26 22:17:04 my.network.com unix: securityalert: no match
found in local screen: TCP if=qe3 srcaddr=213.44.212.89
srcport=3022 dstaddr=my.net.work.163 dstport=21
Jun 26 22:17:04 my.network.com ftp-gw[27081]: deny
host=nodnsquery/213.44.212.89 connect to my.net.work.161
Jun 26 22:17:04 my.network.com ftp-gw[27082]: deny
host=nodnsquery/213.44.212.89 connect to 0.0.0.0
Jun 26 22:17:04 my.network.com ftp-gw[27083]: deny
host=nodnsquery/213.44.212.89 connect to my.net.work.164
Jun 26 22:17:04 my.network.com ftp-gw[27084]: deny
host=nodnsquery/213.44.212.89 connect to my.net.work.162
Jun 26 22:17:05 my.network.com ftp-gw[27085]: deny
host=nodnsquery/213.44.212.89 connect to my.net.work.165
Jun 26 22:17:05 my.network.com ftp-gw[27086]: deny
cut for brevity .....
Jun 26 22:17:05 my.network.com ftp-gw[27110]: deny
host=nodnsquery/213.44.212.89 connect to 0.0.0.0
Jun 26 22:17:05 my.network.com ftp-gw[27111]: deny
host=nodnsquery/213.44.212.89 connect to my.net.work.189
Jun 26 22:17:07 my.network.com ftp-gw[27088]: deny
host=nodnsquery/213.44.212.89 connect to ftp.microsoft.com
Jun 26 22:17:07 my.network.com ftp-gw[27088]: exit
host=nodnsquery/213.44.212.89 cmds=1 in=0 out=0 user=unauth
duration=2
Jun 26 22:17:07 my.network.com ftp-gw[27089]: deny
host=nodnsquery/213.44.212.89 connect to ftp.microsoft.com
Jun 26 22:17:07 my.network.com ftp-gw[27089]: exit
host=nodnsquery/213.44.212.89 cmds=1 in=0 out=0 user=unauth
duration=2
Jun 26 22:17:07 my.network.com ftp-gw[27090]: deny
host=nodnsquery/213.44.212.89 connect to ftp.microsoft.com
```

```
Jun 26 22:17:07 my.network.com ftp-gw[27090]: exit
host=nodnsquery/213.44.212.89 cmds=1 in=0 out=0 user=unauth
duration=2
Jun 26 22:17:13 my.network.com unix: securityalert: no match
found in local screen: TCP if=qe3 srcaddr=213.44.212.89
srcport=3022 dstaddr=my.net.work.163 dstport=21
Jun 27 00:17:04 my.network.com ftp-gw[27082]: exit
host=nodnsquery/213.44.212.89 cmds=0 in=0 out=0 user=unauth
duration=7200
Jun 27 00:17:05 my.network.com ftp-gw[27110]: exit
host=nodnsquery/213.44.212.89 cmds=0 in=0 out=0 user=unauth
duration=7200
```

This part of the detect shows that the attacker was trying to scan for servers running ftp. After exhausting his list, the attacker then tries to use the ftp proxy to connect to ftp.microsoft.com. Take note of the pid ftp-gw process id 27082. This shows that the attacker tried to connect to the gateway itself and the session ended after 7200 seconds. The ftp attempt was denied but it still tied up 1 ftp process.

This could have been successfully used as a denial of service attack against the ftp service running on my firewall but I guess, that was not the attackers intention. He simply wanted to use an anonymous ftp server to connect to Microsoft's site. At least, that is what the logs say.

We have three types of Gauntlet Alerts in this section.

```
Jun 26 22:17:04 my.network.com unix: securityalert: no match
found in local screen: TCP if=qe3 srcaddr=213.44.212.89
srcport=3022 dstaddr=my.net.work.163 dstport=21
```

This type of alert is generated by a packet screen violation. This is normally generated by packets hitting the ip address of one of the interfaces of the firewall.

The other type of alert is generated by the proxies running on the firewall.

```
Jun 26 22:17:04 my.network.com ftp-gw[27083]: deny
host=nodnsquery/213.44.212.89 connect to my.net.work.164
```

Accepts were intentionally left out due to security reasons. Suffice it to say that some ftp attempts were accepted by the proxy but was denied later on when the attacker tried to connect to ftp.microsoft.com.

```
Jun 26 22:17:07 my.network.com ftp-gw[27089]: deny
host=nodnsquery/213.44.212.89 connect to ftp.microsoft.com
```

The number enclosed in brackets is the process identifier for the ftp session.

SNORT

```
[**] INFO FTP anonymous FTP [**]
06/26-22:17:48.615168 213.44.212.89:3027->
my.net.work.168:21
```

```
TCP TTL:112 TOS:0x0 ID:8297 IpLen:20 DgmLen:74
***AP*** Seq: 0x37F9CD9 Ack: 0x82D8E91 Win: 0x7F9B TcpLen:
20
```

```
[**] INFO FTP anonymous FTP [**]
06/26-22:17:48.702484 213.44.212.89:3028->
my.net.work.169:21
TCP TTL:112 TOS:0x0 ID:8553 IpLen:20 DgmLen:74
***AP*** Seq: 0x37F9CDD Ack: 0x82DA446 Win: 0x7F9B TcpLen:
20
```

```
[**] INFO FTP anonymous FTP [**]
06/26-22:17:48.803983 213.44.212.89:3029->
my.net.work.170:21
TCP TTL:112 TOS:0x0 ID:8809 IpLen:20 DgmLen:74
***AP*** Seq: 0x37F9CDF Ack: 0x82E3F67 Win: 0x7F9B TcpLen:
20
```

```
[**] INFO FTP anonymous FTP [**]
06/26-22:17:48.926489 213.44.212.89:3030->
my.net.work.171:21
TCP TTL:112 TOS:0x0 ID:9577 IpLen:20 DgmLen:74
***AP*** Seq: 0x37F9CE1 Ack: 0x82E7560 Win: 0x7F9B TcpLen:
20
```

```
[**] INFO FTP anonymous FTP [**]
06/26-22:17:50.030472 213.44.212.89:3041->
my.net.work.182:21
TCP TTL:112 TOS:0x0 ID:24169 IpLen:20 DgmLen:74
***AP*** Seq: 0x37F9CF9 Ack: 0x8390F9C Win: 0x7F9B TcpLen:
20
```

```
[**] INFO FTP anonymous FTP [**]
06/26-22:17:51.130457 213.44.212.89:3025->
my.net.work.166:21
TCP TTL:112 TOS:0x0 ID:30313 IpLen:20 DgmLen:74
***AP*** Seq: 0x37F9CD5 Ack: 0x82A2135 Win: 0x7F9B TcpLen:
20
```

```
[**] INFO FTP anonymous FTP [**]
06/26-22:17:51.188984 213.44.212.89:3026->
my.net.work.167:21
TCP TTL:112 TOS:0x0 ID:30825 IpLen:20 DgmLen:74
***AP*** Seq: 0x37F9CD7 Ack: 0x82C02AF Win: 0x7F9B TcpLen:
20
```

TCPDUMP

```
22:17:45.955582 213.44.212.89.3028 > my.net.work.169.ftp: S
58694876:58694876(0) win 32768 <mss 1460,nop,nop,eol> (ttl 112,
```

```
id 35688)
4500 0030 8b68 0000 7006 0e6d d52c d459  |  E . . 0 . h . . p .
. m . , . Y
d0c3 36a9 0bd4 0015 037f 9cdc 0000 0000  |  . . 6 . . . . . . .
. . . . . .
7002 8000 a9e9 0000 0204 05b4 0101 0000  |  p . . . . . . . . .
. . . . . .
22:17:45.956062 my.net.work.169.ftp > 213.44.212.89.3028: S
137208800:137208800(0) ack 58694877 win 8760 <mss 1460> (DF)
(ttl 255, id 11150)
4500 002c 2b8e 4000 ff06 9f4a d0c3 36a9  |  E . . , + . @ . . .
. J . . 6 .
d52c d459 0015 0bd4 082d a3e0 037f 9cdd  |  . , . Y . . . . . -
. . . . . .
6012 2238 6c98 0000 0204 05b4 0000 ----  |  ` . " 8 l . . . . .
. . . . . .
22:17:46.296798 213.44.212.89.3028 > my.net.work.169.ftp: . ack
1 win 32768 (ttl 112, id 52328)
4500 0028 cc68 0000 7006 cd74 d52c d459  |  E . . ( . h . . p .
. t . , . Y
d0c3 36a9 0bd4 0015 037f 9cdd 082d a3e1  |  . . 6 . . . . . . .
. . . - . .
5010 8000 268d 0000 0204 05b4 0101 ----  |  P . . . & . . . . .
. . . . . .
22:17:46.525403 my.net.work.169.ftp > 213.44.212.89.3028: P
1:49(48) ack 1 win 8760 (DF) (ttl 255, id 11181)
4500 0058 2bad 4000 ff06 9eff d0c3 36a9  |  E . . X + . @ . . .
. . . . 6 .
d52c d459 0015 0bd4 082d a3e1 037f 9cdd  |  . , . Y . . . . . -
. . . . . .
5018 2238 5952 0000 3530 3120 4e6f 7420  |  P . " 8 Y R . . 5 0
1   N o t
7065 726d 6974 7465 6420 746f 2063 6f6e  |  p e r m i t t e d
t o   c o n
6e65 6374 2074 6f20 3230 382e 3139 352e  |  n e c t   t o   m y
. n e t .
3534 2e31 3639 0d0a ---- ---- ---- ----  |  w o r k  . 1 6 9
\r\n. . . . . 22:17:46.834796 213.44.212.89.3028 >
my.net.work.169.ftp: . ack 49 win 32720 (ttl 112, id 60264)
4500 0028 eb68 0000 7006 ae74 d52c d459  |  E . . ( . h . . p .
. t . , . Y
d0c3 36a9 0bd4 0015 037f 9cdd 082d a411  |  . . 6 . . . . . . .
. . . - . .
5010 7fd0 268d 0000 0204 05b4 0101 ----  |  P . . . & . . . . .
. . . . . .
22:17:46.835249 my.net.work.169.ftp > 213.44.212.89.3028: P
49:102(53) ack 1 win 8760 (DF) (ttl 255, id 11201)
```

```
4500 005d 2bc1 4000 ff06 9ee6 d0c3 36a9  |  E . . ] + . @ . . .
. . . . 6 .
d52c d459 0015 0bd4 082d a411 037f 9cdd  |  . , . Y . . . . . -
. . . . . .
5018 2238 0a97 0000 3232 3020 6e73 6363  |  P . " 8 \n. . . 2 2
0   m y.
6677 2e67 7363 632e 636f 6d20 4654 5020  |  h o s t  . n a m e
. c o m  F T P
7072 6f78 7920 2856 6572 7369 6f6e 2056  |  p r o x y   ( V e r
s i o n   V
342e 3229 2072 6561 6479 2e0d 0a-- ----  |  4 . 2 )   r e a d y
. \r\n. . .
22:17:47.145293 213.44.212.89.3028 > my.net.work.169.ftp: . ack
102 win 32667 (ttl 112, id 65384)
4500 0028 ff68 0000 7006 9a74 d52c d459  |  E . . ( . h . . p .
. t . , . Y
d0c3 36a9 0bd4 0015 037f 9cdd 082d a446  |  . . 6 . . . . . . .
. . . - . F
5010 7f9b 268d 0000 0204 05b4 0101 ----  |  P . . . & . . . . .
. . . . . .
22:17:48.702484 213.44.212.89.3028 > my.net.work.169.ftp: P
1:35(34) ack 102 win 32667 (ttl 112, id 8553)
4500 004a 2169 0000 7006 7852 d52c d459  |  E . . J ! i . . p .
x R . , . Y
d0c3 36a9 0bd4 0015 037f 9cdd 082d a446  |  . . 6 . . . . . . .
. . . - . F
5018 7f9b 67bb 0000 5553 4552 2061 6e6f  |  P . . . g . . . U S
E R   a n o
6e79 6d6f 7573 4066 7470 2e6d 6963 726f  |  n y m o u s @ f t p
. m i c r o
736f 6674 2e63 6f6d 0d0a ---- ---- ----  |  s o f t . c o m
\r\n. . . . . .
22:17:48.703000 my.net.work.169.ftp > 213.44.212.89.3028: . ack
35 win 8760 (DF) (ttl 255, id 11235)
4500 0028 2be3 4000 ff06 9ef9 d0c3 36a9  |  E . . ( + . @ . . .
. . . . 6 .
d52c d459 0015 0bd4 082d a446 037f 9cff  |  . , . Y . . . . . -
. F . . . .
5010 2238 83ce 0000 0000 0000 0000 ----  |  P . " 8 . . . . . .
. . . . . .
22:17:48.709862 my.net.work.169.ftp > 213.44.212.89.3028: P
102:153(51) ack 35 win 8760 (DF) (ttl 255, id 11236)
4500 005b 2be4 4000 ff06 9ec5 d0c3 36a9  |  E . . [ + . @ . . .
. . . . 6 .
d52c d459 0015 0bd4 082d a446 037f 9cff  |  . , . Y . . . . . -
. F . . . .
5018 2238 fb32 0000 3530 3120 4e6f 7420  |  P . " 8 . 2 . . 5 0
```

```
1   N o t
7065 726d 6974 7465 6420 746f 2063 6f6e  |  p e r m i t t e d
t o   c o n
6e65 6374 2074 6f20 6674 702e 6d69 6372  |  n e c t   t o   f t
p . m i c r
6f73 6f66 742e 636f 6d0d 0a-- ---- ----  |  o s o f t . c o m
\r\n. . . . .
22:17:48.717399 my.net.work.169.ftp > 213.44.212.89.3028: F
153:153(0) ack 35 win 8760 (DF) (ttl 255, id 11237)
4500 0028 2be5 4000 ff06 9ef7 d0c3 36a9  |  E . . ( + . @ . . .
. . . . 6 .
d52c d459 0015 0bd4 082d a479 037f 9cff  |  . , . Y . . . . . -
. y . . . .
5011 2238 839a 0000 0000 0000 0000 ----  |  P . " 8 . . . . . .
. . . . . .
22:17:48.875789 213.44.212.89.3028 > my.net.work.169.ftp: . ack
154 win 32616 (ttl 112, id 9321)
4500 0028 2469 0000 7006 7574 d52c d459  |  E . . ( $ i . . p .
u t . , . Y
d0c3 36a9 0bd4 0015 037f 9cff 082d a47a  |  . . 6 . . . . . . .
. . . - . z
5010 7f68 266a 0000 0204 05b4 0101 ----  |  P . . h & j . . . .
. . . . . .
22:17:53.893319 213.44.212.89.3028 > my.net.work.169.ftp: R
58694911:58694911(0) win 0 (ttl 112, id 32617)
4500 0028 7f69 0000 7006 1a74 d52c d459  |  E . . ( . i . . p .
. t . , . Y
d0c3 36a9 0bd4 0015 037f 9cff 082c 02e3  |  . . 6 . . . . . . .
. . . , . .
5004 0000 4777 0000 0204 05b4 0402 ----  |  P . . . G w . . . .
. . . . . .
```

Probability the Address was spoofed:

Address is definitely not spoofed. FTP will only happen if the FTP client receives the response of the ftp server. A RIPE IP address query reveals that the IP Address belongs to another French Internet service provider.

Description of Attack:

Logs seem to indicate that attacker was attempting to copy something over or pull something from ftp.microsoft.com. The attacker seems to be looking for a repository for binaries, trojans, or what have you's.

Attacker could have cause an FTP denial of service by tying up a limited number of ftp child processes configured to run in the system. However, the attacker does not seem to indicate any interest in creating a DOS, he just wants a ftp relay server.

Attack Mechanism:

The attacker wants to use the ftp server to possibly cover his tracks. His intention may also have been to use the ftp server as a repository for tools, and stolen information.

Correlation:

Anonymous FTP attempts are very common. This was eloquently discussed below.

"The most common attack you will see are hackers/crackers looking for "open anonymous" FTP servers. These are servers with directories that can be written to and read from. Hackers/crackers use these machines as way-points for transferring warez (pirated programs) and pr0n (intentionally misspelled word to avoid search engines classifying this document)."[2]

Severity:

Severity = (Criticality + Lethality) - (System Countermeasures + Network Countermeasures)

= (2 + 2) - (4 + 3)

= -5

Criticality = 2 because it was more of a sweep than an actual attack to a live server

Lethality = 2 because it was more of an attempt to use a service rather than to compromise ther server.

System Countermeasures = 4 because unnecessary services have been disabled. FTP is not running on the servers.

Network Countermeasures = 3 because hosts are protected by a firewall but it seems that the firewall is somehow misconfigured because it is accepting ftp sessions initiated from the external interfaces (internet)

Defensive Recommendation

Evaluate firewall rule allowing outsiders (people coming in from external sources) to initiate connections to the Firewall. This could have been an oversight on the part of the firewall admins.

Multiple Choice Question:

```
Jun 26 22:17:04 my.network.com ftp-gw[27082]: deny
host=nodnsquery/213.44.212.89 connect to 0.0.0.0
Jun 26 22:17:05 my.network.com ftp-gw[27110]: deny
host=nodnsquery/213.44.212.89 connect to 0.0.0.0
Jun 27 00:17:04 my.network.com ftp-gw[27082]: exit
host=nodnsquery/213.44.212.89 cmds=0 in=0 out=0 user=unauth
duration=7200
Jun 27 00:17:05 my.network.com ftp-gw[27110]: exit
host=nodnsquery/213.44.212.89 cmds=0 in=0 out=0 user=unauth
duration=7200
```

In the gauntlet alert above, what triggres the connect to 0.0.0.0

[2]http://www.robertgraham.com/pubs/firewall-seen.html

deny message?

a. attacker hits the broadcast address
b. attacker hits a network address
c. attacker connects to 0.0.0.0
d. attacker is unauthorized

ANS: B -When the attacker accidentally connects to the network address, or start of the subnet, gauntlet sends out an alert message saying the connection to 0.0.0.0 has been denied. Then it times out after two hours. This behavior of the FTP-GW has probably been patched, or recent versions probably do not exhibit this behavior.

Trace 5 Squid Scan
Source of Trace: MY NETWORK

Detect was generated by:

Gauntlet

```
Jun 27 05:59:14 my.network.com unix: securityalert: no match
found in local screen: TCP if=qe3 srcaddr=130.92.65.142
srcport=3076 dstaddr=my.net.work.97 dstport=3128
Jun 27 05:59:14 my.network.com unix: securityalert: no match
found in local screen: TCP if=qe3 srcaddr=130.92.65.142
srcport=3142 dstaddr=my.net.work.163 dstport=3128
Jun 27 05:59:17 my.network.com unix: securityalert: no match
found in local screen: TCP if=qe3 srcaddr=130.92.65.142
srcport=3076 dstaddr=my.net.work.97 dstport=3128
Jun 27 05:59:17 my.network.com unix: securityalert: no match
found in local screen: TCP if=qe3 srcaddr=130.92.65.142
srcport=3142 dstaddr=my.net.work.163 dstport=3128
Jun 27 05:59:14 my.network.com unix: securityalert: no match
found in local screen: TCP if=qe3 srcaddr=130.92.65.142
srcport=3076 dstaddr=my.net.work.97 dstport=3128
Jun 27 05:59:14 my.network.com unix: securityalert: no match
found in local screen: TCP if=qe3 srcaddr=130.92.65.142
srcport=3142 dstaddr=my.net.work.163 dstport=3128
Jun 27 05:59:17 my.network.com unix: securityalert: no match
found in local screen: TCP if=qe3 srcaddr=130.92.65.142
srcport=3076 dstaddr=my.net.work.97 dstport=3128
Jun 27 05:59:17 my.network.com unix: securityalert: no match
found in local screen: TCP if=qe3 srcaddr=130.92.65.142
srcport=3142 dstaddr=my.net.work.163 dstport=3128
```

Snort

INFO - Possible Squid Scan	1 sources	65 destinations

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total)
130.92.65.142	406	406	65	65

RAW Snort Alerts

```
[**] INFO - Possible Squid Scan [**]
06/27-05:59:56.803322 130.92.65.142:3075 -> my.net.work.96:3128
TCP TTL:110 TOS:0x0 ID:28035 IpLen:20 DgmLen:64 DF
******S* Seq: 0x4C34887D  Ack: 0x0  Win: 0x2238  TcpLen: 44
TCP Options (9) => MSS: 1460 NOP WS: 0 NOP NOP TS: 0 0 NOP NOP
TCP Options => SackOK

[**] INFO - Possible Squid Scan [**]
06/27-05:59:56.808544 130.92.65.142:3076 -> my.net.work.97:3128
TCP TTL:110 TOS:0x0 ID:28036 IpLen:20 DgmLen:64 DF
******S* Seq: 0x4C357ECE  Ack: 0x0  Win: 0x2238  TcpLen: 44
TCP Options (9) => MSS: 1460 NOP WS: 0 NOP NOP TS: 0 0 NOP NOP
TCP Options => SackOK
[**] INFO - Possible Squid Scan [**]
06/27-05:59:56.812145 130.92.65.142:3077 -> my.net.work.98:3128
TCP TTL:110 TOS:0x0 ID:28037 IpLen:20 DgmLen:64 DF
******S* Seq: 0x4C362B89  Ack: 0x0  Win: 0x2238  TcpLen: 44
TCP Options (9) => MSS: 1460 NOP WS: 0 NOP NOP TS: 0 0 NOP NOP
TCP Options => SackOK
[**] INFO - Possible Squid Scan [**]
06/27-05:59:56.816328 130.92.65.142:3078 -> my.net.work.99:3128
TCP TTL:109 TOS:0x0 ID:28038 IpLen:20 DgmLen:64 DF
******S* Seq: 0x4C36B080  Ack: 0x0  Win: 0x2238  TcpLen: 44
TCP Options (9) => MSS: 1460 NOP WS: 0 NOP NOP TS: 0 0 NOP NOP
TCP Options => SackOK
[**] INFO - Possible Squid Scan [**]
06/27-05:59:56.824330 130.92.65.142:3080 -> my.net.work.101:3128
TCP TTL:110 TOS:0x0 ID:28040 IpLen:20 DgmLen:64 DF
******S* Seq: 0x4C37F26A  Ack: 0x0  Win: 0x2238  TcpLen: 44
TCP Options (9) => MSS: 1460 NOP WS: 0 NOP NOP TS: 0 0 NOP NOP
TCP Options => SackOK
[**] INFO - Possible Squid Scan [**]
06/27-05:59:56.830193 130.92.65.142:3081 -> my.net.work.102:3128
TCP TTL:110 TOS:0x0 ID:28041 IpLen:20 DgmLen:64 DF
******S* Seq: 0x4C388C9A  Ack: 0x0  Win: 0x2238  TcpLen: 44
TCP Options (9) => MSS: 1460 NOP WS: 0 NOP NOP TS: 0 0 NOP NOP
TCP Options => SackOK
Cut for Brevity ....
[**] INFO - Possible Squid Scan [**]
```

```
06/27-06:00:00.915344 130.92.65.142:3169 -> my.net.work.190:3128
TCP TTL:110 TOS:0x0 ID:29807 IpLen:20 DgmLen:64 DF
******S* Seq: 0x4C797633  Ack: 0x0  Win: 0x2238  TcpLen: 44
TCP Options (9) => MSS: 1460 NOP WS: 0 NOP NOP TS: 0 0 NOP NOP
TCP Options => SackOK
[**] INFO - Possible Squid Scan [**]
06/27-06:00:00.915883 130.92.65.142:3170 -> my.net.work.191:3128
TCP TTL:109 TOS:0x0 ID:29814 IpLen:20 DgmLen:64 DF
******S* Seq: 0x4C7A5FF5  Ack: 0x0  Win: 0x2238  TcpLen: 44
TCP Options (9) => MSS: 1460 NOP WS: 0 NOP NOP TS: 0 0 NOP NOP
TCP Options => SackOK
```

TCPDUMP

```
5:59:56.803322 130.92.65.142.3075 > my.net.work.96.3128: S
1278511229:1278511229 (0) win 8760 <mss 1460,nop,wscale
0,nop,nop,timestamp 0 0,nop,nop,sackOK> (DF) (ttl 110, id 28035)
05:59:56.804154 my.net.work.96.3128 > 130.92.65.142.3075: R
0:0(0) ack 1278511230 win 0 (DF) (ttl 110, id 19403)
05:59:56.804219 my.net.work.96.3128 > 130.92.65.142.3075: R
0:0(0) ack 1 win 0 (DF) (ttl 2, id 19404)
05:59:56.808544 130.92.65.142.3076 > my.net.work.97.3128: S
1278574286:1278574286(0) win 8760 <mss 1460,nop,wscale
0,nop,nop,timestamp 0 0,nop,nop,sackOK> (DF) (ttl 110, id 28036)
05:59:56.812145 130.92.65.142.3077 > my.net.work.98.3128: S
1278618505:1278618505(0) win 8760 <mss 1460,nop,wscale
0,nop,nop,timestamp 0 0,nop,nop,sackOK> (DF) (ttl 110, id 28037)
05:59:56.812770 my.net.work.98.3128 > 130.92.65.142.3077: R
0:0(0) ack 1278618506 win 0 (DF) (ttl 110, id 19405)
05:59:56.816328 130.92.65.142.3078 > my.net.work.99.3128: S
1278652544:1278652544(0) win 8760 <mss 1460,nop,wscale
0,nop,nop,timestamp 0 0,nop,nop,sackOK> (DF) (ttl 109, id 28038)
05:59:56.816706 my.net.work.99.3128 > 130.92.65.142.3078: R
0:0(0) ack 1278652545 win 0 (DF) (ttl 109, id 19406)
Cut for brevity ........
06:00:00.520085 130.92.65.142.3103 > my.net.work.124.3128: S
1279799132:1279799132(0) win 8760 <mss 1460,nop,wscale
0,nop,nop,timestamp 0 0,nop,nop,sackOK> (DF)(ttl 110, id 29555)
06:00:00.520524 my.net.work.124.3128 > 130.92.65.142.3103: R
0:0(0) ack 1 win 0 (DF) (ttl 110, id 19809)
06:00:00.613756 130.92.65.142.3139 > my.net.work.160.3128: S
1281530276:128153027
6(0) win 8760 <mss 1460,nop,wscale 0,nop,nop,timestamp 0
0,nop,nop,sackOK> (DF)(ttl 109, id 29602)
06:00:00.614100 my.net.work.160.3128 > 130.92.65.142.3139: R
0:0(0) ack 1 win 0 (DF)(ttl 109, id 19810)
```

```
06:00:00.614303 my.net.work.160.3128 > 130.92.65.142.3139: R
0:0(0) ack 1 win 0 (DF) (ttl 2, id 19811)
06:00:00.615218 130.92.65.142.3140 > my.net.work.161.3128: S
1281586215:1281586215(0) win 8760 <mss 1460,nop,wscale
0,nop,nop,timestamp 0 0,nop,nop,sackOK> (DF)(ttl 110, id 29612)
06:00:00.615656 my.net.work.161.3128 > 130.92.65.142.3140: R
0:0(0) ack 1 win 0 (DF) (ttl 110, id 19812)
06:00:00.812119 130.92.65.142.3148 > my.net.workh.3128: S
1281993394:128199339
4(0) win 8760 <mss 1460,nop,wscale 0,nop,nop,timestamp 0
0,nop,nop,sackOK> (DF)(ttl 109, id 29720)
06:00:00.915883 130.92.65.142.3170 > my.net.work.191.3128: S
1283088373:1283088373(0) win 8760 <mss 1460,nop,wscale
0,nop,nop,timestamp 0 0,nop,nop,sackOK> (DF)(ttl 109, id 29814)
06:00:00.916221 my.net.work.191.3128 > 130.92.65.142.3170: R
0:0(0) ack 1 win 0 (DF) (ttl 109, id 19831)
06:00:00.916420 my.net.work.191.3128 > 130.92.65.142.3170: R
0:0(0) ack 1 win 0 (DF) (ttl 2, id 19832)b
```

Probability the Address was spoofed:

Address is not spoofed because attacker needs to see the response to his probes.

IP Address Lookups reveal the following information:

University of Berne (NET-UNIBE)
Institute of Informatics and Applied Mathematics
Laenggassstrasse 51
CH-3012 Berne
CH
Netname: UNIBE
Netblock: 130.92.0.0 - 130.92.255.255
Coordinator:
Buetikofer, Fritz (FB61-ARIN) btkfr@ID.UNIBE.CH
+41 31 65 3843

Domain System inverse mapping provided by:

ARWEN.UNIBE.CH 130.92.9.52
SWIBE9.UNIBE.CH 130.92.1.1
SCSNMS.SWITCH.CH 130.59.10.30

Record last updated on 17-Feb-1994.
Database last updated on 3-Jul-2001 23:15:43 EDT.

> server arwen.unibe.ch
Default Server: mailhub.unibe.ch
Address: 130.92.9.52

Aliases: arwen.unibe.ch

> 130.92.65.142
Server: mailhub.unibe.ch
Address: 130.92.9.52
Aliases: arwen.unibe.ch

Name: zarkov.unibe.ch
Address: 130.92.65.142

Description of Attack:

Port 3128 is very closely related to port 8080. Both of them are commonly used for http proxy services. 3128 is the default port for Squid. The attacker is looking for open proxies for relaying or anonymizing purposes. He/She may also be looking for squid servers so that he can exploit some vulnerabilities in the future.

Attack Mechanism:

This is more of a reconnaissance effort by the attacker. This was an attempt to identify servers that is running SQUID, not a generic proxy server (Netscape, Micosoft, etc.). Once the target servers are identified, attacker will probably come back for the actual exploit or use the system to attack other servers.

Correlation:

A Bugtraq Security Advisory released on 7-19-2001 also says that Squid, versions TSL 1.01, 1.1, and 1.2, when configured with the httpd_accel_with_proxy off, accepts any requests to the proxy. This allows attacker to use the proxy to scan remote systems, etc. A vulnerability that may also be related to squid is as follows: Bugtraq ID 2059 and CVE Identifier CVE-1999-0710 which is classified as a relay or information gathering attempt.

Severity:

Severity = (Criticality + Lethality) - (System Countermeasures + Network Countermeasures)

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

= -4

Criticality = 2 because it was more of a sweep than an actual attack to a live server

Lethality = 2 because it was more of an reconnaissance effort rather than an actual attack

System Countermeasures = 4 because unnecessary services have been disabled. Squid is not running on the servers.

Network Countermeasures = 4 because hosts are protected by a firewall and the attempt to connect to 3128 was denied

Defensive Recommendation

Review firewall configuration in an attempt to tighten up the rulebase.

Review Router ACLs and is possible, tighten up ACL config to allow only necessary services to go in and out of the network.

Multiple Choice Question:

[**] INFO - Possible Squid Scan [**]
06/27-05:59:56.824330 130.92.65.142:3080 -> my.net.work.101:3128
TCP TTL:110 TOS:0x0 ID:28040 IpLen:20 DgmLen:64 DF
******S* Seq: 0x4C37F26A Ack: 0x0 Win: 0x2238 TcpLen: 44
TCP Options (9) => MSS: 1460 NOP WS: 0 NOP NOP TS: 0 0 NOP NOP
TCP Options => SackOK

[**] INFO - Possible Squid Scan [**]
06/27-05:59:56.830193 130.92.65.142:3081 -> my.net.work.102:3128
TCP TTL:110 TOS:0x0 ID:28041 IpLen:20 DgmLen:64 DF
******S* Seq: 0x4C388C9A Ack: 0x0 Win: 0x2238 TcpLen: 44
TCP Options (9) => MSS: 1460 NOP WS: 0 NOP NOP TS: 0 0 NOP NOP
TCP Options => SackOK

2.0 IDS Approach or Stategy

Great! Good Job! Your team has finally convinced upper management to integrate Intrusion Detection in the Corporate Infrastructure. They have approved a budget of $500K to purchase new equipment just for this initial roll out. Your teams role is to build, deploy, monitor and manage the IDS infrastructure.

Then it hits you. Is this a victory or a punishment? Your team of two, you and another admin, is expected to do the following:

1. Deploy IDS in X subnets.

   Deployment is the easy part. Tying the sensors all up and trying to correlate between all the alerts is the tricky part. You realize that you need to do this in phases. You may decide to deploy it, and work out the backend details in Phases. Management requires you to build the infrastructure ASAP.

2. Monitor and Analyze logs generated by the NIDs and HIDs

   Once the Sensors are deployed, you start getting alerts. You now have to analyze those alerts, zero in on a particular alert of interest, and decide whether you want to escalate this to your Organizations CSIRT or not.

   Your team is also responsible for assisting any Investigation required by the CSIRT to determine possible exposures or compromise.

3. Integrate CSIRT Procedures in the Organizational Chart

   OK. Let's face it. Organizations are not created equal. Your team also has to negotiate the challenges of the political environment in your organization; there are certain rules of engagement that you have to follow.

4. Intrusions do not happen on Mondays to Fridays only, management expects 24x7 Coverage.

   This is not possible. Yes your team can program the systems to give you a page or to send you email but that cannot replace trained on-site staff looking at the logs and continuously analyzing data.

   The Company needs to hire and train people. But until then, it's you and your admin buddy.

5. On the Job Training

   Intrusion Detection is a moving target. Sending people to training greatly increases the chance of success but experience is what really counts. Experience can only gained through time and exposure. Therefore, somehow, the feeling of inadequacy is there.

The "What If?" question haunts you, and somehow, you look for gurus to point you to the right path. It is quite clear; your team is not yet as strong as management expects it to be. And considering that your infrastructure is very critical to your organizations success, you need to beef up you IDS defenses. Your organization is a target. Management hopes that your IDS infrastructure will beef up security.

So, who are the people you can turn to? How do you handle the situation? Who has the skill,

the infrastructure, and the willingness to do IDS monitoring for you? Outsourced Security Services come in to the picture. What do the bring into the plate? Outsourced Security Service Providers are not created equal. However, the common threads that most OSS Providers seem to offer are:

1. Security Operations Center staffed 24*7*365.
2. Experienced Staff with the correct skill set focused and dedicated only to Intrusion Detection.
3. Ability to correlate IDS data with other customers thereby establishing trends that enables them to quickly integrate signatures to Managed Sensors.
4. A reporting component that allows the customer to securely view IDS reports and analysis.
5. Incident Response and Forensic capability that integrates with various external Law Enforcement Agencies.
6. Customized Training and Assistance.

At this point, we need to differentiate between Management and Monitoring. Monitoring Services is basically asking the OSSP to look at the logs and to alert you when they see something critical. On top of this, Management Services requires them to make sure that the signatures in the Sensor are current. They may consider adding a service or two depending on what your Company includes in the Service Package. Management Services for Intrusion Detection Sensors range from $2,400-$9,000 per device, per month. Monitoring services is a little bit cheaper ranging from $1,500-$3,200. The management services figures roughly translate to $28,800 per year, per sensor, 24*7*365.

The latest Salary Survey posted in the SANs webpage estimates that an Intrusion Detection Analyst earns approximately 120K per year. If an organization will do 24*7*365, this accumulates to about 4-5 doing 8-12 hour shifts round the clock, 7 days a week. In salaries alone, the organization spends roughly 600K to provide 24*7*365 coverage for IDS alone. Not to mention the recruitment process that seems to always indicate that there is not enough people with the correct experience and background, considering that IDS is relatively a new field that is evolving at light speed.

So, spending $1500, per device, for 24*7*365 is I would say, very reasonable. Why can't we just let them do the IDS for the Organization? ANSWER: Because some organizations want to keep Proprietary Data private. By allowing OSSPs to configure, tweak, and manage your IDS systems, you are giving them access to your network traffic. This can reveal trade secrets, and other proprietary data that your company is trying to protect. By limiting OSSPs to Monitoring Services, you are allowing them to see what you want them to see, just firewall logs and/or IDS Alerts.

Outsourced Security Services as a Tool

Just like any hardware and software solutions, IDS monitoring can be used as a tool designed to meet specific goals or targets.

IDS Monitoring helps organizations meet 24*7*365 coverage from day 1 of the implementation cycle. There is no lull time for recruitment and hiring. Most IDS Monitoring vendors claim that their staff is experienced and trained specifically for IDS.

Since my team is new in this field, it is very likely that my team will make rookie mistakes. OSSPs give me a safety net. A room for errors that I hope, in the process of correcting them, will train and transfer knowledge and expertise to me and my team.

OSSPs are not part of the Organization. Therefore, they are immune to the political climate in the organization. They can be very decisive with regards to giving advice and recommendation without fearing any political repercussions.

Most OSSPs have a direct channel to the Law Enforcement. If ever an incident happens, your selected OSSP will have a direct liason with the international community and law enforcement. This gives your organization an added level of coverage if the need to prosecure arises. The organizations CSIRT then gets a chance establish rapport, trust, and confidence with these people before the actual need to escalate or prosecute comes. This option of course, depends on the Contract or the Service Level Agreement with the vendor. There may be some retainer fees involved if this option is included in the SLA.

In IDS Sensor Placement Is Key

A successful Intrusion Detection Implementation requires strategic sensor placement. This requires the installation of Sensors all over the infrastructure specially the ingress points of the network. External Networks are monitored for possible attacks coming from the Internet, and Extranet, while Internal Networks are monitored for possible attacks coming from within the organization.

Organizations normally employ a multi-tiered approach in enforcing security. The first level of firewall protects web Servers or application servers, while backend systems are protected by another level of firewalls. This clearly demarks what is considered external and internal by the Organization.

Monitoring Services is a customization of the service offering of Outsource Security Services Providers. As mentioned earlier, this came about due to the need of some organizations to keep their data private.

Sensors that will be monitored by the external entity must be identified right from the very start.

a. which part of the network will most likely be an entry point

b. which part of the network is exposed 7*24

c. which part of the network is considered public

d. which part of the network is the organization most vulnerable

These questions only point to the external part of the network, after the external router, and before the firewall, the segment facing the Internet. A common approach is to have the OSSP monitor the external sensors, and have the in-house staff monitor the internal ones. This is of course, a case-to-case basis and may vary from organization to organization.

Conclusion

For Organizations that consider network data and traffic proprietary, OSSPs can be used as a tool to help meet IDS requirements. Service Level Agreements can be customized to delimit and scope what is going to be monitored and sent to the Provider. Like any tool, the organization should know the strengths of their OSSP. In-House expertise complements the selected OSSP where access to proprietary data becomes an issue. They should work hand in hand in delivering a well developed, well managed, and well-implemented Intrusion Detection Strategy.

The selection of Outsource Security Service Providers can make or break the IDS initiatives of any organization. Very much the like the selection of Hardware and Software solutions, Outsourced Security Service Providers for IDS Monitoring need to be evaluated.

3.0 Analyze This

Five days of scans, OOS files, and alerts were analyzed. I would have preferred having the knowledge of the location of the sensor to make more accurate conclusions about the data set. The intriguing part about the data set is that I could not really identify whether the sensor was in the internal part of the network or whether it was external. If I assumed it was on the internal side of the network, why am I seeing a lot of external addresses? As if the network was not protected at all.

If it was on the external side of the network, why was I seeing a lot of internal addresses? As if there was no restriction. No concept of acceptable use policy or anything like that.

I guess this is probably because I am not at all familiar with the University environment. I am used to having networks delimited by firewalls and ACLs. All servers must conform to the standard build procedure, patched, and scanned for vulnerabilities before it goes online.

3.1 File Selection - April 6-10 2001

Scans	OOS	Alerts
scans.010406	oos_April.6.2001	alert.010406
scans.010407	oos_April.7.2001	alert.010407
scans.010408	oos_April.8.2001	Alert.010408
scans.010409	oos_April.9.2001	alert.010409
scans.010410	oos_April.10.2001	alert.010410

3.2 List of Detects

Earliest alert at **00:00:09**.156032 *on 04/06/2001*
Latest alert at **23:31:24**.699257 *on 04/10/2001*

Signature (click for sig info)	# Alerts	# Sources	# Destinations
SYN-FIN scan!	1	1	1
STATDX UDP attack	2	2	1
connect to 515 from inside	5	3	3
ICMP SRC and DST outside network	6	3	4
NMAP TCP ping!	10	4	5
Back Orifice	16	1	16
Port 55850 tcp - Possible myserver activity - ref. 010313-1	23	10	13
High port 65535 udp - possible Red Worm - traffic	29	20	15
Tiny Fragments - Possible Hostile Activity	31	2	5
Null scan!	52	16	16
High port 65535 tcp - possible Red Worm - traffic	60	14	16
SUNRPC highport access!	65	4	4
Queso fingerprint	74	19	25
TCP SRC and DST outside network	79	13	37
Watchlist 000222 NET-NCFC	94	10	7
SMB Name Wildcard	120	79	68
WinGate 1080 Attempt	134	38	87
External RPC call	409	9	341
UDP SRC and DST outside network	483	37	240
connect to 515 from outside	717	11	523
Attempted Sun RPC high port access	5177	1	1
Watchlist 000220 IL-ISDNNET-990517	5308	25	30
Possible trojan server activity	7296	892	4881
Russia Dynamo - SANS Flash 28-jul-00	23989	3	2

3.3 Top Talkers

IP Address	Count
MY.NET.204.18	21440
MY.NET.15.214	10114
MY.NET.217.242	7756
MY.NET.202.34	6832
210.220.73.117	6329
MY.NET.229.130	6209
64.229.232.100	3864
MY.NET.226.190	3814
MY.NET.227.222	3442
63.163.94.13	3349
MY.NET.217.230	3251
202.145.57.82	3231

3.4 External IPs with Registration Info

The following IP Addresses were selected because of the following reasons:

1.Sheer Volume of Alerts

2.Criticality of the Alert

A. Watchlist 000220 IL-ISDNNET-990517

What are these Israeli IP Addresses doing in the Universities Network? Why are they generating so much noise? The first thee IP addresses seem to have a communication channel on the the 37000 range.

Watchlist 000220 IL-ISDNNET-990517	25 sources	30 destinations

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total)
212.179.79.2	2204	2204	3	3
212.179.84.195	614	614	1	1
212.179.80.232	518	518	1	1
212.179.80.225	417	417	1	1
212.179.24.155	395	395	1	1

212.179.5.90	334	334	1	1
212.179.125.114	291	291	2	2
212.179.95.5	274	274	6	6
212.179.40.130	78	78	1	1
212.179.29.216	47	47	1	1
212.179.2.185	45	45	1	1
212.179.80.67	29	29	1	1
212.179.48.82	23	23	2	2
212.179.27.6	15	15	2	2
212.179.7.10	6	6	3	3
212.179.67.67	4	4	1	1
212.179.72.53	3	3	1	1
212.179.25.27	2	2	1	1
212.179.81.73	2	2	1	1
212.179.82.30	2	2	1	1
212.179.80.231	1	1	1	1
212.179.76.22	1	1	1	1
212.179.84.246	1	1	1	1
212.179.81.254	1	1	1	1
212.179.82.31	1	1	1	1

inetnum: 212.179.79.0 - 212.179.79.63
netname: CREOSCITEX
descr: CREOSCITEX-SIFRA
country: IL
admin-c: ZV140-RIPE
tech-c: NP469-RIPE
status: ASSIGNED PA
notify: hostmaster@isdn.net.il
mnt-by: RIPE-NCC-NONE-MNT
changed: hostmaster@isdn.net.il 20001109
source: RIPE

route: 212.179.0.0/17
descr: ISDN Net Ltd.

origin: AS8551
notify: hostmaster@isdn.net.il
mnt-by: AS8551-MNT
changed: hostmaster@isdn.net.il 19990610
source: RIPE

person: Zehavit Vigder
address: bezeq-international
address: 40 hashacham
address: petach tikva 49170 Israel
phone: +972 52 770145
fax-no: +972 9 8940763
e-mail: hostmaster@bezeqint.net
nic-hdl: ZV140-RIPE
changed: zehavitv@bezeqint.net 20000528
source: RIPE

person: Nati Pinko
address: Bezeq International
address: 40 Hashacham St.
address: Petach Tikvah Israel
phone: +972 3 9257761
e-mail: hostmaster@isdn.net.il
nic-hdl: NP469-RIPE
changed: registrar@ns.il 19990902
source: RIPE

B. Attempted Sun RPC high port access

Why is this guy doing sun rpc scans from home? Is this guy even aware that his box is doing it? Hostname is ilm26-3-204.ec.rr.com. I think we should consider talking to the service provider and getting to the bottom of it. Somehow, I think, organizations connecting to the Internet should do due diligence by securing the boxes so that it will not be used to generate attacks.

Attempted Sun RPC high port access 1 sources 1 destinations

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total)
66.26.3.204	5177	5177	1	1

ROADRUNNER-MIDSOUTH (NETBLK-ROADRUNNER-MIDSOUTH)
13241 Woodland Park Road
Herndon, VA 20171
US

Netname: ROADRUNNER-MIDSOUTH

Netblock: 66.26.0.0 - 66.26.255.255
Maintainer: RRMS

Coordinator:
ServiceCo LLC (ZS30-ARIN) abuse@rr.com
1-703-345-3416

Domain System inverse mapping provided by:

DNS1.RR.COM	24.30.200.3
DNS2.RR.COM	24.30.201.3
DNS3.RR.COM	24.30.199.7
DNS4.RR.COM	65.24.0.172

ADDRESSES WITHIN THIS BLOCK ARE NON-PORTABLE

Record last updated on 14-Jun-2001.
Database last updated on 30-Jun-2001 22:59:10 EDT.
Rogers@Home Cambr (NETBLK-ON-ROG-CMDG-3)
1 Mount Pleasant Road
Toronto, ON M4Y 2Y5
CA

Netname: ON-ROG-CMDG-3
Netblock: 24.112.202.0 - 24.112.203.255

Coordinator:
Network Security, Fraud (AD30-ARIN) abuse@rogers.home.net
(416) 935-4729

Record last updated on 05-Feb-1999.
Database last updated on 30-Jun-2001 22:59:10 EDT.

C. Possible trojan server activity

Just like the previous alert, MY.NET.15.214 is communicating with an IP Addresses that happens to be registered to rr.com. In this instance, it is very likely that MY.NET.15.214 is compromised and is currently configured to hunt down vulnerable systems belonging to the range rr.com, which is a high-speed Internet service provider. The next source, 24.112.202.176 is an IP Address belonging to a Canadian ISP. This attacker seems to be looking for machines that is already running Subseven.

Possible trojan server activity	892 sources	4881 destinations

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total)
MY.NET,15.214	5176	5176	3880	3880
24.112.202.176	913	913	858	858
MY.NET.98.193	59	59	48	48
206.132.75.244	13	13	1	1
24.180.160.210	6	6	3	3
MY.NET.219.86	6	6	3	3
211.219.138.228	6	6	4	4
211.135.37.98	6	6	6	6
130.205.77.148	5	5	3	3
168.131.99.155	4	4	4	4
MY.NET.97.71	4	4	2	2
MY.NET.181.137	4	4	1	1
24.132.56.141	3	3	1	1
24.132.58.198	3	3	1	1
4.41.188.147	3	3	1	1

Port Number	Protocol	Port Name	General Description
27374	tcp	ODD Packet -Subseven	ODD Packet - SubSeven
27374	tcp	trojan / subseven	ImagePump

Global Crossing (NET-GBLX-9)
960 Hamlin Court
Sunnyvale, CA 94089
US

Netname: GBLX-9
Netblock: 206.132.0.0 - 206.132.255.255
Maintainer: GBLX

Coordinator:
Global Crossing (IA12-ORG-ARIN) ipadmin@gblx.net
+1 800 404-7714

Domain System inverse mapping provided by:

NAME.ROC.GBLX.NET 209.130.187.10
NAME.PHX.GBLX.NET 206.165.6.10

Record last updated on 06-Apr-2001.
Database last updated on 30-Jun-2001 22:59:10 EDT.
Netname: USWEST
Netblock: 130.13.0.0 - 130.13.255.255

Coordinator:
Qwest Communications (ZQ10-ARIN) abuse@tempe-vdoc.com
480-768-4338

Domain System inverse mapping provided by:

NS1.USWEST.NET 204.147.80.5
NS2.DNVR.USWEST.NET 206.196.128.1

Record last updated on 28-Mar-2001.
Database last updated on 30-Jun-2001 22:59:10 EDT.

@Home Network (NETBLK-BLTMMD1-MD-2)
425 Broadway
Redwood City, CA 94063
US

Netname: BLTMMD1-MD-2
Netblock: 24.180.160.0 - 24.180.175.255

Coordinator:
Operations, Network (HOME-NOC-ARIN) noc-abuse@noc.home.net
(601) 556-5599

Record last updated on 12-Jul-2000.
Database last updated on 30-Jun-2001 22:59:10 EDT.

D. Russia Dynamo - SANS Flash 28-jul-00

This detect is quite old. MY.NET.178.42 is still sending data to the black listed addresses. It could have been overlooked, or it could be a new infection. Regardless, this item really needs to be looked at. This is the top alert from the April 6-10 time period.

Russia Dynamo - SANS Flash 28-jul-00	3 sources	2 destinations

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total)
MY.NET.178.42	20790	20790	1	1
194.87.6.106	3071	3071	1	1
194.87.6.201	128	128	1	1

inetnum: 194.87.0.0 - 194.87.255.255
netname: RU-DEMOS-940901
descr: Provider Local Registry
country: RU
admin-c: DNOC-ORG
tech-c: RR-ORG
status: ALLOCATED PA
remarks: changed from SU-DOMES to RU-DEMOS 970415
mnt-by: RIPE-NCC-HM-MNT
changed: auto-dbm@ripe.net 19950424
changed: hostmaster@ripe.net 19960514
changed: hostmaster@ripe.net 19970415
changed: hostmaster@ripe.net 19981102
changed: hostmaster@ripe.net 19981209
changed: hostmaster@ripe.net 20000526
source: RIPE

route: 194.87.0.0/19
descr: DEMOS
origin: AS2578
notify: noc@demos.net
mnt-by: AS2578-MNT
changed: noc@demos.net 20000927
source: RIPE

role: Demos Internet NOC
address: Demos Company Ltd.
address: 6-1 Ovchinnikovskaya nab.
address: Moscow 113035
address: Russia
phone: +7 095 737 0436
phone: +7 095 737 0400
fax-no: +7 095 956 5042
e-mail: ncc@demos.net
admin-c: KEV6-RIPE
admin-c: RVP18-RIPE
admin-c: GK41-RIPE

tech-c: KEV6-RIPE
tech-c: RVP18-RIPE
tech-c: GK41-RIPE
nic-hdl: DNOC-ORG
notify: hm-dbm-msgs@ripe.net
notify: ncc@demos.net
notify: ip-reg@ripn.net
mnt-by: AS2578-MNT
changed: noc@demos.net 20010413
changed: evgeny@demos.su 20010607
source: RIPE

role: ROSNIIROS Registry
address: Russian Institute for Public Networks
address: 1, Kurchatov sq
address: Moscow
address: Russia
remarks: ********
We're not an ISP. We only provide registration services
for russian ISPs and not responsible for ISP's customer's spam
and illegal activity.
However we're ready to help you to identify
abused network contacts in case of any lookup problems.
remarks: ********
phone: +7 095 737 0604
fax-no: +7 095 946 9841
e-mail: ip-reg@ripn.net
e-mail: ip-dbm-request@ripn.net
admin-c: LY10-RIPE
tech-c: OB36-RIPE
tech-c: MNK1-RIPE
tech-c: EVK10-RIPE
nic-hdl: RR-ORG
notify: ip-reg@ripn.net
mnt-by: ROSNIIROS-MNT
changed: bon@ripn.net 19980930
changed: bon@ripn.net 19990622
source: RIPE

E. Connect to 515 from outside

An LPD or spooler vulnerability was released a couple of months back. I would be very surprised if the University allows outsiders from China to send print jobs to the Universities print servers. These are clearly attempts to gain access to the Universities computers.

connect to 515 from outside	11 sources	523 destinations

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total)
140.122.140.57	165	165	165	165
63.195.112.230	145	145	141	141
207.8.203.106	106	106	106	106
216.130.139.13	87	87	83	83
207.102.158.10	67	67	57	57
64.18.0.162	55	55	55	55
64.14.243.59	44	44	43	43
199.179.16.236	19	19	19	19
61.142.74.8	16	16	16	16
24.170.117.247	11	11	11	11
65.1.190.220	2	2	2	2

Ministry of Education Computer Center (NET-TANET-B88)
12th Fl, 106, Hoping E. Road, Sec 2.
Taiwan Republic of China, R.O.C
TW

Netname: TANET-BNET14
Netblock: 140.122.0.0 - 140.122.255.255

Coordinator:
TANet, Administrator (AT122-ARIN) tanetadm@moe.edu.tw
886-2-27377010-295

Domain System inverse mapping provided by:

CC.NTNU.EDU.TW 140.122.65.9
MOEVAX.EDU.TW 140.111.1.2

Record last updated on 14-Apr-1999.
Database last updated on 30-Jun-2001 22:59:10 EDT.
SNFC21 RBACK13 BASIC 63.195.112.0 (NETBLK-SBCIS39528)
303 2nd Street Suite 850N
San Francisco, CA 94107
US

Netname: SBCIS39528
Netblock: 63.195.112.0 - 63.195.119.255

Coordinator:
Pacific Bell Internet (PIA2-ORG-ARIN) ip-admin@PBI.NET
888-212-5411

Record last updated on 08-Sep-1999.
Database last updated on 30-Jun-2001 22:59:10 EDT.
Net Access (NETBLK-NETAXS-BLK)
PO Box 502
Glenside, PA 19038
US

Netname: NETAXS-BLK
Netblock: 207.8.128.0 - 207.8.255.255
Maintainer: NTAC

Coordinator:
Freedman, Avi (AF39-ARIN) freedman@NETAXS.COM
215 576 8669

Domain System inverse mapping provided by:

NS1.NETAXS.COM 207.8.186.1
NS2.NETAXS.COM 207.8.186.2

Record last updated on 04-Jun-1996.
Database last updated on 30-Jun-2001 22:59:10 EDT.
Newnan Utilities (NETBLK-WEST-GA-NET1)
70 Sewell Road
Newnan, GA 30264
US

Netname: WEST-GA-NET1
Netblock: 216.130.128.0 - 216.130.159.255
Maintainer: NEWN

Coordinator:
Morrow, Larry (LM435-ARIN) larry@a-plus.net
1 770 683 8324 (FAX) 1 770 252 4230

Domain System inverse mapping provided by:

DNS.NEWNANUTILITIES.ORG 216.130.152.71
DNS2.NEWNANUTILITIES.ORG 216.130.152.72

DNS.A-PLUS.NET 216.130.132.5

ADDRESSES WITHIN THIS BLOCK ARE NON-PORTABLE

Record last updated on 12-Mar-2001.
Database last updated on 30-Jun-2001 22:59:10 EDT.
US West Advanced Technologies (NET-USWEST)
4001 Discovery Drive
Boulder, CO 80303
US

3.5 Link Graph of OOS

MY.NET.227.130 is looking for external addresses running the service GNUtella (port 6346/tcp, 6346/udp). This was a very quite scan that only generated 169 logs when compared to the very loud scans generated by the subseven slaves. An article released last Feb. 27, 2001 talks about a virus, Mandragore[3], that seems to be spreading across Peer-to-Peer file exchange implementations like Gnutella. Another port that seems to be scanned frequently in the OOS files is 6699/tcp, Napster. Both these peer-to-peer file sharing implementations have documented vulnerabilities.

Other ports that were being probed were the usual 110, 80, and 22. Crafted flag combinations were being sent to the mentioned ports to possibly bypass network security measures implemented in the University.

3.6 Anomalous Activity / Possibility of Compromised Systems

Alerts suggest that MY.NET.204.18 and MY.NET.15.214 is a compromised system running subseven. They are both being controlled by the same master, 24.2.52.25. Scan logs also indicate that the servers have different directives. The 15.214 IP Address is looking for other trojans residing on desktops of possibly home users, 15.214 appears to be concentrating on the ISPs like @home, southwestern, and a bbn planet. The 204.18 however, seems to mapping the following networks: 63.88.120.21 - a UUNET block that has been assigned to Amrik Singh, and 129.21.112.10, an IP Address that belongs to Rochester Institute of Technology, NY. These scans could be decoy scans; they are not at all subtle.

Why is MY.NET.217.242 so active? The alert logs and the scan logs show that this server has been communication with the same IP Addresses from 9:41-10:45 on April 9. On this date, IP Address 66.26.3.204, started connection to the rpc port at 7:55 PM, this went on for the next 2 hours, and by 9:53, the internal host was scanning external hosts for rpc vulnerabilities. This is the same vulnerability that was exploited to compromise the system.

3.6 Defensive Recommendation

Fixing this will be a real challenge. There is evidence of compromise, and I do not see any evidence of any security mechanism in place other than the IDS sensors that gathered all the

[3]Http://www.kaspersky.com/news.asp?tnews=0&nview=1&id=162&page=0

data. I think the security posture of the University needs to be evaluated. I also think that there should be some acceptable use policy that limits the students as to what they can, and they cannot do. This way, anything out of the norm will be considered an Intrusion and will therefore be investigated.

The University might want to consider making an example out of one hacker by prosecuting him/her. At this point, I would assume that the University has enough data and forensic evidence to go after the uninvited visitors. If not, I think the University should seriously consider going after some of them. Hopefully, the successful prosecution might scare the other hackers away.

I would recommend containment. The University needs to localize the compromise. Easier said than done right? I do not have the slightest idea as to how I should go about it. A lot of systems show signs of compromise and I do not know where to start. I see a particular attacker having multiple slaves, so if you shut one down, he kills you with the other. The containment or localization issue needs to be addressed ASAP because the longer we wait, the more boxes gets compromised. The University needs to start somewhere until the last compromised system is rebuilt, and made more resistant to break-ins.

Second thing I would do is to prevent or at least make compromises more difficult. This can be done by installing firewalls, router filters, and active IDSs.

I would also like to make the students responsible for the security of their desktops or servers. Hardening documents should be readily available wherever a computer is present. I guess this is more of creating security awareness or consciousness in the University. Something as simple as making the students use stronger passwords, using screen savers, etc. It can also be as technical as having the students harden their servers themselves. I think a lot of mileage can be gained by making the students accountable to the security of their desktop. After all, it is their work that gets lost if a compromise takes place.

3.8 Analysis Process

Data Preparation

I used GAWK, SED, GREP, SORT, and EGREP to format the data in order to extract the top talker and top listener. I had to replace ":" with a white space in order to separate the port number with the source ip address and the destination ip address. I used vi to do this since I only had to process five files and I also did not have enough time to look up the man pages for SED.

The command below counts the number of instances of the source IP address by assigning it to a numeric array named source using the IP Address as the element name or subscript.

```
gawk -F " " '{ source[$6]++; } END { for (ip in source)  print ip, source[ip]}' apr6 apr7 apr8 apr9
apr10 > toptalker-6-10
```

The next command counts the number of instances of the source IP address by assigning it to a numeric array named dst.

```
gawk -F " " '{ dst[$9]++; } END { for (ip in dst)  print ip, dst[ip]}' apr6 apr7 apr8 apr9 apr10 >
toplistener-6-10
```

SED was also used to replace MY.NET with 10.0 so that snortsnarf can execute correctly.

Data Processing

As soon as I finished replacing MY.NET with 10.0, I was ready to run snortsnarf against all the alert files. Initially, I ran snortsnarf one file at a time. It was not long until I realized the need to correlate alerts. Since I only had to analyze five days of data, it was simpler for me to merge five days of alert to 1 big file and run snortsnarf against the merged file.

Data Analysis

The real challenge for me was to decide which event or particular alert I will pursue and really investigate. This was difficult for me because I cannot really decide which particular host is critical or not. I had to decide based on what I thought was severe based on the volume of alerts, and the lethality of the attack.

The Top Talker List was also compared against the consolidated alert list, and OOS in an effort to establish trends and links. I used utilities like grep and egrep to manually extract data that were relevant to the IP Addresses that I was trying to investigate.

REFERENCES

Stevens, Richard. TCP/IP Illustrated, Volume 1. Reading: Addison Wesley, 1994.

Northcut, Stephen etal. Intrusion Detection Signatures and Analysis. Indianapolis, IN: New Riders, January, 2001.

Northcut, Stephen and Judy Novak. Network Intrusion Detection, An Analysts Handbook 2nd Edition. Indianapolis, IN: New Riders, September 2000.

Scambray, Joel etal. Hacking Exposed: Network Security Secrets and Solutions 2nd Edition. Berkely, CA: Osborne/Mcgraw-Hill, 2001

Garfinkel, Simson and Gene Spafford. Practical UNIX and Internet Security 2nd Edition. Sebastopol, CA: O'Reilly & Associates, Inc., 1996

Varine, Brian. "Should we outsource monitoring?" April 3, 2001.http://www.sans.org/newlook/resources/IDFAQ/outsource.html (28 June 2001)

Graham, Robert. "Firewall Forensics (What am I seeing?)" Version 0.4.1. June 20, 2000. http://www.robertgraham.com/pubs/firewall-seen.html (21 June, 2001)

De Jesus, Edmund. "Managing Managed Security." Information Security Magazine January 2001:34-46

http://www.securityfocus.com
http://www.whitehats.com
http://www.sans.org
http://www.incidents.org
http://www.arin.net
http://www.ripe.net
http://www.apnic.net
http://www.robertgraham.com
http://www.sans.org/ids/faq
http://cve.mitre.org
http://www.kaspersky.com
http://www.riptech.com
http://www.silicondefense.com
http://www.counterpane.com
http://www.predictive.com
http://www.insecure.org

Last Updated: July 25th, 2017

Upcoming Training

SANS San Antonio 2017	San Antonio, TX	Aug 06, 2017 - Aug 11, 2017	Live Event
SANS Boston 2017	Boston, MA	Aug 07, 2017 - Aug 12, 2017	Live Event
SANS Virginia Beach 2017	Virginia Beach, VA	Aug 21, 2017 - Sep 01, 2017	Live Event
SANS Adelaide 2017	Adelaide, Australia	Aug 21, 2017 - Aug 26, 2017	Live Event
SANS Network Security 2017	Las Vegas, NV	Sep 10, 2017 - Sep 17, 2017	Live Event
SANS vLive - SEC503: Intrusion Detection In-Depth	SEC503 - 201709,	Sep 11, 2017 - Oct 18, 2017	vLive
SANS Baltimore Fall 2017	Baltimore, MD	Sep 25, 2017 - Sep 30, 2017	Live Event
Baltimore Fall 2017 - SEC503: Intrusion Detection In-Depth	Baltimore, MD	Sep 25, 2017 - Sep 30, 2017	vLive
SANS London September 2017	London, United Kingdom	Sep 25, 2017 - Sep 30, 2017	Live Event
Community SANS Scottsdale SEC503	Scottsdale, AZ	Oct 02, 2017 - Oct 07, 2017	Community SANS
SANS October Singapore 2017	Singapore, Singapore	Oct 09, 2017 - Oct 28, 2017	Live Event
Community SANS Ottawa SEC503	Ottawa, ON	Oct 16, 2017 - Oct 21, 2017	Community SANS
SANS Berlin 2017	Berlin, Germany	Oct 23, 2017 - Oct 28, 2017	Live Event
SANS Seattle 2017	Seattle, WA	Oct 30, 2017 - Nov 04, 2017	Live Event
San Diego Fall 2017 - SEC503: Intrusion Detection In-Depth	San Diego, CA	Oct 30, 2017 - Nov 04, 2017	vLive
SANS San Diego 2017	San Diego, CA	Oct 30, 2017 - Nov 04, 2017	Live Event
SANS Paris November 2017	Paris, France	Nov 13, 2017 - Nov 18, 2017	Live Event
Community SANS Pensacola SEC503	Pensacola, FL	Nov 27, 2017 - Dec 02, 2017	Community SANS
SIEM & Tactical Analytics Summit & Training	Scottsdale, AZ	Nov 28, 2017 - Dec 05, 2017	Live Event
SANS Cyber Defense Initiative 2017	Washington, DC	Dec 12, 2017 - Dec 19, 2017	Live Event
SANS Security East 2018	New Orleans, LA	Jan 08, 2018 - Jan 13, 2018	Live Event
SANS Las Vegas 2018	Las Vegas, NV	Jan 28, 2018 - Feb 02, 2018	Live Event
SANS Dallas 2018	Dallas, TX	Feb 19, 2018 - Feb 24, 2018	Live Event
SANS OnDemand	Online	Anytime	Self Paced
SANS SelfStudy	Books & MP3s Only	Anytime	Self Paced