



Global Information Assurance Certification Paper

Copyright SANS Institute
Author Retains Full Rights

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

Interested in learning more?

Check out the list of upcoming events offering
"Intrusion Detection In-Depth (Security 503)"
at <http://www.giac.org/registration/gcia>

GIAC Intrusion Analyst Certification Practical version 2.9

by

Wes Bateman

June 27, 2001

[Assignment 1 - Network Detects](#)

[Detect 1: Successful Compromise of rpc.statd on a Default RedHat 6.2 Installation](#)

[Detect 2: named Version Probe](#)

[Detect 3: Microsoft Tide Server Traffic](#)

[Detect 4: Massive Flood of Tiny Fragments](#)

[Detect 5: synscan Probe for DNS Services](#)

[Assignment 2 - State of Intrusion Detection](#)

[Assignment 3 - Analyze This](#)

[Appendix \(collateral information for detects and other assignments\)](#)

[List of References](#)

Assignment 1 - Network Detects

Detect 1: Successful compromise of rpc.statd on a default RedHat 6.2 installation

Mar 5 04:25:47 MV/IDS10/portmap-request-rstatd: 148.243.136.7:837 -> test.network.70.133:111
Mar 5 04:25:47 MV/IDS362/shellcode-x86-nops-udp: 148.243.136.7:838 ->
test.network.70.133:610
Mar 5 04:25:49 MV/IDS362/shellcode-x86-nops-udp: 148.243.136.7:838 ->
test.network.70.133:610
Mar 5 04:25:51 MV/IDS362/shellcode-x86-nops-udp: 148.243.136.7:838 ->
test.network.70.133:610
Mar 5 04:34:13 MISC - MISC - id check returned root: test.network.70.133:39168 ->
148.243.136.7:3909

Packet Dumps

1. Source of Trace:

This successful attack occurred on my company's network.

2. Detect was generated by:

Linux sensor running Snort 1.6.3 with a combination of whitehats, snort.org, and customized rulesets.

The rules that were triggered:

```
alert UDP $EXTERNAL any -> $INTERNAL 111 (msg: "MV/IDS10/portmap-request-rstatd";  
content: "|01 86 A0 00 00|");  
alert UDP $EXTERNAL any -> $INTERNAL any (msg: "MV/IDS362/shellcode-x86-nops-udp";  
content: "|90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90|");  
alert tcp any any -> any any (msg:"MISC - MISC - id check returned root"; content:  
"uid=0(root)");
```

3. Probability the source address was spoofed:

Very low. While these UDP packets could have been spoofed, it looks very unlikely that they were. First, the portmap request to port 111 would have required a response to inform the attacker that port 610 is where the vulnerable rpc.statd could be found. Further, another clue was seen in the last packet:

```
03/05-04:34:13.947820 0:3:E3:61:29:50 -> 0:30:71:2C:8:0 type:0x800 len:0xB0  
test.network.70.133:39168 -> 148.243.136.7:3909 TCP TTL:63 TOS:0x0 ID:3651 IpLen:20  
DgmLen:162 DF  
***AP*** Seq: 0xC81E7E72 Ack: 0xC172737E Win: 0x7D78 TcpLen: 32  
TCP Options (3) => NOP NOP TS: 50002261 177275836  
75 69 64 3D 30 28 72 6F 6F 74 29 20 67 69 64 3D uid=0(root) gid=  
30 28 72 6F 6F 74 29 0A 62 69 6E 0A 62 6F 6F 74 0(root).bin.boot  
0A 64 65 76 0A 65 74 63 0A 68 6F 6D 65 0A 6C 69 .dev.etc.home.li  
62 0A 6C 6F 73 74 2B 66 6F 75 6E 64 0A 6D 69 73 b.lost+found.mis  
63 0A 6D 6E 74 0A 6F 70 74 0A 70 72 6F 63 0A 72 c.mnt.opt.proc.r  
6F 6F 74 0A 73 62 69 6E 0A 74 66 74 70 62 6F 6F oot.sbin.tftpboo  
74 0A 74 6D 70 0A 75 73 72 0A 76 61 72 0A t.tmp.usr.var.
```

This packet is the only one captured by snort from a TCP stream that transpired. It is an ack/push, so it stands to reason that a full three-way handshake had successfully occurred before this packet. While an attacker could forge any TCP flags they chose, this packet originated on my network, and was a response to stimulus from the attacker. The RedHat Linux 6.2 installation's stack would have made sequence number prediction quite difficult, further reducing the odds that the source address could have been spoofed.

Basically, the source address was not spoofed. While it is entirely possible that the IP address of this attacker is simply another compromised host, being controlled by the real hostile party, the IP address shown as the source of the attack is almost certainly where this attack was launched from.

4. Description of attack:

This attack appears to be generated by `statdx.c` (<http://www.securityfocus.com/data/vulnerabilities/exploits/statdx.c>). According to whitehats (<http://www.whitehats.com/info/IDS442>) this attack is CVE-2000-0666 (<http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2000-0666>) and BugTraq #1480 (<http://www.securityfocus.com/bid/1480>).

I conducted additional forensic work on the compromised host at the time of the incident. This information can be found here:

[Detect 1 - Forensics Performed On Host](#)

5. Attack mechanism:

This attack succeeds by causing the remote `rpc.statd` daemon to execute code injected by the attacker. Since `rpc.statd` never drops its root privileges, the attacker's code is executed as root.

From the network intrusion analyst/network voyeur's point of view, this detect consisted of a UDP packet sent to the target host's portmap daemon. The attacker requested the target return information about which port the target's `rpc.statd` daemon listens on. The next three packets from the attacker, which snort alerted on and captured, were the actual attempts to exploit `rpc.statd`'s vulnerability. The packet dumps show many ix86 NOOPS (0x90) followed by something that looks alarmingly similar to `/bin/sh`. The appearance of something resembling `/bin/sh` should worry a UNIX admin, as something such as `cmd.exe` should worry a Windows admin. One might note that the attack seems to have been automated, as the time between the portmapper request and the first exploit packet was just under 1/10th of a second. Also possibly interesting is that the attacker's first probe packet, and his/her three attack packets all originated on ports lower than 1024. If the attacking host is running UNIX, then one might assume that the attacker has root privileges on that host, to be able to open low-numbered ports. Another observation is that the probing packet precedes the first attack packet by 1/10th of a second, while the three attack packets are each separated by two seconds. Then the last packet that snort captured comes in nine minutes later. I would infer that the first four packets were part of the automated attack process. The last packet appears to be the output from the "id" command. Since it came so much later, chronologically, it is possible that it came not as a result of the automated attack, but rather from the actual attacker, working on his/her newly owned host.

As for how the actual exploit works, a good technical description of the mechanics already exists on SecurityFocus.com (<http://www.securityfocus.com/bid/1480>):

A vulnerability exists in the `rpc.statd` program which is part of the `nfs-utils` packages, distributed with a number of popular Linux distributions. Because of a format string vulnerability when calling the `syslog()` function a malicious remote user can execute code as root.

The `rpc.statd` server is an RPC server that implements the Network Status and Monitor RPC protocol. It's a component of the Network File System (NFS) architecture.

The logging code in `rpc.statd` uses the `syslog()` function passing it as the format string

user supplied data. A malicious user can construct a format string that injects executable code into the process address space and overwrites a function's return address, thus forcing the program to execute the code.

rpc.statd requires root privileges for opening its network socket, but fails to drop these privileges later on. Thus code executed by the malicious user will execute with root privileges.

Debian, Red Hat and Connectiva have all released advisories on this matter. Presumably, any Linux distribution which runs the statd process is vulnerable, unless patched for the problem.

Further information can be gleaned from the statdx.c source code, written by ron1n :

```
*** background info
*** -----
*** rpc.statd is an ONC RPC server that implements the Network Status
*** Monitor RPC protocol to provide reboot notification. It is used by
*** the NFS file locking service (rpc.lockd) when it performs lock
*** recovery.
***
*** Due to a format string vulnerability in a call to syslog() within
*** its logging module, rpc.statd can be exploited remotely by script
*** kids bent on breaking into your Redhat Linux box and defacing your
*** website with crackpot political musings.
***
*** This is not a traditional buffer overflow vulnerability. The data
*** are kept within the bounds of the buffer by means of a call to
*** vsnprintf(). The saved return address can be overwritten indirectly
*** without a contiguous payload. syslog() is given, for the most part,
*** a user-supplied format string with no process-supplied arguments.
*** Our format string will, if carefully constructed, cause the process
*** to cull non-arbitrary addresses from the top of the stack for
*** sequential writes using controlled values. Exploitation requires
*** an executable stack on the target host -- almost invariably the
*** case. This problem was corrected in the nfs-utils-0.1.9.1 rpm.
```

6. Correlations:

This attacking IP address had never been observed connecting to any host across any of our networks before. This first contact seems to indicate that the attacker had some prior knowledge of our network. Since this target host's life on the internet was rather short, it is likely that this information was gleaned from a portmap scan which originated from another remote host (controlled by the same adversary) which would have occurred chronologically close to this event. Unfortunately, any such data is no longer available. Making this interesting and potentially valuable correlation opportunity impossible.

7. Evidence of active targeting:

This attack did seem to come out of nowhere. All traffic alerted on from this attacking host (148.243.136.7) can be viewed at the link below. While this host did conduct portmap scans of several of our networks AFTER this compromise, it was not observed doing so prior to this

break-in. It seems that this attacker had some previous knowledge that the portmapper was running on port 111 of this victim host.

So, my summary would be that this victim host was not the sole target of this attacker, as this same attacking host was later observed scanning large ranges of IP space looking for portmapper daemons. It is concerning however, that he/she did go right to this host and successfully compromise it, and that this was the first activity we ever saw from this attacking host.

[Detect 1 - Scans from this attacking host \(148.243.136.7\)](#)

8. Severity: 3

Criticality: 2

This system was a test system placed on the public network without prior approval from the Security Department. It was a stock Redhat 6.2 box with "everything" installed. It didn't survive long in the real world, but it was only a test system, and the engineer had all of his data replicated elsewhere. Further, this box was not dual-homed into any private network segment(s), nor was it "trusted" by any other host, so collateral damage was limited.

Lethality: 5

This system gave up root level access after its rpc.statd daemon was successfully compromised via the internet.

System Countermeasures: 1

Okay, the OS was reasonably up-to-date, and the root password was not "root," "password," or "toor." I'll give him that, otherwise this box was about as safe as a tourist who drinks the water in Mexico City. The system was unpatched and ran every dangerous, unneeded service that could be activated. I suppose the fact that it was compromised is another indicator that perhaps this host was not up to par. :)

Network Countermeasures: 3

I am, perhaps, being generous here. This host was placed on the network in such a way that the PIX was not filtering any traffic for it. It was completely unprotected. The only reason I give it a '3' here is because the network IDS caught this very quickly and the host was taken offline within about 20 minutes.

9. Defensive recommendation:

This incident helped to change the suggestion that systems be approved before placed onto the public network into a policy. Future systems so placed, had to be approved by the Security Department prior to going live on the network.

Generally speaking, I would recommend that the victim and all other systems to be placed on the publicly-accessible internet should be protected by a restrictively configured firewall. Short of this, this was a Linux system and it would have been trivial to add ipchains rules to this host so that it could do some filtering of its own. Further, such systems should be audited to insure that they only run the services needed to perform their function. This box had no need of portmapper or rpc services. When it is determined which services are required, then it should be ascertained what the latest versions of those programs are, and those should be installed.

10. Multiple choice test question:

When examining packet dumps from an suspected buffer overflow attempt against a UNIX host, the following would be cause for concern:

- A) A very large packet with apparently random or binary data sent to a host Solaris on a Sparc platform.
- B) A packet with a large number of 0x90's followed by something resembling /bin/sh sent to a host running Linux on an ix86 platform.
- C) A packet with a large number of 0x90's followed by something resembling /bin/sh sent to a host running Solaris on a Sparc platform.
- D) A large number of packets destined for the same port destined for a host running Linux on an ix86 platform.

Answer: B
 The 0x90 is NOOP for hosts with an ix86 architecture.

Detect 2: named Version Probe

```
May 29 16:27:53 MV/IDS278/named-probe-version: 63.110.182.244:1513 ->
dallas.network.182.103:53
May 29 16:27:53 MV/IDS278/named-probe-version: 63.110.182.244:1513 ->
dallas.network.182.108:53
May 29 16:27:56 MV/IDS278/named-probe-version: 63.110.182.244:1513 ->
dallas.network.182.104:53
May 29 16:27:56 MV/IDS278/named-probe-version: 63.110.182.244:1513 ->
dallas.network.182.107:53
```

Begin packet dump listing:

====

```
05/29-16:27:53.342505 0:30:71:2C:8:1 -> 0:6:29:DE:60:7D type:0x800 len:0x48
63.110.182.244:1513 -> dallas.network.182.103:53 UDP TTL:56 TOS:0x0 ID:57602 IpLen:20
DgmLen:58
Len: 38
13 44 01 00 00 01 00 00 00 00 00 07 56 45 52 .D.....VER
53 49 4F 4E 04 42 49 4E 44 00 00 10 00 03 SION.BIND.....
```

====

```
05/29-16:27:53.449651 0:30:71:2C:8:1 -> 0:6:29:DE:60:23 type:0x800 len:0x48
63.110.182.244:1513 -> dallas.network.182.108:53 UDP TTL:56 TOS:0x0 ID:57605 IpLen:20
DgmLen:58
Len: 38
B7 80 01 00 00 01 00 00 00 00 00 07 56 45 52 .....VER
53 49 4F 4E 04 42 49 4E 44 00 00 10 00 03 SION.BIND.....
```

=====
05/29-16:27:56.296898 0:30:71:2C:8:1 -> 0:6:29:DE:60:33 type:0x800 len:0x48
63.110.182.244:1513 -> dallas.network.182.104:53 UDP TTL:56 TOS:0x0 ID:58984 IpLen:20
DgmLen:58
Len: 38
C2 B3 01 00 00 01 00 00 00 00 00 00 07 56 45 52VER
53 49 4F 4E 04 42 49 4E 44 00 00 10 00 03 SION.BIND.....

=====
05/29-16:27:56.399662 0:30:71:2C:8:1 -> 0:6:29:DE:73:92 type:0x800 len:0x48
63.110.182.244:1513 -> dallas.network.182.107:53 UDP TTL:56 TOS:0x0 ID:59005 IpLen:20
DgmLen:58
Len: 38
71 2B 01 00 00 01 00 00 00 00 00 00 07 56 45 52 q+VER
53 49 4F 4E 04 42 49 4E 44 00 00 10 00 03 SION.BIND.....

=====
1. Source of Trace:
This scan occurred on my company's network.

2. Detect was generated by:
Linux sensor running Snort 1.7 with a combination of whitehats, snort.org, and customized rulesets.

The rule that triggered these alerts was:
alert UDP \$EXTERNAL any -> \$INTERNAL 53 (msg: "MV/IDS278/named-probe-version"; content: "|07|version|04|bind"; depth: 26; offset: 12; nocase;)

3. Probability the source address was spoofed:
While it would be trivial to spoof the source address of these UDP packets, I believe that it is unlikely that these source addresses were spoofed. I base that assumption entirely on the fact that the attacker will only benefit from this scan if he/she receives information back from the scan. There were no other hosts conducting any similar activity toward these target hosts observed either, making it even less likely that this source address was spoofed to act as a decoy of some sort.

4. Description of attack:
This is a scan of DNS servers to determine what version of the BIND daemon they are running. It is considered a pre-attack probe that allows an attacker to enumerate exactly what version of the daemon he/she might be contemplating attacking. This information will allow he/she to prepare the proper exploit code in a future attack attempt. According to whitehats (<http://www.whitehats.com/info/IDS442>) this attack is CVE-2000-0666 (<http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2000-0666>) and BugTraq #1480 (<http://www.securityfocus.com/bid/1480>).

5. Attack mechanism:
This scan sends a UDP packet to port 53 of the target system. The packet queries the target system, which is then expected to return version number information regarding its BIND daemon.

6. Correlations:
No previous traffic has ever been captured on any of our networks to or from these hosts. While I am

not able to correlate their apparent knowledge of our infrastructure to a previous event, it is certain that at some point this attacker successfully conducted some sort of reconnaissance.

7. Evidence of active targeting:

Strong. These are the first and only packets seen from these source IP addresses. They went directly to active nameservers on our network, indicating previous knowledge of these hosts and their function. These scans were very specifically targeted.

8. Severity: 0

Criticality: 4

These systems are active nameservers and as such is an important part of our network infrastructure.

Lethality: 3

While this is only a scan, it is a strong indicator that an outsider is attempting to obtain information about our nameservers that most likely are only useful to someone planning on attacking the named process on those hosts.

System Countermeasures: 4

Well patched version of OS. Well hardened with only minimal services running. I'd feel better if it was running djbdns (<http://cr.yo.com>) but the version of BIND it is running has no known vulnerabilities at this time.

Network Countermeasures: 3

Permissive firewall, but IDS system caught this traffic and it is both hoped and expected it will catch future hostilities launched against these nameservers, either from this scanner, or other hostile IP addresses.

9. Defensive recommendation:

Verify the version of BIND running on our nameservers regularly. Perhaps tighten firewall rules to contain damages in the event a compromise does take place. I would also like to see D.J. Bernstein's djbdns running in place of BIND on these servers. In lieu of that, however, the admin could/should also configure named to run as a user other than root. This user should not have a shell and the account should be disabled by placing a '*' in the password field of /etc/passwd for this user. This account then, should only be used for running the named service. Earlier versions of BIND only required root privilege to open TCP and UDP port 53. Newer versions will allow the named process to run as a regular user. There are many more compelling reasons not to run older versions of BIND as well. As an additional step, I would also block TCP traffic to port 53 on the nameserver(s) to prevent zone transfers to outside hosts.

10. Multiple choice test question:

Version queries are:

- A) Sometimes part of routine name resolution.
- B) Prevented by blocking TCP connections to port 53 on the nameserver.
- C) Conducted with specialized scanning tools, as they are not supported by standard tools, such as "dig."
- D) All of the above
- E) None of the above

Answer: E

Detect 3: Microsoft Tide Server Traffic

[Full List of Alerts](#)

Nov 29 17:55:56 2dw IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.76:32344 -> seattle.network.3.254:80
Nov 29 17:55:56 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.76:32350 -> seattle.network.3.254:80
Nov 29 17:55:56 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.76:32352 -> seattle.network.3.254:80
Nov 29 17:55:56 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.76:32361 -> seattle.network.3.254:80
Nov 29 17:55:56 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.86:8958 -> seattle.network.3.254:80
Nov 29 17:55:56 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.76:32365 -> seattle.network.3.254:80
Nov 29 17:55:56 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.76:32368 -> seattle.network.3.254:80
Nov 29 17:55:57 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.76:32369 -> seattle.network.3.254:80
Nov 29 17:55:57 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.76:32374 -> seattle.network.3.254:80

Jan 3 16:33:45 WBID Tide Server Activity: 131.107.3.86:11616 -> denver.network.46.254:80
Jan 3 16:33:45 WBID Tide Server Activity: denver.network.46.254:80 -> 131.107.3.86:11616
Jan 3 16:33:45 WBID Tide Server Activity: 131.107.3.86:11616 -> denver.network.46.254:80
Jan 3 16:33:45 WBID Tide Server Activity: denver.network.46.254:80 -> 131.107.3.86:11616
Jan 3 16:33:45 WBID Tide Server Activity: denver.network.46.254:80 -> 131.107.3.86:11616
Jan 3 16:33:45 WBID Tide Server Activity: 131.107.3.86:11616 -> denver.network.46.254:80
Jan 3 16:33:45 WBID Tide Server Activity: denver.network.46.254:80 -> 131.107.3.86:11616
Jan 3 16:33:45 WBID Tide Server Activity: denver.network.46.254:80 -> 131.107.3.86:11616
Jan 3 16:33:45 WBID Tide Server Activity: denver.network.46.254:80 -> 131.107.3.86:11616
Jan 3 16:33:45 WBID Tide Server Activity: 131.107.3.86:11616 -> denver.network.46.254:80
Jan 3 16:33:45 WBID Tide Server Activity: 131.107.3.79:21174 -> denver.network.46.254:80
Jan 3 16:33:45 WBID Tide Server Activity: denver.network.46.254:80 -> 131.107.3.79:21174
Jan 3 16:33:45 WBID Tide Server Activity: 131.107.3.79:21174 -> denver.network.46.254:80
Jan 3 16:33:45 WBID Tide Server Activity: 131.107.3.79:21174 -> denver.network.46.254:80
Jan 3 16:33:45 WBID Tide Server Activity: denver.network.46.254:80 -> 131.107.3.79:21174

01/03-13:29:17.315694 0:2:4B:19:8B:1 -> 2:E0:52:FB:3:FE type:0x800 len:0x131
131.107.3.74:17145 -> seattle.network.3.254:80 TCP TTL:54 TOS:0x0 ID:62004 DF
*****PA* Seq: 0x7F76ECB9 Ack: 0x66C6B7 Win: 0x4470

2D 32 31 0D 0A 43 6F 6E 74 65 6E 74 2D 4C 65 6E -21..Content-Len
67 74 68 3A 20 30 0D 0A 55 73 65 72 2D 41 67 65 gth: 0..User-Age
6E 74 3A 20 4D 69 63 72 6F 73 6F 66 74 20 44 61 nt: Microsoft Da
74 61 20 41 63 63 65 73 73 20 49 6E 74 65 72 6E ta Access Intern
65 74 20 50 75 62 6C 69 73 68 69 6E 67 20 50 72 et Publishing Pr
6F 76 69 64 65 72 20 50 72 6F 74 6F 63 6F 6C 20 ovider Protocol
44 69 73 63 6F 76 65 72 79 0D 0A 48 6F 73 74 3A Discovery..Host:
XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX www.XXXXXXXXXXX
XX XX 2E 63 6F 6D 0D 0A 54 72 61 6E 73 6C 61 74 XX.com..Translat
65 3A 20 66 0D 0A 50 72 61 67 6D 61 3A 20 6E 6F e: f..Pragma: no
2D 63 61 63 68 65 0D 0A 43 6F 6F 6B 69 65 3A 20 -cache..Cookie:
XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XXXXXXXXXXXXXXXXXXXX
XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XXXXXXXXXXXXXXXXXXXX
XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX 0D XXXXXXXXXXXXXXXXXXXX.
0A 43 6F 6E 6E 65 63 74 69 6F 6E 3A 20 4B 65 65 .Connection: Kee
70 2D 41 6C 69 76 65 0D 0A 0D 0A p-Alive....

1. Source of Trace:

My company's network.

2. Detect was generated by:

Linux sensor running Snort 1.6.3 with a combination of whitehats, snort.org, and customized rulesets. After the initial discovery that Microsoft's "tide" servers were initiating quite a bit of this type of traffic, a rule was added to alert on all traffic to/from the class C address space of the tide servers. This is where the "Tide Server Activity" alerts came from. The snort rule that was triggered initially that brought attention to this traffic was:

alert TCP !\$HOME_NET any -> \$HOME_NET 80 (msg:"IDS305 - WEB IIS - View Source via Translate Header"; flags: PA; content: "Translate|3A| F"; nocase;)

3. Probability the source address was spoofed:

Very low. This traffic is legitimate, although unusual. The packets captured are a part of a TCP stream, after a successful 3-way handshake had occurred.

4. Description of attack:

This traffic is better classified as a rather unfriendly series of web crawls originating from Microsoft's "Tide" proxy servers, and covering our and several of our customer's websites. Since this turned out to be semi-legitimate traffic, there are no CVE or BugTraq numbers to reference.

5. Attack mechanism:

The tide servers seemed to be performing a massive mirroring or similar operation. There were successful and normal three-way handshakes and then many many HTTP GETs. This traffic was legitimate in that if not for the volume would have appeared completely normal. It is still unknown to me exactly what this traffic was caused by and/or why it occurred. It was so interesting though, that I thought to include my analysis of it here.

6. Correlations:

After investigating the alerts being generated, I was surprised to see that the traffic was all originating from Microsoft servers. They resolved to tide##.microsoft.com, where ## was the last octet of their IP address. The number of alerts caused me to be concerned about how much other traffic was moving between our networks and microsoft.com, which was not tripping any snort rules. I therefore added a rule to capture and alert on all traffic to/from the 131.107.3 network. This caused massive alerting and confirmed my fear that this traffic was placing large demands on our available bandwidth. I had to remove the extra snort rule before too long, as the amount of traffic I was capturing was becoming

unmanageable.

I attempted some further correlation by searching for occurrences of the first three octets of the IP (the entire class C) on google (<http://www.google.com>). My findings were less than exciting. No other mention of people detecting unusual traffic from these tide servers. It did become apparent that some or all of the hosts in this class C were proxying traffic for people. There were many posts to all sorts of websites and discussion groups. This didn't seem to give me any additional information (besides the confirmation of the proxy role of these hosts) that would be of help in determining what was going on, exactly. It did make me wonder if these proxies were proxying requests from inside Microsoft...if so, they appear to have many employees possibly not making the best use of their time :)

7. Evidence of active targeting:

The tide proxy servers certainly were targeting our web servers, but I have never been able to determine exactly why.

8. Severity: 1

Criticality: 3

These systems are web proxies and are not individually important, but the number of systems involved makes this traffic of more concern.

Lethality: 2

While not exploit attempts, and not apparently purposeful attempts to create a denial of service condition, these Microsoft tide proxies were generating massive amounts of bandwidth utilization and quite likely reducing availability to our customers.

System Countermeasures: 4

These systems are recent OS deployments which are well-patched. Further they only run minimal required services, custom configured.

Network Countermeasures: 0

There is nothing in place currently to block bandwidth abuses such as this.

9. Defensive recommendation:

It was decided not to contact Microsoft, but rather wait to see if the traffic subsided, it eventually did, but as can be seen from the timestamps in the alerts this traffic was observed over quite a period of time. It would be interesting to contact Microsoft and inquire as to the purpose of this activity. Also, I would recommend a policy that allowed for Network Operations personnel to make a call to begin blocking traffic from abusive usage such as this. Finally, it might be worth investigating a method to prioritize bandwidth utilization so that massive traffic spikes like this could be metered and contained, still providing the tide servers with requested objects, but at a measured pace, maintaining availability for other customers. The distributed nature of our network probably minimized any service interruption that might have been caused to our customers.

Perhaps what I found interesting is how this event highlighted the tendency of technology companies to try to ignore this kind of incident. There appeared to be no immediate threat of compromise. The network seemed to handle the load. Content (both ours and that of our customers) was being requested and delivered by/to an outside entity. These were generally thought of as good things, and it really is a challenge to get management to think otherwise. Reacting to this type of incident by either restricting service or contacting an end-user is a big deal. Many companies whose business revolves around end-users being able to use the companies network resources, would rather wait out such an incident. These "grey areas" are hard to defend against and/or react to.

10. Multiple choice test question:

Large spikes in internet traffic often indicates:

- A) A misconfigured host or network device.
- B) A denial of service attack.
- C) Legitimate traffic.
- D) Any of the above

Answer: D

Detect 4: Massive Flood of Tiny Fragments

```
[wes@somehost ~]$ head tinyfragments
```

```
Jan 15 09:48:17 Tiny Fragments - Possible Hostile Activity: 209.141.75.25 -> austin.network.139.126
Jan 15 09:48:17 Tiny Fragments - Possible Hostile Activity: 209.141.75.25 -> austin.network.139.126
Jan 15 09:48:17 Tiny Fragments - Possible Hostile Activity: 209.141.75.25 -> austin.network.139.126
Jan 15 09:48:17 Tiny Fragments - Possible Hostile Activity: 209.141.75.25 -> austin.network.139.126
Jan 15 09:48:17 Tiny Fragments - Possible Hostile Activity: 209.141.75.25 -> austin.network.139.126
Jan 15 09:48:17 Tiny Fragments - Possible Hostile Activity: 209.141.75.25 -> austin.network.139.126
Jan 15 09:48:17 Tiny Fragments - Possible Hostile Activity: 209.141.75.25 -> austin.network.139.126
Jan 15 09:48:17 Tiny Fragments - Possible Hostile Activity: 209.141.75.25 -> austin.network.139.126
Jan 15 09:48:17 Tiny Fragments - Possible Hostile Activity: 209.141.75.25 -> austin.network.139.126
Jan 15 09:48:17 Tiny Fragments - Possible Hostile Activity: 209.141.75.25 -> austin.network.139.126
```

```
[wes@somehost ~]$ tail tinyfragments
```

```
Jan 15 14:46:43 Tiny Fragments - Possible Hostile Activity: 209.141.73.139 ->
sanfrancisco.network.247.190
Jan 15 14:46:43 Tiny Fragments - Possible Hostile Activity: 209.141.73.139 ->
sanfrancisco.network.247.190
Jan 15 14:46:43 Tiny Fragments - Possible Hostile Activity: 209.141.73.139 ->
sanfrancisco.network.247.190
Jan 15 14:46:43 Tiny Fragments - Possible Hostile Activity: 209.141.73.139 ->
sanfrancisco.network.247.190
Jan 15 14:46:43 Tiny Fragments - Possible Hostile Activity: 209.141.73.139 ->
sanfrancisco.network.247.190
Jan 15 14:46:43 Tiny Fragments - Possible Hostile Activity: 209.141.73.139 ->
sanfrancisco.network.247.190
Jan 15 14:46:43 Tiny Fragments - Possible Hostile Activity: 209.141.73.139 ->
sanfrancisco.network.247.190
Jan 15 14:46:43 Tiny Fragments - Possible Hostile Activity: 209.141.73.139 ->
sanfrancisco.network.247.190
Jan 15 14:46:43 Tiny Fragments - Possible Hostile Activity: 209.141.73.139 ->
sanfrancisco.network.247.190
Jan 15 14:46:43 Tiny Fragments - Possible Hostile Activity: 209.141.73.139 ->
sanfrancisco.network.247.190
```

```
[wes@somehost ~]$ du -hsc tinyfragments
206M tinyfragments
```

```
206M total
[wes@somehost ~]$ du -hsc tinyfragdumps
839M tinyfragdumps
839M total
```

```
03/03-00:54:02.934789 0:2:16:BB:E4:81 -> 2:E0:52:84:2:FE type:0x800 len:0x3C
202.9.169.27 -> 202.132.2.254 TCP TTL:112 TOS:0x10 ID:62722 IpLen:20 DgmLen:20 DF MF
Frag Offset: 0x0 Frag Size: 0x0
=====
03/03-00:54:03.902585 0:2:16:BB:E4:81 -> 2:E0:52:84:2:FE type:0x800 len:0x3C
202.9.169.27 -> 202.132.2.254 TCP TTL:112 TOS:0x10 ID:3587 IpLen:20 DgmLen:20 DF MF
Frag Offset: 0x0 Frag Size: 0x0
=====
03/03-00:54:04.093912 0:2:16:BB:E4:81 -> 2:E0:52:84:2:FE type:0x800 len:0x3C
202.9.169.27 -> 202.132.2.254 TCP TTL:112 TOS:0x10 ID:3843 IpLen:20 DgmLen:20 DF MF
Frag Offset: 0x0 Frag Size: 0x0
=====
03/09-07:16:22.433848 0:2:16:BB:DB:1 -> 2:E0:52:35:20:FE type:0x800 len:0x3C
202.9.185.18 -> mexicocity.network.32.254 TCP TTL:113 TOS:0x10 ID:48266 IpLen:20 DgmLen:20
DF MF
Frag Offset: 0x0 Frag Size: 0x0
=====
03/09-07:16:22.439846 0:2:16:BB:DB:1 -> 2:E0:52:35:20:FE type:0x800 len:0x3C
202.9.185.18 -> mexicocity.network.32.254 TCP TTL:113 TOS:0x10 ID:48522 IpLen:20 DgmLen:20
DF MF
Frag Offset: 0x0 Frag Size: 0x0
=====
03/09-07:16:22.502056 0:2:16:BB:DB:1 -> 2:E0:52:35:20:FE type:0x800 len:0x3C
202.9.185.18 -> mexicocity.network.32.254 TCP TTL:113 TOS:0x10 ID:50314 IpLen:20 DgmLen:20
DF MF
Frag Offset: 0x0 Frag Size: 0x0
=====
03/09-07:16:22.529241 0:2:16:BB:DB:1 -> 2:E0:52:35:20:FE type:0x800 len:0x3C
202.9.185.18 -> mexicocity.network.32.254 TCP TTL:113 TOS:0x10 ID:52618 IpLen:20 DgmLen:20
DF MF
Frag Offset: 0x0 Frag Size: 0x0
=====
01/14-09:11:51.038363 0:2:16:16:4D:C1 -> 2:E0:52:34:53:CC type:0x800 len:0x3C
209.140.168.9 -> sanjose.network3.83.204 TCP TTL:241 TOS:0x0 ID:6732 IpLen:20 DgmLen:20 DF
MF
Frag Offset: 0x0 Frag Size: 0x0
=====
01/14-09:11:51.058524 0:2:16:16:4D:C1 -> 2:E0:52:34:53:CC type:0x800 len:0x3C
209.140.168.9 -> sanjose.network3.83.204 TCP TTL:241 TOS:0x0 ID:6736 IpLen:20 DgmLen:20 DF
MF
Frag Offset: 0x0 Frag Size: 0x0
=====
01/14-09:11:51.078503 0:2:16:16:4D:C1 -> 2:E0:52:34:53:CC type:0x800 len:0x3C
209.140.168.9 -> sanjose.network3.83.204 TCP TTL:241 TOS:0x0 ID:6740 IpLen:20 DgmLen:20 DF
MF
Frag Offset: 0x0 Frag Size: 0x0
=====
```

```

01/14-09:11:51.089317 0:2:16:16:4D:C1 -> 2:E0:52:34:53:CC type:0x800 len:0x3C
209.140.168.9 -> sanjose.network3.83.204 TCP TTL:241 TOS:0x0 ID:6744 IpLen:20 DgmLen:20 DF
MF
Frag Offset: 0x0 Frag Size: 0x0
==+==+==+==+==+==+==+==+==+==+==+==+==+==+==+==+==+==+==+==+==+==+==+==+==+==+==+==+
01/14-09:11:51.108492 0:2:16:16:4D:C1 -> 2:E0:52:34:53:CC type:0x800 len:0x3C
209.140.168.9 -> sanjose.network3.83.204 TCP TTL:241 TOS:0x0 ID:6748 IpLen:20 DgmLen:20 DF
MF
Frag Offset: 0x0 Frag Size: 0x0
==+==+==+==+==+==+==+==+==+==+==+==+==+==+==+==+==+==+==+==+==+==+==+==+==+==+==+==+
01/14-09:11:51.138273 0:2:16:16:4D:C1 -> 2:E0:52:34:53:CC type:0x800 len:0x3C
209.140.168.9 -> sanjose.network3.83.204 TCP TTL:241 TOS:0x0 ID:6752 IpLen:20 DgmLen:20 DF
MF
Frag Offset: 0x0 Frag Size: 0x0
==+==+==+==+==+==+==+==+==+==+==+==+==+==+==+==+==+==+==+==+==+==+==+==+==+==+==+==+

```

1. Source of Trace:

My company's network.

2. Detect was generated by:

Linux sensor running Snort 1.6.3 with a combination of whitehats, snort.org, and customized rulesets.

3. Probability the source address was spoofed:

It would be completely possible for someone to spoof the source IP addresses in these packets. Although they're TCP packets, they don't appear to be part of a TCP stream that is following a successful 3-way handshake. In fact, no TCP flags are found at all, due to the extremely small size of each of these packets.

Some of the correlation work I did, watching other traffic from hosts in the netcarrier.net network seemed to lend validity to the thought that this traffic might have actually come from the source addresses seen in the IP headers. This is a tough call. My personal belief is that these packets truly originated from the netcarrier.net addresses. There are the previous and subsequent network connections seen from their network. Also the user209 seen in both of these IPs was odd to me. This same name was seen in portions of the names resolved to other IPs from netcarrier.net observed making connections on other dates.

Unfortunately though, like most of this detect, this is just guesswork. Much of this incident is still a mystery to me. An interesting puzzle to ponder, but I still don't know as much as I'd like to about what truly was happening here.

4. Description of attack:

A closer look at a dump of a suspect packet is in order, as I'd like to insure that the fields that snort is interpreting for me, are being accurately reported:

```

01/15-11:09:48.000638 209.141.73.139 -> 216.148.247.190
TCP TTL:239 TOS:0x0 ID:63278 IpLen:20 DgmLen:20 MF
Frag Offset: 0x0 Frag Size: 0x0
0x0000: 02 E0 52 94 F7 BE 00 02 16 BB CF C1 08 00 45 00 ..R.....E.
0x0010: 00 14 F7 2E 20 00 EF 06 C9 48 D1 8D 49 8B D8 94 ....H..I...
0x0020: F7 BE 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
0x0030: 00 00 00 00 00 00 00 00 00 00 00 00 .....

```

IP Header

Version: 4

Header Length: 5 (x4=20)

T.O.S.: 0

Total Length: 0014 (20 in decimal)

IP ID: F72E (63278 in decimal)

2 0 0 0

1 1 1 1 1 1 1 1 1 1 1 1

x

IP Flags: 0 0 1 (These control fragmentation. The first bit is not currently in use. The next bit is set to zero (may fragment). The last bit is set to one (more fragments coming.)

Fragment Offset: 0 (indicating that this packet, like all of the others, are the first packet in the fragmented datagram)

TTL: EF (239 in decimal)

Protocol: 06 (TCP)

Header Checksum: C9 48 (51528 in decimal)

Source IP: D1.8D.49.8B (209.141.73.139 in decimal)

Destination IP: D8.94.F7.BE (216.148.247.190 in decimal)

This packet was very typical of all of those in this detect. It seems that every packet is the first fragment in the group (frag offset of 0). Also, this fragment does have the MF flag set, indicating that the receiving host should keep its buffer open and await further fragments. Some of the packets also had the DF (do not fragment) flag set, along with the MF flag...this certainly appears to be an odd combination. :)

The IP ID numbers vary, the TTL's stay the same for each source host, for the most part. Some fluctuation was seen, but not by much and not frequently. Nothing here that appeared way out of band anyway. These fragments still puzzle me.

As far as I can guess, one of three things is occurring here:

- 1) There is some type of network configuration problem somewhere along the path, causing these fragments, and repeatedly spewing them at us.
- 2) This is an attempt to slow and/or stop network connectivity to our customer's origin webserver.
- 3) This is an attempt at diversion. Where the adversary is generating all of this traffic in an attempt to hide his/her true intention within the noise. Not only would it make alerts harder to find within the volume of alerts generated by the fragmentation traffic, but it might also cause any IDS tools to go blind (as partially occurred to our system, we didn't really go blind, but ALL of the alerts weren't evaluated as part of our normal IDS review process).

I don't like to admit it, but I'm not certain...to this day.

There are CVE numbers for fragmentation attacks, but since I haven't been able to specifically identify the purpose of this, it is difficult to know if any of the fragmentation vulnerabilities CVE numbers are applicable. CVE-1999-0052 specifically refers to FreeBSD, but seems somewhat similar (at least in that it references a DoS attack using fragmentation).

5. Attack mechanism:

If a denial of service attack was truly the motive, then these attacks are mistakenly being sent to our proxying servers, in an attempt to swamp them by forcing them to attempt to store all of the fragments they receive, awaiting further fragments and looking for one without the MF flag set. I assume that the

buffer would hold all packets, not overwriting each packet (since their fragment offset is always 0) until AFTER reassembly started (which it never will because the MF bit is always set). I would think though, that all fragments would be dropped after 60 seconds (UNIX destination hosts) and an ICMP time exceeded message would be sent to the source. Then more 0 offset fragments would come right in and the process would repeat. This could be effective, but only so long as the massive volume of fragmented packets is maintained. Also, this would also be a one against one attack, not leveraging additional hosts against the one to swamp its resources. And since the source also needs to generate massive numbers of packets for this to have any chance of succeeding, then the only leverage point I can see for the attacker is that he/she doesn't have to use as much bandwidth sending these tiny fragments, as he/she would to send other packets. This might allow for a larger number to be transmitted from the one attacking host. In this manner, the attacker would not be attacking bandwidth, but rather the IP stack of the destination host.

6. Correlations:

Across all of the sensors on our entire network, for a period of approximately nine months which are on record, "Tiny Fragments" alerts all together total 1,956,543 individual alerts, occupying 248MB of space when collected together in one file. On January 15, 2001, however, during a five hour period 1,631,381 of those alerts were generated, occupying 206MB when collected in a single file. Therefore 83.4% of all "Tiny Fragment" alerts were generated during this one five hour period.

I even went to the trouble of determining which hour periods had how many tiny fragment alerts, then I broke it down into minutes. There were some spotty areas where alerts came in at a rate of only about 25000 per minute, but there was a period of several minutes with numbers all right around 71,500 alerts per minute. My first thought was that this was the limit of the abilities of the host on the other end to generate these packets. It now seems that this first assumption was incorrect. The alert numbers above are accurate for the number of alerts seen. The snort sensors were configured to record all packets that generated an alert in the libpcap format. These files were stored locally on the remote sensor. The alerts were all transmitted back as syslog messages to a central reporting machine via syslog channel. After reviewing the actual packets captured, I determined that there number wasn't 1.6 million, but instead 4 million! So, it appears that 71,500 per minute was the limit of my systems' ability to send and receive syslog alert messages.

The two source IP addresses resolved to:
209.141.75.25 - user209-141-75-25.netcarrier.net
209.141.73.139 - user209-141-73-139.netcarrier.net

I was able to go back in our records and see other connections from netcarrier.net. I was also able to determine that they seemed intent on viewing web pages of one of our customers, but nobody else's. Also, the targets of their tiny fragments could have appeared to them (due to DNS answers from our nameservers) to have been the origin webserver of our customer. This certainly seemed to make the theory of a DoS attack attempt against this customer a possibility.

7. Evidence of active targeting:

These packets certainly were singling out two destination IP addresses within our network. Also, due to the other information uncovered about the website browsing habits of netcarrier.net, it seemed quite plausible that they may have been targeting what appeared to be the origin webserver of the one customer of ours whose website they kept browsing.

8. Severity:

Criticality: 3

These target IP addresses would have been proxied in to a farm of web servers. Each of these hosts is valuable, but not individually so.

Lethality: 3

This is a tough one to quantify. As far as availability, it is/was not lethal, as any network congestion on these segments, would have caused future users of our network services to be routed to completely different network segments. Also, the methods involved, if this was a DoS attempt didn't seem optimal for that purpose. On the other hand, however, this type of massive fragmentation bombardment could quite possibly take down a host and/or a network segment.

System Countermeasures: 4

Modern OS, well-patched, running minimal services. Really not much you can do to protect against a DoS like this though.

Network Countermeasures: 3

Not anything in place to block such traffic currently. IDS was able to catch this quite easily. This did show some weakness in our ability to receive all syslog alerts, but this would only be of issue if they were possibly trying to sneak other attacks in simultaneously.

9. Defensive recommendation:

I would continue to monitor connections to/from the netcarrier.net network. I would actually add logging rules to the snort sensor so that all such traffic could be observed. In addition, I would recommend altering the sensor configuration to insure that alert logs were stored locally as well as sent to a remote syslog server. I would also encourage detailed analysis of other events which transpired chronologically near any future occurrence of these tiny fragment floods, to try to see any attack which might be trying to sneak past us in all of the noise and confusion generated by this amount of traffic.

10. Multiple choice test question:

Packets on an ethernet segment will be fragmented once their size exceeds:

- A) 68 bytes
- B) 1500 bytes
- C) 4352 bytes
- D) 65535 bytes

Answer: B

Detect 5: synscan Probe for DNS Services

Packet Dumps

```
Mar 11 17:15:13 MV/IDS198/SYN FIN Scan: 216.63.85.125:53 -> nyc.network2.42.5:53
Mar 11 17:15:13 MV/IDS198/SYN FIN Scan: 216.63.85.125:53 -> nyc.network2.42.8:53
Mar 11 17:15:13 MV/IDS198/SYN FIN Scan: 216.63.85.125:53 -> nyc.network2.42.9:53
Mar 11 17:15:13 MV/IDS198/SYN FIN Scan: 216.63.85.125:53 -> nyc.network2.42.11:53
Mar 11 17:15:13 MV/IDS198/SYN FIN Scan: 216.63.85.125:53 -> nyc.network2.42.23:53
Mar 11 17:15:13 MV/IDS198/SYN FIN Scan: 216.63.85.125:53 -> nyc.network2.42.24:53
```


1. Source of Trace:

This scan was detected on my company's network.

2. Detect was generated by:

Linux sensor running Snort 1.6.3 with a combination of whitehats, snort.org, and customized rulesets.

3. Probability the source address was spoofed:

It is most likely that the source address is not spoofed. While completely possible, as this TCP traffic does not indicate a successful 3-way handshake has occurred, for this scan to be of value to the attacker, he/she would need to receive responses back from these packets. No other similar scans were observed during or around this time period, so there is little likelihood that this source address is a decoy (as can be produced in nmap).

4. Description of attack:

This scan is almost certainly a product of Psychoid's "synscan" tool. The [packet dumps](#) show some telltale signs of this tool. First, ID:39426 and a window size of 0x404 seem to indicate this quite strongly. Also, although not enough information existed to check how frequently the sequence and ack numbers changed, they did change once, which they should do at a precise interval if synscan was used. No matching CVE or BugTraq ID numbers were found.

5. Attack mechanism:

The synscan scanner produces packets with a distinct signature. Part of this unique behavior is scanning with syn/fin packets. Since this TCP flag combination will not occur naturally, it is a certain sign of some type of probe. Also, being an illegal flag combination the responses sent back by target hosts are not completely predictable. I have seen some mention in a whitehats forum (http://whitehats.com/cgi/forum/messages.cgi?bbs=get_topic&f=3&t=000019) that "I'm not sure if it's just because the port is open, but I have seen DNS servers respond to SYN/FIN packets with a SYN/FIN/ACK packet." Further in the section of Max Vision's analysis of the Ramen Worm that shows the packet dumps from his test of the Ramen Worm's usage of the synscan tool, (<http://www.whitehats.com/print/library/worms/ramen/breakout-synscan.txt>) a listening FTP service responded with a syn/fin, to which the scanner sent a reset. So it is possible that this scan is not only intended to find listening nameservers on port 53, but also could conceivably be used in single packet OS fingerprinting attempts. Also noteworthy is that these are TCP packets destined for port 53, so responding hosts accept TCP 53 traffic, and as such might also provide zone transfers.

6. Correlations:

These are the first and only packets ever detected from this source IP address. In fact, they are the only observed connections from the entire 216.63.85 class C network. Interestingly though, this host appears to have been in trouble before :) A google (<http://www.google.com>) search returned several other sites with mention of 216.63.85.125 and its connection attempts to others' hosts. In fact, the first two results were hits on sans.org! :) One of the other references to 216.63.85.125 was sent to sans.org (<http://www.sans.org/y2k/032401-1230.htm>) by Nina Barr who observed a DNS scan by the offending host of systems at Arizona State University. This scan was very similar to the one I captured. It also seems that in some previous scans, it targeted services other than named (port 53). It was observed on 02/18/2001 attempting lpd connections per <http://www.sans.org/y2k/022201.htm> and on 03/11/2001 (the same day as my detect) it was accused of conducting a "TCP port probe" at <http://www.usaor.net/users/jon/log.html> (a less than detailed description, but it is enough to know it is not just unfriendly toward my network). It also made the hosts.deny file listed at <http://tomii.erols.com/lusers.txt>.

I was also surprised to see that the offending netblock is registered to a physical address which is only about five miles away from my company's location.

MAX3 BUMTTX Dial Pool (NETBLK-SBCIS21767)
2701 W. 15th St.
PMB 236
Plano, TX 75075
US

Netname: SBCIS21767
Netblock: 216.63.85.0 - 216.63.85.127

Coordinator:
Southwestern Bell Internet Services (ZS44-ARIN) ipadmin@swbell.net
888-212-5411

7. Evidence of active targeting:

None. This scan seems to be probing large segments of IP space looking for responses from active DNS servers. Actually, the above scan crossed two class C networks that I was monitoring, indicating massive, progressive scanning of IP addresses in sequential order.

8. Severity: -1

Criticality: 3

It is difficult to assign a criticality to this, as this scan targeted anything with an IP address :) In this instance those systems scanned, were primarily web servers. DNS services were not running on any boxes in either of these two targeted networks. So, I'll assign a '3' here, as the web servers are rather important, but our network architecture could withstand losing many of them, and still provide services from the hundreds of others located in other networks, which could provide redundancy.

Lethality: 2

I only give Lethality a score of '2' because this probe will not be able to find any DNS servers in our network. So, we will not be attacked via DNS in these networks. The one problem here is that if what they're really after is a network map and/or host OS fingerprinting results, rather than DNS servers, they will have been able to determine information which may be valuable to an attacker.

System Countermeasures: 4

These systems are running a reasonably hardened OS and only very minimal services. Also, the lack of listening DNS servers is helpful for this scan.

Network Countermeasures: 2

These systems are not behind a firewall, so I can't say much for them there. The IDS sensor running there did detect this scan though, and would detect other port 53 traffic on those networks, so I'll give it a '2.'

9. Defensive recommendation:

Firewalls would be nice additions to these two networks. Short of that though, ACLs on the routers to silently block all traffic to unused ports would be recommended. It would add some measure of comfort to not solely rely on the hosts not running services (I think we've all been surprised to find a "new" service running on one of our hosts before. Sometimes it's due to a compromise, but most of the time it seems that another sysadmin has done it for testing or whatever other purpose. Invariably those test scenarios are never turned off either). Further, such ACLs could minimize the amount of information given out to those who might be trying to map our network(s).

I might also encourage that due to BIND's terrible security history, that on those servers that must provide DNS services, that they run D.J. Bernstein's djbdns (<http://cr.yip.to/djbdns.html>) as a more secure alternative.

10. Multiple choice test question:

When observing the familiar pattern of Syn/Fin, matching source and destination ports, ID:39426, and a window size of 0x404, what are we most likely seeing?

- A) A network scan using the Synscan tool.
- B) A network scan using the nmap tool.
- C) A scan originating from a host infected by the Ramen Worm.
- D) Answer A and/or C
- E) Answer B and/or C

Answer: D

This pattern indicates the synscan tool. Answer C is also plausible though, as the synscan tool was incorporated into the Ramen worm.

Assignment 2 - State of Intrusion Detection

Notes on the Practical Application of Network Intrusion Detection Systems in a Business Setting

The proper role of IDS systems and intrusion analysts in the enterprise is often not clearly defined.

Network intrusion detection is often perceived by management as being an active defense mechanism. This perception leads them to believe that if we, as analysts see something of concern happen, that we will be able to react in a sufficiently short period of time to prevent damage from being done. This notion that by the purchase of a commercial IDS product, or by investing corporate resources into building an open source IDS solution, that the organization is "buying" some type of shielding for their network, needs to be addressed. I think it is important to explore what the organizational goals are for running an IDS system. There are not too many situations in which organizations spend funds unless they clearly understand what it is they are trying to attain in exchange. As the technology industry experienced explosive growth, so to did budgeting and spending of resources without clearly defining and understanding the desired outcomes. Today management increasingly depends upon their technical staff to provide information about what is needed in the technology area. This is not always an area of expertise for the technical staff called upon to assist in making these decisions. Most disturbing though is the trend that systems, are added and resources expended based not on achieving defined business goals, but rather due to lack of understanding, in an attempt to "keep up." If IDS is the latest buzzword, then everyone wants an IDS. Knowing what one is and what one does is secondary. Knowing why one might be of benefit is not considered. As Benjamin D. Thomas notes "Intrusion detection is a necessary process that must be fully understood and executed to maintain network security." (http://www.linuxsecurity.com/feature_stories/feature_story-8.html).

While the above is perhaps an oversimplified description of the problem, I do believe that it is an

accurate description of what too often happens today. While simply stating "we're secure, we have a firewall, and an intrusion detection system even" sounds sexy and will make stockholders and other concerned parties breath an ignorant sigh of relief, that is not why (we would hope) such a system is purchased, implemented, and managed. I believe that the three primary reasons for implementing a network intrusion detection system within one's organization are:

- 1. Intrusion Detection - Sounds pretty simple. This means, though, that we run such systems to be able to detect an intrusion, after it has happened. This forensic application is probably the primary reason for an IDS system to exist.**
- 2. Network Design - Seeing trends in what type of traffic your network is subjected to, allows the organization to make intelligent, informed decisions about network architecture. This is the second most valuable contribution an IDS system can make, and this benefit is often overlooked and ignored.**
- 3. Active Defense - This one sounds really cool, and has some merit, but much less than is generally thought.**

A detailed look at each of these benefits is in order. First, detecting intrusions is the primary use of IDS systems. Although apparently obvious, often times the actual detection of intrusions is not what the implementers of IDS systems have in mind. Active response and defense are often what individuals think about when intrusion detection is discussed. When properly installed, configured, and maintained, however, intrusion detection systems are ideal for detecting when other network and host countermeasures have failed. The system and the analyst should be able to detect a system that was compromised. Further the analyst should be able to describe how the system was compromised, including service(s) attacked, when the incident occurred, and possibly even the tools used to conduct the exploit. Traffic from the compromised host can be analyzed to determine what types of new, unexpected activity that host generates. Also the analyst can assist with the containment of further damage by scrutinizing network activity to determine if other hosts may have been affected. Being able to rapidly detect compromises and assist with recovery is a valuable function and often can, in and of itself, justify the existence of an IDS system.

The second most valuable contribution an IDS system can make is in aiding network design efforts. This often overlooked benefit can be quite valuable, especially considering how modern networks are entities that constantly evolve. Knowing what you're up against "in the wild" can give valuable insights into how best to architect the network. An IDS system can help the network engineers to realize what types of services are being probed for, what services are actively being attacked, what types of router ACLs or firewall rules need to be put in place, what unexpected traffic is getting past routers and/or firewalls, and even anomalous traffic that could indicate a network problem. Often an IDS is only placed in front of access control devices (router or firewall), the information pertaining to what traffic was successfully getting past ACLs would not be available in this scenario. Knowing in advance whether one of the design goals for the network intrusion detection system is to be able to provide such information would allow for proper placement of sensors from the onset. This type of foresight, often lacking in modern implementations of IDS systems, can save the organization significant resources, in both personnel and capital expenditures.

Finally, active defense is what is often imagined when an IDS system is put into place. Often IDS systems with this capability never implement it because it can be a dangerous prospect to allow machines to blindly reconfigure your firewall(s) or other access control capabilities. If this functionality is discernible to adversaries outside of the organization, then they may choose to take advantage of this by spoofing source addresses and causing one's active defense responses to create a DoS (denial

of service) condition. If this truly is to be a role the IDS system is to play, however, then a detailed description of what this means to the organization must be drafted well in advance of implementation of the system. This draft can be the beginnings of, or supplementary information to an already existing, security policy. One function of this policy should be to codify what types of events the IDS is expected to react to, and what types of reactions are considered acceptable for the various types of anticipated stimuli. The importance of this type of planning cannot be overstated. Far too often an intrusion detection system is placed into an organization, at great cost, only to sit idle. According to Frederick M. Avolio in his composition "Foundations of Enterprise Network Security" (<http://www.avolio.com/Foundations.html>) such policy planning is critical as it "...matches the corporate culture, defines rules of behavior and individual responsibilities, matches responsibility and authority, details consequences of misuse or abuse, and lists services supported and controls to be employed."

Before an organization decides to incorporate an IDS system(s) into their network, they should have already come to terms with what specific business objective they are trying to achieve, and how the IDS will help them attain that objective. If any of the three benefits listed above is of value to the organization, and it helps the organization achieve its business goals, then an IDS system may be warranted.

Additionally, thought should be given to how an organization will manage such a device, or system of devices, after initial installation. In large organizations, there often will be the appropriate level of technical proficiency available to manage intrusion detection systems in-house. Often though, skilled network administrators are not well-versed in security issues. If it is decided that the organization possesses the necessary talent pool and that the IDS system will be maintained and manned by employees, then external training should be considered to give these soon-to-be analysts the best possible chance for success. The current "state of the art" training curriculum for intrusion detection analysts is the SANS GIAC Intrusion Detection track (<http://www.sans.org>).

For smaller organizations, cultivating this level of expertise can be unrealistic. With network intrusion detection systems, "The foreseen drawbacks include...long training times..." as observed by Aurobindo Sundaram in "An Introduction to Intrusion Detection" (<http://www.acm.org/crossroads/xrds2-4/intrus.html>). This leads to a rift in the information assurance capabilities of smaller organizations. To combat this problem, several companies are offering IDS analysis services. Some provide only the installation of automated systems, while others provide true analysis. A company that falls into this latter category is Bruce Schneier's Counterpane Internet Security (<http://www.counterpane.com>). While expensive and catering primarily to larger businesses, Counterpane provides "...real intrusion detection and response, not just an automated, programmed alert. It's human intelligence, with a wealth of expertise and experience analyzing intrusions and using critical judgment to instantly determine the appropriate response." (<http://www.counterpane.com/expertise.html>). While expensive, the cost of hiring and retaining this level of talent and experience would be many times more costly for an individual organization. There are other, smaller companies as well that provide similar services targeted at small to medium sized organizations. These companies have the added advantage of being able to adapt their services to meet the requirements of individual customers. This is where a company like ManISec comes in to play. "We provide nIDS architecture and highly qualified security professionals to handle all aspects of a secure nIDS system for your networks; from installation and configuration, to realtime monitoring and analysis and active response. Protecting your assets with an nIDS solution is now affordable and reliable." (<http://www.manisec.com/ids-monitoring.shtml>). It is good to see such companies providing more affordable solutions geared toward the small to medium-sized enterprise.

If an organization is able to clearly identify benefits to their operations which will assist them in meeting their business objectives, and they are able to properly plan for systems and expertise to monitor them, then IDS solutions can dramatically improve an organization's security posture. It is important though to

return to the philosophy of evaluating expenditures in this manner. It's better for the company, it's better for the intrusion analysts, and it's better for the bottom line.

Assignment 3 - Analyze This

Files Analyzed

While I didn't want to neglect any information in the data files provided which might be important, the assignment only called for analysis of at least five days' worth of detects. This allowed me to select a range of dates which seemed to have the most complete and reliable data. Many of the files included for dates in March and early April seemed to show inconsistencies which might hinder proper analysis. For this reason, I elected to analyze the detects from the period 04/15/2001 - 06/19/2001 (the last date in the dataset). This provides over two months of data, a sizable dataset, which should provide a good enough sample to begin to determine what is "average" network activity for this organization.

Below is a list of all files analyzed:

prac2/data_alert.010415
prac2/data_alert.010416
prac2/data_alert.010417
prac2/data_alert.010418
prac2/data_alert.010419
prac2/data_alert.010420
prac2/data_alert.010421
prac2/data_alert.010422
prac2/data_alert.010423
prac2/data_alert.010424
prac2/data_alert.010425
prac2/data_alert.010426
prac2/data_alert.010427
prac2/data_alert.010428
prac2/data_alert.010429
prac2/data_alert.010430
prac2/data_alert.010501
prac2/data_alert.010502
prac2/data_alert.010503
prac2/data_alert.010504
prac2/data_alert.010505
prac2/data_alert.010506
prac2/data_alert.010507

prac2/data_alert.010508
prac2/data_alert.010509
prac2/data_alert.010510
prac2/data_alert.010511
prac2/data_alert.010512
prac2/data_alert.010513
prac2/data_alert.010514
prac2/data_alert.010515
prac2/data_alert.010516
prac2/data_alert.010517
prac2/data_alert.010518
prac2/data_alert.010519
prac2/data_alert.010520
prac2/data_alert.010521
prac2/data_alert.010522
prac2/data_alert.010523
prac2/data_alert.010524
prac2/data_alert.010525
prac2/data_alert.010526
prac2/data_alert.010527
prac2/data_alert.010528
prac2/data_alert.010529
prac2/data_alert.010530
prac2/data_alert.010531
prac2/data_alert.010601
prac2/data_alert.010602
prac2/data_alert.010603
prac2/data_alert.010604
prac2/data_alert.010605
prac2/data_alert.010606
prac2/data_alert.010607
prac2/data_alert.010608
prac2/data_alert.010609
prac2/data_alert.010610
prac2/data_alert.010611
prac2/data_alert.010612
prac2/data_alert.010613
prac2/data_alert.010614
prac2/data_alert.010615
prac2/data_alert.010616
prac2/data_alert.010617
prac2/data_alert.010618
prac2/data_alert.010619
prac2/data_doublescan_010610
prac2/data_oos.010415
prac2/data_oos.010416
prac2/data_oos.010417
prac2/data_oos.010418
prac2/data_oos.010419
prac2/data_oos.010420
prac2/data_oos.010421
prac2/data_oos.010422

prac2/data_oos.010423
prac2/data_oos.010424
prac2/data_oos.010425
prac2/data_oos.010426
prac2/data_oos.010427
prac2/data_oos.010428
prac2/data_oos.010429
prac2/data_oos.010430
prac2/data_oos.010501
prac2/data_oos.010502
prac2/data_oos.010503
prac2/data_oos.010504
prac2/data_oos.010505
prac2/data_oos.010506
prac2/data_oos.010507
prac2/data_oos.010508
prac2/data_oos.010509
prac2/data_oos.010510
prac2/data_oos.010511
prac2/data_oos.010512
prac2/data_oos.010513
prac2/data_oos.010514
prac2/data_oos.010515
prac2/data_oos.010516
prac2/data_oos.010517
prac2/data_oos.010518
prac2/data_oos.010519
prac2/data_oos.010520
prac2/data_oos.010521
prac2/data_oos.010522
prac2/data_oos.010523
prac2/data_oos.010524
prac2/data_oos.010525
prac2/data_oos.010526
prac2/data_oos.010527
prac2/data_oos.010528
prac2/data_oos.010529
prac2/data_oos.010530
prac2/data_oos.010531
prac2/data_oos.010601
prac2/data_oos.010602
prac2/data_oos.010603
prac2/data_oos.010604
prac2/data_oos.010605
prac2/data_oos.010606
prac2/data_oos.010607
prac2/data_oos.010608
prac2/data_oos.010609
prac2/data_oos.010610
prac2/data_oos.010611
prac2/data_oos.010612
prac2/data_oos.010613

prac2/data_oos.010614
prac2/data_oos.010615
prac2/data_oos.010616
prac2/data_oos.010617
prac2/data_oos.010618
prac2/data_oos.010619
prac2/data_scans.010415
prac2/data_scans.010416
prac2/data_scans.010417
prac2/data_scans.010418
prac2/data_scans.010419
prac2/data_scans.010420
prac2/data_scans.010421
prac2/data_scans.010422
prac2/data_scans.010423
prac2/data_scans.010424
prac2/data_scans.010425
prac2/data_scans.010426
prac2/data_scans.010427
prac2/data_scans.010428
prac2/data_scans.010429
prac2/data_scans.010430
prac2/data_scans.010501
prac2/data_scans.010502
prac2/data_scans.010503
prac2/data_scans.010504
prac2/data_scans.010505
prac2/data_scans.010506
prac2/data_scans.010507
prac2/data_scans.010508
prac2/data_scans.010509
prac2/data_scans.010510
prac2/data_scans.010511
prac2/data_scans.010512
prac2/data_scans.010513
prac2/data_scans.010514
prac2/data_scans.010515
prac2/data_scans.010516
prac2/data_scans.010517
prac2/data_scans.010518
prac2/data_scans.010519
prac2/data_scans.010520
prac2/data_scans.010521
prac2/data_scans.010522
prac2/data_scans.010523
prac2/data_scans.010524
prac2/data_scans.010525
prac2/data_scans.010526
prac2/data_scans.010527
prac2/data_scans.010528
prac2/data_scans.010529
prac2/data_scans.010530

prac2/data_scans.010531
prac2/data_scans.010601
prac2/data_scans.010602
prac2/data_scans.010603
prac2/data_scans.010604
prac2/data_scans.010605
prac2/data_scans.010606
prac2/data_scans.010607
prac2/data_scans.010608
prac2/data_scans.010609
prac2/data_scans.010610
prac2/data_scans.010611
prac2/data_scans.010612
prac2/data_scans.010613
prac2/data_scans.010614
prac2/data_scans.010615
prac2/data_scans.010616
prac2/data_scans.010617
prac2/data_scans.010618
prac2/data_scans.010619

Overview

In preparing this security review of the university's network, I utilized a group files provided by the university which contained ASCII output from the snort lightweight IDS (<http://www.snort.org>). I also compared the current findings with those of Marc Bayerkohler, whose analysis of this same network took place on December 11, 2000 (http://www.sans.org/y2k/practical/Marc_Bayerkohler_GCIA.html).

It should be noted, that as stated in the "Files Analyzed" section of this document, only snort data from 04/15/2001-06/19/2001 was analyzed. This is due to some issues with the data provided prior to that. As for sensor placement and other defensive recommendations, I will address those in the appropriate section in this paper. I have one recommendation though, that isn't specifically a defensive action, but would assist myself or other future consultants called upon to assist with the security of the university's network. This would be to store, in addition to the ASCII alert, scan, and OOS files, snort's packet captures in the native libpcap format, and provide those to the analyst as well. This will allow the analyst to be able to work with this data in a standard format, without having to rely solely on text parsing. This information could prove valuable to an analyst who wants to analyze portions of data in greater detail.

Still, however, the data provided did allow for me to extract some important information. Further, the comparisons with Marc Bayerkohler's work also provide some useful information. All of this, can certainly be used to improve the state of network security at the university. Most of this information, though, will only be of assistance in future planning and network defense. Many of the traffic seen in the current dataset cannot be ruled on conclusively. To do so, I would need more information (specifically complete packet dumps, preferably in libpcap format).

Selected Detects

Example Alert:

05/11-09:44:31.703926 [**] UDP SRC and DST outside network [**] 63.250.213.73:1043 -> 233.28.65.227:5779

The most frequently seen alert source address, 63.250.213.73 is solely involved (all 717,352 alerts) with sending packets to 233.28.65.227, which is also the second most frequently seen alert destination address. Since both the source and destination addresses are outside of the MY.NET network, these communications are obviously suspect. One possible explanation for this anomalous traffic would be if an internal (MY.NET) host was spoofing the source address of these packets. Since this is UDP traffic, one-way communication channels such as this would be quite possible. If any response is expected or desired, then an asynchronous return channel might be involved. If the outside host knew the real source IP of the sender, then that host could send UDP traffic back in, establishing a bi-directional communication channel. If this is the situation here, then the source of the return traffic is also changed from 233.28.65.227, as that host doesn't appear in any alerts that don't involve two hosts outside of the MY.NET network. Unless, of course, the return communication channel is operating in a manner which doesn't activate any snort rules, thereby avoiding any logging.

04/17-03:28:25.186609 [**] Possible trojan server activity [**] 129.82.88.173:1358 -> MY.NET.200.130:27374

04/17-03:28:25.191391 [**] Possible trojan server activity [**] 129.82.88.173:1361 -> MY.NET.200.133:27374

04/17-03:28:25.198755 [**] Possible trojan server activity [**] 129.82.88.173:1366 -> MY.NET.200.138:27374

04/17-03:28:25.198982 [**] Possible trojan server activity [**] 129.82.88.173:1367 -> MY.NET.200.139:27374

04/17-03:28:25.206805 [**] Possible trojan server activity [**] 129.82.88.173:1372 -> MY.NET.200.144:27374

04/17-03:28:25.228948 [**] Possible trojan server activity [**] 129.82.88.173:1388 -> MY.NET.200.160:27374

04/17-03:28:25.236021 [**] Possible trojan server activity [**] 129.82.88.173:1393 -> MY.NET.200.165:27374

04/17-03:28:25.249904 [**] Possible trojan server activity [**] 129.82.88.173:1403 -> MY.NET.200.175:27374

04/17-03:28:26.204274 [**] Possible trojan server activity [**] 129.82.88.173:1413 -> MY.NET.200.185:27374

04/17-03:28:26.209710 [**] Possible trojan server activity [**] 129.82.88.173:1417 -> MY.NET.200.189:27374

04/17-03:28:26.212584 [**] Possible trojan server activity [**] 129.82.88.173:1403 -> MY.NET.200.175:27374

04/17-03:28:26.212922 [**] Possible trojan server activity [**] 129.82.88.173:1374 -> MY.NET.200.146:27374

04/17-03:28:26.213015 [**] Possible trojan server activity [**] 129.82.88.173:1376 -> MY.NET.200.148:27374

04/17-03:28:26.213061 [**] Possible trojan server activity [**] 129.82.88.173:1392 -> MY.NET.200.164:27374

04/17-03:28:26.213106 [**] Possible trojan server activity [**] 129.82.88.173:1394 -> MY.NET.200.166:27374

04/17-03:28:26.213151 [**] Possible trojan server activity [**] 129.82.88.173:1406 -> MY.NET.200.178:27374

04/17-03:28:26.213196 [**] Possible trojan server activity [**] 129.82.88.173:1408 -> MY.NET.200.180:27374
04/17-03:28:26.219752 [**] Possible trojan server activity [**] 129.82.88.173:1423 -> MY.NET.200.195:27374

This host just seems to be trolling for hosts infected by the SubSeven trojan. It was revealing in that it found 24 hosts that answered on port 27374.

04/20-11:05:25.387466 [**] SYN-FIN scan! [**] 210.160.190.244:21 -> MY.NET.195.156:21
04/20-11:05:25.387745 [**] SYN-FIN scan! [**] 210.160.190.244:21 -> MY.NET.195.155:21
04/20-11:05:25.480245 [**] SYN-FIN scan! [**] 210.160.190.244:21 -> MY.NET.195.161:21
04/20-11:05:25.567674 [**] SYN-FIN scan! [**] 210.160.190.244:21 -> MY.NET.195.165:21
04/20-11:05:25.928524 [**] SYN-FIN scan! [**] 210.160.190.244:21 -> MY.NET.195.183:21
04/20-11:05:26.649259 [**] SYN-FIN scan! [**] 210.160.190.244:21 -> MY.NET.195.218:21
04/20-11:05:37.733723 [**] SYN-FIN scan! [**] 210.160.190.244:21 -> MY.NET.198.9:21
04/20-11:05:37.825062 [**] SYN-FIN scan! [**] 210.160.190.244:21 -> MY.NET.198.14:21
04/20-11:05:37.825203 [**] SYN-FIN scan! [**] 210.160.190.244:21 -> MY.NET.198.15:21
04/20-11:05:38.183491 [**] SYN-FIN scan! [**] 210.160.190.244:21 -> MY.NET.198.29:21
04/20-11:05:38.183811 [**] SYN-FIN scan! [**] 210.160.190.244:21 -> MY.NET.198.31:21
04/20-11:05:38.183968 [**] SYN-FIN scan! [**] 210.160.190.244:21 -> MY.NET.198.33:21
04/20-11:05:38.273783 [**] SYN-FIN scan! [**] 210.160.190.244:21 -> MY.NET.198.34:21
04/20-11:05:38.274084 [**] SYN-FIN scan! [**] 210.160.190.244:21 -> MY.NET.198.37:21

This host is also conducting a noisy scan. This time looking for ftp servers. The source host is scanning large numbers of targets in the MY.NET network. The packets use the illegal TCP flag combination of syn/fin, which caused snort to alert on these packets. What was interesting here though, was that the scanning proceeded from the MY.NET.195 network to the MY.NET.198 network and only slowed down by nine seconds during the transition. Although this scanner went from MY.NET.1.x to MY.NET.254.x in only 22 minutes, a jump of three class c address blocks in only nine seconds seemed a bit fast. This kind of jump could indicate some type of previous knowledge about which hosts are up or not on the university's network.

04/24-03:01:08.677196 [**] Possible trojan server activity [**] 216.220.164.133:2102 -> MY.NET.217.2:27374
04/24-03:01:08.677242 [**] Possible trojan server activity [**] MY.NET.217.2:27374 -> 216.220.164.133:2102
04/24-03:01:08.961437 [**] Possible trojan server activity [**] MY.NET.217.29:27374 -> 216.220.164.133:2129
04/24-03:01:08.962044 [**] Possible trojan server activity [**] MY.NET.217.25:27374 -> 216.220.164.133:2125
04/24-03:01:08.972365 [**] Possible trojan server activity [**] MY.NET.217.22:27374 -> 216.220.164.133:2122
04/24-03:01:09.405503 [**] Possible trojan server activity [**] 216.220.164.133:2102 -> MY.NET.217.2:27374
04/24-03:01:09.405605 [**] Possible trojan server activity [**] MY.NET.217.2:27374 -> 216.220.164.133:2102
04/24-03:01:10.286836 [**] Possible trojan server activity [**] MY.NET.217.21:27374 -> 216.220.164.133:2121
04/24-03:01:15.700245 [**] Possible trojan server activity [**] 216.220.164.133:2303 -> MY.NET.217.205:27374

04/24-03:01:15.700711 [**] Possible trojan server activity [**] MY.NET.217.205:27374 -> 216.220.164.133:2303
04/24-03:01:15.702772 [**] Possible trojan server activity [**] 216.220.164.133:2304 -> MY.NET.217.206:27374
04/24-03:01:15.706108 [**] Possible trojan server activity [**] MY.NET.217.206:27374 -> 216.220.164.133:2304
04/24-03:01:15.717080 [**] Possible trojan server activity [**] 216.220.164.133:2306 -> MY.NET.217.208:27374
04/24-03:01:15.728602 [**] Possible trojan server activity [**] MY.NET.217.209:27374 -> 216.220.164.133:2307
04/24-03:01:15.736059 [**] Possible trojan server activity [**] 216.220.164.133:2308 -> MY.NET.217.210:27374
04/24-03:01:15.740313 [**] Possible trojan server activity [**] 216.220.164.133:2310 -> MY.NET.217.212:27374
04/24-03:01:16.057054 [**] Possible trojan server activity [**] 216.220.164.133:2345 -> MY.NET.217.247:27374
04/24-03:01:16.070812 [**] Possible trojan server activity [**] 216.220.164.133:2351 -> MY.NET.217.253:27374

The host 216.220.164.133 was observed communicating with MY.NET.217.

Top Talkers

The "top ten lists" shown below are properly deciphered as "number of occurrences", "whitespace", then "Thing counted (i.e. IP address, port #, etc.). These are the result of shell commands such as:

```
cat data_alert* | "then some awk, grep, and other shell tools" | sort | uniq -c | sort -r -n -k 1
```

Top 10 source addresses

```
717352 63.250.213.73  
655428 63.250.213.119  
158583 MY.NET.160.114  
156180 63.250.213.26  
127124 212.179.58.200  
118645 MY.NET.150.225  
99514 MY.NET.60.16  
69130 205.188.233.153  
68985 205.188.233.185  
58119 205.188.233.121
```

This list contains the ten most active hosts during the analysis period. This is defined as the ten hosts who appeared as the source address the most frequently in all alert, scan, and oos files. It is alarming to see some hosts from MY.NET appearing in this list. These hosts will be scrutinized more deeply.

Top 10 source ports

```
718550 1042
```


658948 1036
214930 28800
158578 777
156859 21
151118 1038
96522 13139
94890 3697
78899 6112
52308 24452

Generally the source ports are probably less interesting than the source IP, destination IP, and destination port. Here, however, we still see some interesting information. 28800 might be involved with gaming. Although I usually think of UDP 28800 as the destination port for MSN Gaming Zone activity, I'm not certain if these gaming sessions might also use source port 28800. The source port 21 was seen very often in scans of hosts for port 21. These scans used both source and destination port 21. Port 777 is interesting because it is a port below 1024. My first guess was maybe ssh connections (since by default an ssh client will open a port below 1024 to make its connection to a sshd server on port 22). As it turned out though, after looking at the source port 777 traffic, it was only targeting high numbered ports and all of the connections were UDP.

Top 10 destination addresses

734213 233.28.65.62
717473 233.28.65.227
147087 233.28.65.164
128657 MY.NET.150.220
23933 233.28.65.222
23760 158.75.57.4
20795 MY.NET.71.69
18219 24.13.123.8
16671 MY.NET.145.166
16168 MY.NET.15.214

This could be almost as alarming as the occurrence of MY.NET addresses in the top ten source IP list. Six of the top ten most common destinations are outside of the MY.NET network. The top three are all in the 233.28.65 network, which are reserved networks allocated by IANA for multicast traffic. Hopefully the other addresses' appearances are the result of what can be determined to be legitimate traffic. While outside destinations could certainly be expected, not this number of them in the alerts file.

Top 10 destination ports

1651472 5779
539695 21
268554 53
235968 28800
217473 6970
129326 1234
94865 27005
91956 13139
84529 27374
69569 6112

Port 5779 is the clear winner here. The ports database at snort.org didn't return anything for this port (<http://www.snort.org/Database/portsearch.asp>), but it's frequency makes it suspect. This one will be further investigated as we get deeper into who's talking to who here. Ports 21 and 53 aren't too surprising, as they're common target services for port scanners. Port 28800 comes up again, again possibly MSN Gaming Zone game hosting is happening. Port 1234 is sometimes associated with Ultors trojan. Port 13139 didn't turn up anything in the previously mentioned snort.org port database. I was able to find a reference to it on GameSpy Arcade (<http://www.gamespyarcade.com/support/firewalls.shtml>) which refers to use of this port for "Custom UDP Pings," possibly attempting to determine latency measurements for use in gaming?

Total number of IP addresses observed
245069

Total number of IP addresses from MY.NET active/observed
34452

As can be seen from the above two statements, this is an extremely busy network. Most telling is that of 65535 IP addresses in the university's class B address space, 34452 or 52.6% were active in the snort logs. While some of these may not be active hosts (they may have simply been listed in a scan log as an outside entity conducted a wide scan, targeting some IP addresses that were not active), this is still strong indication that this is a massive network.

External Source Addresses and Registration Information

Selection of hosts to perform lookups on was based upon those hosts who were in any of the top talkers lists. The top talkers who were seen conducting any type of attack were evaluated for placement on this list. Special attention was paid to hosts on the Watchlist.

```
212.179.27.6
[whois.ripe.net]
% This is the RIPE Whois server.
% The objects are in RPSL format.
% Please visit http://www.ripe.net/rpsl for more information.
% Rights restricted by copyright.
% See http://www.ripe.net/ripencr/pub-services/db/copyright.html
```

```
inetnum: 212.179.27.4 - 212.179.27.7
netname: ADI-ASSOCIATION
descr: ADI-ASSOCIATION-SERIAL
country: IL
admin-c: NP469-RIPE
tech-c: NP469-RIPE
status: ASSIGNED PA
notify: hostmaster@isdn.net.il
mnt-by: RIPE-NCC-NONE-MNT
changed: hostmaster@isdn.net.il 20000106
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: 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

212.179.58.2
[whois.ripe.net]
% This is the RIPE Whois server.
% The objects are in RPSL format.
% Please visit <http://www.ripe.net/rpsl> for more information.
% Rights restricted by copyright.
% See <http://www.ripe.net/ripenc/pdb-services/db/copyright.html>

inetnum: 212.179.58.0 - 212.179.58.255
netname: NV-PICTUREVISION
descr: network
country: IL
admin-c: NP469-RIPE
tech-c: NP469-RIPE
status: ASSIGNED PA
notify: hostmaster@isdn.net.il
mnt-by: RIPE-NCC-NONE-MNT
changed: hostmaster@isdn.net.il 20000229
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: 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

212.179.58.200
[whois.ripe.net]
% This is the RIPE Whois server.
% The objects are in RPSL format.
% Please visit <http://www.ripe.net/rpsl> for more information.
% Rights restricted by copyright.
% See <http://www.ripe.net/ripenc/pub-services/db/copyright.html>

inetnum: 212.179.58.0 - 212.179.58.255
netname: NV-PICTUREVISION
descr: network
country: IL
admin-c: NP469-RIPE
tech-c: NP469-RIPE
status: ASSIGNED PA
notify: hostmaster@isdn.net.il
mnt-by: RIPE-NCC-NONE-MNT
changed: hostmaster@isdn.net.il 20000229
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: 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

129.82.88.173
[whois.arin.net]
Colorado State University (NET-CSUNET)
Computer Center
Fort Collins, CO 80523
US

Netname: CSUNET
Netblock: 129.82.0.0 - 129.82.255.255

Coordinator:
McPherson, Stew (SM83-ARIN) stew@YUMA.ACNS.COLOSTATE.EDU
+1-970-491-7214 (FAX) +1-970-491-1958

Domain System inverse mapping provided by:

YUMA.ACNS.COLOSTATE.EDU 129.82.100.64
RS1.NETSOL.COM 216.168.224.207

Record last updated on 11-Dec-2000.
Database last updated on 26-Jun-2001 23:09:41 EDT.

The ARIN Registration Services Host contains ONLY Internet
Network Information: Networks, ASN's, and related POC's.
Please use the whois server at rs.internic.net for DOMAIN related
Information and whois.nic.mil for NIPRNET Information.

This host was included because it was conducting a massive scan of the university's network
searching for hosts that were infected with the SubSeven trojan. Unfortunately it found 24 of them.

158.75.57.4
POLIP (NET-TORUNPOLIP2)
Computer Centre, Nicolaus Copernicus University
ul. Chopina 12/18, 87-100 Torun, Poland
PL

Netname: TORUNPOLIP2
Netblock: 158.75.0.0 - 158.75.255.255

Coordinator:
Szewczak, Zbigniew S. (ZSS-ARIN) zssz@TORUN.PL
(56) 260-17 ext. 70

Domain System inverse mapping provided by:

ALFA.CS.TORUN.PL 158.75.10.75
BILBO.NASK.ORG.PL 148.81.16.51

Record last updated on 11-Oct-1995.
Database last updated on 26-Jun-2001 23:09:41 EDT.

The scans from 158.75.57.4 are interesting. They target the destination port TCP 6346, usually
associated with Gnutella. This doesn't look like actual Gnutella traffic, however, because 158.75.57.4
scans multiple MY.NET hosts apparently looking for services answering on that port. Additionally
interesting is the use of the TCP flags 21S. The syn isn't surprising, but the use of reserved flags 1 and
2 are. These reserved flags are now being seen in some legitimate traffic, as they are now being used
in implementations of ECN (Explicit Congestion Notification). The behavior here though, definitely
looks more like a scan than any legitimate use of ECN.

211.240.28.66

Korea Internet Information Service V1.0 (created by KRNIC, 1999.6)

query: 211.240.28.66

* ÇÑ±Û ±â°ü,í¿i´ëÇÑ whois Á¶È,´Â À¥(http://whois.nic.or.kr)¿j¼
Çí½Ã±â´Ù¶ø´í´Ù.

Á¶È,Çí½Ã Çø´ç IPÁÖ¼Ò´Â ¾Æ·jÀÇ °jÀÔ±â°ü¿j ÇÒ´çµÈ °í·ÀÔ´í´Ù.

ENGLISH

IP Address : 211.240.28.64-211.240.28.127

Connect ISP Name : ELIMNET

Connect Date : 20010111

Registration Date : 20010601

Network Name : ITBUSINESS

[Organization Information]

Orgnization ID : ORG213206

Name : ITBUSINESS

State : SEOUL

Address : 202 NAMKANG B/D, 692-3 DAERIM3 DONG, YOUNGDEUNGPO GU,

Zip Code : 150-073

[Admin Contact Information]

Name : JAEHAK HAN

Org Name : ITBUSINESS

State : SEOUL

Address : 202 NAMKANG B/D, 692-3 DAERIM3 DONG, YOUNGDEUNGPO GU,

Zip Code : 150-073

Phone : +82-2-841-4114

Fax : +82-2-841-0815

E-Mail : domain@elim.net

[Technical Contact Information]

Name : JAEHAK HAN

Org Name : ITBUSINESS

Address : 202 NAMKANG B/D, 692-3 DAERIM3 DONG, YOUNGDEUNGPO GU,

Zip Code : 150-073

Phone : +82-2-841-4114

Fax : +82-2-841-0815

E-Mail : domain@elim.net

210.160.190.244

[whois.nic.ad.jp]

[JPNIC database provides information on network administration. Its use is]

[restricted to network administration purposes. For further information, use]

['whois -h whois.nic.ad.jp help'. To suppress Japanese output, add '/e' at]

[the end of command, e.g. 'whois -h whois.nic.ad.jp xxx/e'.]

[(Open Computer Network)
SUBA-131-409 210.160.190.0
(Nangoku Corporation)
NANGOKU [210.160.190.240 <-> 210.160.190.255] 210.160.190.240/28

Suggested abuse email addresses include: postmaster@nangoku.co.jp,
postmaster@linux.nangoku.co.jp, and abuse@ocn.ad.jp.

194.78.218.248
[whois.ripe.net]
% This is the RIPE Whois server.
% The objects are in RPSL format.
% Please visit <http://www.ripe.net/rpsl> for more information.
% Rights restricted by copyright.
% See <http://www.ripe.net/ripenc/pub-services/db/copyright.html>

inetnum: 194.78.216.0 - 194.78.219.255
netname: BE-SKYNET
descr: SKY-ADSL-PRO/bru-stro
descr: ADSL Access Infrastructure
descr: Belgacom Skynet SA/NV
country: BE
admin-c: SN2068-RIPE
tech-c: SN2068-RIPE
rev-srv: ns1.skynet.be
rev-srv: ns2.skynet.be
rev-srv: ns3.skynet.be
rev-srv: ns4.skynet.be
status: ASSIGNED PA
mnt-by: SKYNETBE-MNT
changed: piet@skynet.be 20001204
source: RIPE

route: 194.78.0.0/16
descr: SKYNETBE-CUSTOMERS
origin: AS5432
notify: noc@skynet.be
mnt-by: SKYNETBE-MNT
changed: jef@interpac.be 19960506
changed: jfs@skynet.be 19990420
source: RIPE

person: Skynet NOC administrators
address: Belgacom Skynet SA/NV
address: rue colonel Bourg 124
address: B-1140 Brussels
address: Belgium
phone: +3227061311
fax-no: +3227269311

e-mail: noc@skynet.be
nic-hdl: SN2068-RIPE
remarks: -----
remarks: Abuse notifications to: abuse@skynet.be
remarks: Network problems to: noc@skynet.be
remarks: Peering requests to: peering@skynet.be
remarks: -----
notify: noc@skynet.be
mnt-by: SKYNETBE-MNT
changed: piet@skynet.be 20000320
changed: piet@skynet.be 20010129
source: RIPE

Conducting lpd scans of the university's network, presumably to find vulnerable services to later attack.

216.220.168.222
[whois.arin.net]
Pennsylvania Online (NETBLK-PAONLINE-1)
PO Box 6501
Harrisburg, PA 17112
US

Netname: PAONLINE-1
Netblock: 216.220.160.0 - 216.220.175.255
Maintainer: PAON

Coordinator:
Peace, George (GP11-ARIN) george@PAONLINE.NET
(717) 657-0000 (FAX) (717) 657-0132

Domain System inverse mapping provided by:

NS1.PAONLINE.COM 198.69.90.250
NS2.PAONLINE.COM 198.69.90.11
NS3.PAONLINE.COM 207.44.20.1

ADDRESSES WITHIN THIS BLOCK ARE NON-PORTABLE

Record last updated on 27-Feb-2001.
Database last updated on 26-Jun-2001 23:09:41 EDT.

The ARIN Registration Services Host contains ONLY Internet
Network Information: Networks, ASN's, and related POC's.
Please use the whois server at rs.internic.net for DOMAIN related
Information and whois.nic.mil for NIPRNET Information.

Was exchanging traffic with MY.NET.217.2 in what may have been SubSeven trojan activity. There is
not enough information to say conclusively.

212.179.79.2

[whois.ripe.net]

% This is the RIPE Whois server.

% The objects are in RPSL format.

% Please visit <http://www.ripe.net/rpsl> for more information.

% Rights restricted by copyright.

% See <http://www.ripe.net/ripenc/db/copyright.html>

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

Correlations

In referencing Marc Bayerkohler's review of the university's network (see link to his work in the Overview section of this document), I will concentrate on furthering the trending he did on certain IP addresses. Marc based some of his work on comparing the current (at that time) state of certain hosts that had been highlighted as top sources and destinations previously by yet another analyst, Lenny Zeltzer (http://www.sans.org/y2k/practical/Lenny_Zeltser.htm). Marc began with the top alert destination hosts.

Previously Observed Top Alert Destination Hosts

Host	Original # of Alerts	Last # of Alerts	Current # of Alerts	Updated Status
MY.NET.253.105	22118	47	6	Minor Target
MY.NET.217.2	4197	6	54	Low Activity - Possible Trojan Activity, Though
MY.NET.253.41	4176	4387	640	Increase Monitoring - Decreased Volume, Still Quite Active Though
MY.NET.100.230	3462	749	5756	Conducting outside scans - Possible compromise

Previously Observed Top Alert Source Hosts

Host	Original # of Alerts	Last # of Alerts	Current # of Alerts	Updated Status
202.38.128.188	22338	0	0	Issue Appears Resolved
MY.NET.253.12	18869	0	3	Still Appears To Be Okay - Two Curious Alerts But That's All
204.60.176.2	13619	0	0	Issue Appears Resolved
159.226.45.3	5066	1558	703	Watchlist Host - But Activity Seems Acceptable, SMTP and Ident
142.150.225.137	4594	0	0	Issue Appears Resolved

Previously Observed Top Scan Destination Hosts

Host	Original # of Alerts	Last # of Alerts	Current # of Alerts	Updated Status
MY.NET.101.89	4115	253	1	Issue Appears Resolved
MY.NET.70.234	2238	52	1	One Alert Possible Scan For Trojan - No Reply From This Host, Low Threat
MY.NET.179.78	2231	461	425	Some Connection Flurries Tripping Portscan Preprocessor, A Few Other Odd Connections - Would Like Additional Info - This Host Bears Continued Watching
MY.NET.97.73	1252	6	1	One Alert Possible Scan For Trojan - No Reply From This Host, Issue Appears Resolved

Previously Observed Top Scan Source Hosts

Host	Original # of Alerts	Last # of Alerts	Current # of Alerts	Updated Status
24.2.169.101	65864	0	0	Issue Appears Resolved
202.235.50.12	30363	0	0	Issue Appears Resolved
208.220.120.13	23391	0	0	Issue Appears Resolved
24.13.87.239	21022	0	0	Issue Appears Resolved
202.38.128.188	20762	0	0	Issue Appears Resolved

While not strictly correlation material, for completeness, and for any possible benefit to future auditors of the university's network, the current information for these four categories is listed below:

Current Top 10 Alert Destination Hosts

Host	Current # of Alerts
233.28.65.62	734213
233.28.65.227	717473
233.28.65.164	147087
MY.NET.150.220	128154
233.28.65.222	23933
MY.NET.71.69	20780
MY.NET.15.214	16139
233.28.65.171	9093
MY.NET.202.222	8442
233.28.65.170	8094

Current Top 10 Alert Source Hosts

Host	Current # of Alerts
63.250.213.73	717352
63.250.213.119	655428
63.250.213.26	156180
212.179.58.200	127124
206.190.36.120	49926
63.250.213.124	28859
63.250.213.122	23513
205.167.0.160	20779
147.52.74.115	16137
211.240.28.66	14348

Current Top 10 Scan Destination Hosts

Host	Current # of Alerts
158.75.57.4	23751
24.13.123.8	18215
MY.NET.145.166	16667
MY.NET.178.154	15034
MY.NET.110.33	14868
MY.NET.145.197	14256
MY.NET.108.15	12795

MY.NET.178.222 12681

MY.NET.110.169 12564

MY.NET.71.28 10650

Current Top 10 Scan Source Hosts

Host	Current # of Alerts
------	---------------------

MY.NET.160.114	158583
----------------	--------

MY.NET.150.225	118644
----------------	--------

MY.NET.60.16	99317
--------------	-------

205.188.233.153	69130
-----------------	-------

205.188.233.185	68985
-----------------	-------

205.188.233.121	58119
-----------------	-------

MY.NET.150.41	46631
---------------	-------

193.253.243.190	43135
-----------------	-------

MY.NET.229.74	40678
---------------	-------

213.93.23.218	37543
---------------	-------

OOS Files Analysis

It appears that the OOS (Out Of Spec) files are created of a single criteria, which is that they are TCP packets with illegal TCP flag combinations. In the dataset I selected to analyze (all files from April 15th - June 19th, 2001) there were 66 oos files present. When providing an ASCII dump of packets, snort, by default, includes tallies by IP protocol. All 66 of these files show 100% of packets to be of the TCP protocol. Further, the shell command string:

```
[wes@somehost somedirectory]$ sum=0 ; for i in `cat data_oos.010* | grep 'TCP:' | awk '{print $2}'` ;  
do sum=$((sum+$i)) ; done ; echo $sum  
68504
```

Produces the sum total of all packets, which is 68504 TCP packets (and total packets) included in the oos files.

Further distillation of the data available in the oos files revealed:

Top 10 oos unique source ip addresses

16584 211.240.28.66

8877 61.13.106.35

7990 210.160.190.244

5111 132.248.100.200

3869 206.139.131.244

3130 192.168.0.1

2832 62.238.69.199

2830 211.130.90.210

1716 158.75.57.4

1512 199.183.24.194

Top 10 oos unique source ports

51601 21
750 6346
507 0
265 18245
188 20
72 111
64 706
61 6699
51 6688
34 1531

Some of the most common source and destination ports (6688 and 6699) in the oos files are commonly associated with Napster peer-peer networking.

Top 10 oos unique destination ip addresses

5392 MY.NET.100.165
711 MY.NET.98.139
709 MY.NET.109.234
529 MY.NET.253.41
502 MY.NET.253.43
502 MY.NET.253.42
373 MY.NET.218.46
357 MY.NET.98.88
345 MY.NET.253.125
271 MY.NET.253.114

Top 10 oos unique destination ports

51614 21
6275 80
4081 6346
1611 25
261 21536
183 6347
165 1214
160 6688
156 0
124 113

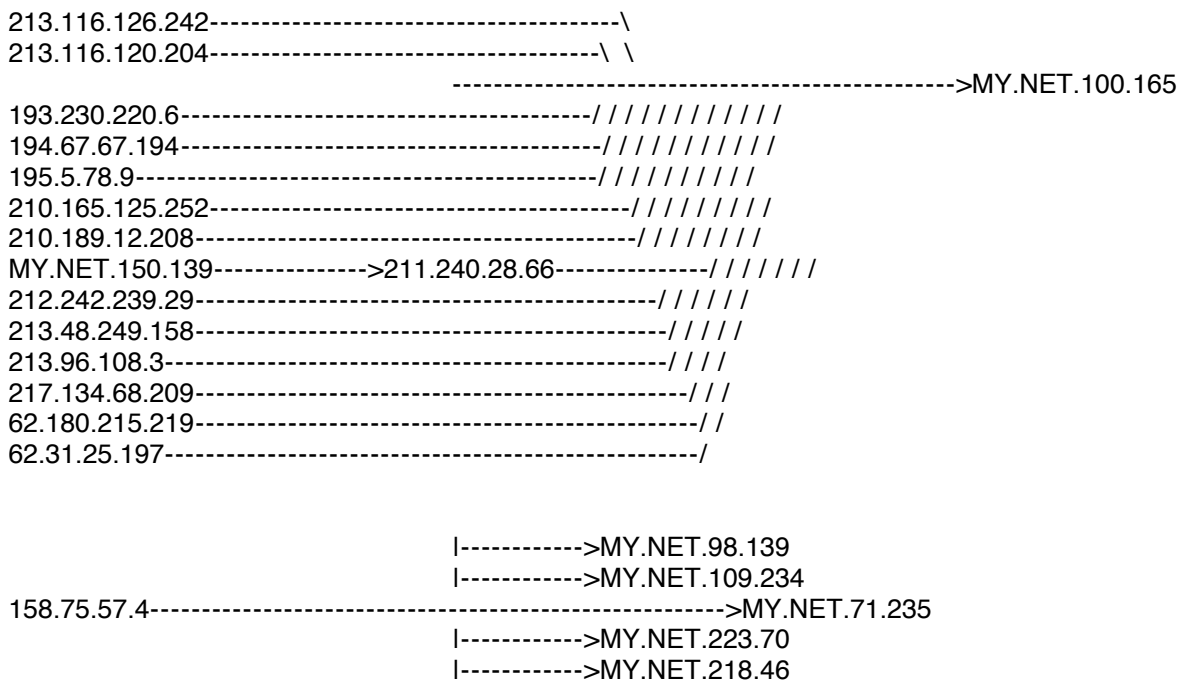
Top 10 address pairs

1257 213.116.126.242 MY.NET.100.165
1237 213.116.120.204 MY.NET.100.165
709 158.75.57.4 MY.NET.98.139
708 158.75.57.4 MY.NET.109.234
521 199.183.24.194 MY.NET.253.41
496 199.183.24.194 MY.NET.253.43
495 199.183.24.194 MY.NET.253.42
378 213.116.160.28 MY.NET.100.165
367 213.116.166.188 MY.NET.100.165

Top 10 port pairs

51594 21 21
 325 0 6346
 261 18245 21536
 72 111 111
 34 1531 2099
 24 4176 5501
 24 1419 412
 20 6346 1750
 19 1603 6346
 19 1147 6346

An ASCII representation of a couple of link graphs used in the analysis of oos files (spacing works great in lynx, but font selection doesn't seem to work to well in Netscape or Mozilla). Hosts included were based upon some of the more active hosts in the oos files.



Internal Host Issues

MY.NET.202.34 This host is of concern at this time. It was observed in bidirectional communication with an outside host (207.55.74.26). MY.NET.202.34 was using a high numbered port, and was communicating with port 27374 on the remote host. Port 27374 is commonly associated with the SubSeven trojan. This channel of communications certainly is of concern and may indicate a compromise of host MY.NET.202.34. This host should be carefully examined and more information about this activity should be acquired.

MY.NET.180.1
MY.NET.180.185
MY.NET.180.192
MY.NET.181.105
MY.NET.181.112
MY.NET.181.172
MY.NET.181.180
MY.NET.181.37
MY.NET.182.107
MY.NET.182.119
MY.NET.182.138
MY.NET.182.19
MY.NET.182.71
MY.NET.182.95
MY.NET.182.98
MY.NET.184.200
MY.NET.184.28
MY.NET.184.29
MY.NET.185.21
MY.NET.185.28
MY.NET.185.73
MY.NET.188.1
MY.NET.198.171
MY.NET.198.179

A SubSeven scan from an outside host revealed all 24 of these hosts responding to a connection attempt to port 27374. These hosts may very well be infected by the SubSeven trojan, and should all be inspected to verify their integrity.

There are large numbers of alerts reporting packets whose source and destination addresses are both outside the network. I already discussed one possible explanation for this traffic involving covert channels. Looking at the alerts, they seem to revolve around certain large class a and class b networks. It seems more likely (although this doesn't rule out any hostile activity) that there is some routing or other network configuration error. Routing devices around the university's network should be examined to verify that they are not causing these anomalous alerts.

Defensive Recommendations

Some network architecture changes appear to offer benefit to the university. One thing that would make securing this network more realistic would be subnetting this class B network into smaller subnets. This should be done in a way which separates the university's machines into logical segments, such as classrooms, student labs, faculty machines, administrative workstations, servers, etc. This would provide a level of manageability which appears to be currently unavailable. This would allow for enhanced IDS functionality (by segregating sensors onto separate subnets, and monitoring different subnets for different types of activity, based upon what is expected for the particular function of each subnet), quite possibly improved network performance, potentially increased security by allowing for more precise access controls tailored to each subnet, and improved flexibility in scaling the

network for future growth.

Touched on above, the idea of adding additional sensors could be beneficial. Separating the traffic analysis across multiple sensors monitoring multiple subnets would allow for faster, more useful analysis. This is due to the fact that the analysts could more readily determine what traffic is appropriate to see in a student lab versus a faculty member's workstation, versus an administration office mail server. These widely varying functions justify quite different intrusion detection signatures and general focus.

It is not readily apparent, but if lacking, the university could benefit from some sort of ingress and egress filtering at the router(s) and/or firewall device(s). Besides protecting potentially vulnerable hosts. It could also contain damages caused by such hosts if/when they are ever under the control of a hostile entity.

I would additionally insure that the sensor(s) were configured to store packet dumps of all packets alerted on in the native libpcap format. This format is considerably smaller on drive space usage than ASCII dumps, even more so when compressed with a tool such as gzip. I would then make certain that these files were available to future analysts attempting to audit this network.

Analysis Process

I began by using UNIX's wget tool to retrieve all data files from the website indicated in the GCIA assignment. I then reviewed the files to try to insure that they were properly labeled. I encountered problems with the integrity of one of the gzip archives. After review, I decided that the most consistent data began to appear in mid-April. I then decided that analysis would be best conducted on data from April 15th through the last date available (June 19th). I then put together some [simple shell scripts](#), which called upon tools such as awk, grep, uniq, sort, and wc. These seemed to be able to adequately dissect the large amount of information. I used similar shell commands to do various other small tasks to prepare data or count detects for other collateral material throughout the Analyze This portion of this document.

Appendix

List of Written References (online references listed within the body of the document)

74 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 t.....
00 00 00 00 00 00 00 00 00 00 00 03 E7 18 F7 FF BF
18 F7 FF BF 19 F7 FF BF 19 F7 FF BF 1A F7 FF BF
1A F7 FF BF 1B F7 FF BF 1B F7 FF BF 25 38 78 25%8x%
38 78 25 38 78 25 38 78 25 38 78 25 38 78 25 38 8x%8x%8x%8x%8x%8
78 25 38 78 25 38 78 25 32 33 36 78 25 6E 25 31 x%8x%8x%236x%n%1
33 37 78 25 6E 25 31 30 78 25 6E 25 31 39 32 78 37x%n%10x%n%192x
25 6E 90 90 90 90 90 90 90 90 90 90 90 90 90 90 %n.....
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90

Detect 1 - Scans from this attacking host (148.243.136.7)

Mar 5 04:25:47 src@dfw1a-core/10.24.0.123 snort[27872]: MV/IDS10/portmap-request-rstatd: 148.243.136.7:837 -> test.network.70.133:111

Mar 5 04:25:47 src@dfw1a-core/10.24.0.123 snort[27872]: MV/IDS362/shellcode-x86-nops-udp: 148.243.136.7:838 -> test.network.70.133:610

Mar 5 04:25:49 src@dfw1a-core/10.24.0.123 snort[27872]: MV/IDS362/shellcode-x86-nops-udp: 148.243.136.7:838 -> test.network.70.133:610

Mar 5 04:25:51 src@dfw1a-core/10.24.0.123 snort[27872]: MV/IDS362/shellcode-x86-nops-udp: 148.243.136.7:838 -> test.network.70.133:610

Mar 5 04:34:13 src@dfw1a-core/10.24.0.123 snort[27872]: MISC - MISC - id check returned root: test.network.70.133:39168 -> 148.243.136.7:3909

Mar 5 08:09:41 ALID/111 TCP portmap Scan detected: 148.243.136.7:3168 -> mexicocity.network.32.5:111

Mar 5 08:09:41 ALID/111 TCP portmap Scan detected: 148.243.136.7:3170 -> mexicocity.network.32.7:111

Mar 5 08:09:41 ALID/111 TCP portmap Scan detected: 148.243.136.7:3169 -> mexicocity.network.32.6:111

Mar 5 08:09:41 ALID/111 TCP portmap Scan detected: 148.243.136.7:3171 -> mexicocity.network.32.8:111

Mar 5 08:09:41 ALID/111 TCP portmap Scan detected: 148.243.136.7:3174 -> mexicocity.network.32.11:111

Mar 5 08:09:41 ALID/111 TCP portmap Scan detected: 148.243.136.7:3186 -> mexicocity.network.32.23:111

Mar 5 08:09:41 ALID/111 TCP portmap Scan detected: 148.243.136.7:3187 -> mexicocity.network.32.24:111

Mar 5 08:09:41 ALID/111 TCP portmap Scan detected: 148.243.136.7:3188 -> mexicocity.network.32.25:111

Mar 5 08:09:41 ALID/111 TCP portmap Scan detected: 148.243.136.7:3189 -> mexicocity.network.32.26:111

Mar 5 08:09:41 ALID/111 TCP portmap Scan detected: 148.243.136.7:3190 -> mexicocity.network.32.27:111

Mar 5 08:09:41 ALID/111 TCP portmap Scan detected: 148.243.136.7:3191 -> mexicocity.network.32.28:111

Mar 5 08:09:41 ALID/111 TCP portmap Scan detected: 148.243.136.7:3192 -> mexicocity.network.32.29:111

Mar 5 08:09:41 ALID/111 TCP portmap Scan detected: 148.243.136.7:3195 -> mexicocity.network.32.32:111

Mar 5 08:09:41 ALID/111 TCP portmap Scan detected: 148.243.136.7:3195 -> mexicocity.network.32.32:111

Mar 5 08:09:41 ALID/111 TCP portmap Scan detected: 148.243.136.7:3196 -> mexicocity.network.32.33:111

Mar 5 08:09:41 ALID/111 TCP portmap Scan detected: 148.243.136.7:3199 -> mexicocity.network.32.36:111

Mar 5 08:09:41 ALID/111 TCP portmap Scan detected: 148.243.136.7:3201 -> mexicocity.network.32.38:111

Mar 5 08:09:41 ALID/111 TCP portmap Scan detected: 148.243.136.7:3202 -> mexicocity.network.32.39:111

Mar 5 08:09:41 ALID/111 TCP portmap Scan detected: 148.243.136.7:3203 ->

mexicocity.network.32.40:111
Mar 5 08:09:41 ALID/111 TCP portmap Scan detected: 148.243.136.7:3208 ->
mexicocity.network.32.45:111
Mar 5 08:09:41 ALID/111 TCP portmap Scan detected: 148.243.136.7:3209 ->
mexicocity.network.32.46:111
Mar 5 08:09:41 ALID/111 TCP portmap Scan detected: 148.243.136.7:3217 ->
mexicocity.network.32.54:111
Mar 5 08:09:41 ALID/111 TCP portmap Scan detected: 148.243.136.7:3219 ->
mexicocity.network.32.56:111
Mar 5 08:09:41 ALID/111 TCP portmap Scan detected: 148.243.136.7:3221 ->
mexicocity.network.32.58:111
Mar 5 08:09:41 ALID/111 TCP portmap Scan detected: 148.243.136.7:3226 ->
mexicocity.network.32.63:111
Mar 5 08:09:41 ALID/111 TCP portmap Scan detected: 148.243.136.7:3227 ->
mexicocity.network.32.64:111
Mar 5 08:09:41 ALID/111 TCP portmap Scan detected: 148.243.136.7:3228 ->
mexicocity.network.32.65:111
Mar 5 08:09:41 ALID/111 TCP portmap Scan detected: 148.243.136.7:3417 ->
mexicocity.network.32.254:111
Mar 6 00:54:43 ALID/111 TCP portmap Scan detected: 148.243.136.7:4454 ->
mexicocity.network.32.5:111
Mar 6 00:54:43 ALID/111 TCP portmap Scan detected: 148.243.136.7:4456 ->
mexicocity.network.32.7:111
Mar 6 00:54:43 ALID/111 TCP portmap Scan detected: 148.243.136.7:4455 ->
mexicocity.network.32.6:111
Mar 6 00:54:43 ALID/111 TCP portmap Scan detected: 148.243.136.7:4457 ->
mexicocity.network.32.8:111
Mar 6 00:54:43 ALID/111 TCP portmap Scan detected: 148.243.136.7:4460 ->
mexicocity.network.32.11:111
Mar 6 00:54:43 ALID/111 TCP portmap Scan detected: 148.243.136.7:4473 ->
mexicocity.network.32.24:111
Mar 6 00:54:43 ALID/111 TCP portmap Scan detected: 148.243.136.7:4478 ->
mexicocity.network.32.29:111
Mar 6 00:54:43 ALID/111 TCP portmap Scan detected: 148.243.136.7:4489 ->
mexicocity.network.32.40:111
Mar 6 00:54:43 ALID/111 TCP portmap Scan detected: 148.243.136.7:4494 ->
mexicocity.network.32.45:111
Mar 6 00:54:43 ALID/111 TCP portmap Scan detected: 148.243.136.7:4495 ->
mexicocity.network.32.46:111
Mar 6 00:54:43 ALID/111 TCP portmap Scan detected: 148.243.136.7:4504 ->
mexicocity.network.32.54:111
Mar 6 00:54:43 ALID/111 TCP portmap Scan detected: 148.243.136.7:4506 ->
mexicocity.network.32.56:111
Mar 6 00:54:43 ALID/111 TCP portmap Scan detected: 148.243.136.7:4508 ->
mexicocity.network.32.58:111
Mar 6 00:54:43 ALID/111 TCP portmap Scan detected: 148.243.136.7:4513 ->
mexicocity.network.32.63:111
Mar 6 00:54:43 ALID/111 TCP portmap Scan detected: 148.243.136.7:4514 ->
mexicocity.network.32.64:111
Mar 6 00:54:43 ALID/111 TCP portmap Scan detected: 148.243.136.7:4515 ->
mexicocity.network.32.65:111
Mar 6 00:54:43 ALID/111 TCP portmap Scan detected: 148.243.136.7:4799 ->

mexicocity.network.32.254:111
Mar 6 00:54:52 ALID/111 TCP portmap Scan detected: 148.243.136.7:4460 ->
mexicocity.network.32.11:111
Mar 6 00:54:52 ALID/111 TCP portmap Scan detected: 148.243.136.7:4475 ->
mexicocity.network.32.26:111
Mar 6 00:54:52 ALID/111 TCP portmap Scan detected: 148.243.136.7:4476 ->
mexicocity.network.32.27:111
Mar 6 00:54:52 ALID/111 TCP portmap Scan detected: 148.243.136.7:4477 ->
mexicocity.network.32.28:111
Mar 6 00:54:52 ALID/111 TCP portmap Scan detected: 148.243.136.7:4482 ->
mexicocity.network.32.33:111
Mar 6 00:54:52 ALID/111 TCP portmap Scan detected: 148.243.136.7:4472 ->
mexicocity.network.32.23:111
Mar 6 00:54:52 ALID/111 TCP portmap Scan detected: 148.243.136.7:4474 ->
mexicocity.network.32.25:111
Mar 7 00:22:28 ALID/111 TCP portmap Scan detected: 148.243.136.7:2320 ->
tokyo.network.105.5:111
Mar 7 00:22:28 ALID/111 TCP portmap Scan detected: 148.243.136.7:2322 ->
tokyo.network.105.7:111
Mar 7 00:22:28 ALID/111 TCP portmap Scan detected: 148.243.136.7:2326 ->
tokyo.network.105.11:111
Mar 7 00:22:28 ALID/111 TCP portmap Scan detected: 148.243.136.7:2323 ->
tokyo.network.105.8:111
Mar 7 00:22:28 ALID/111 TCP portmap Scan detected: 148.243.136.7:2358 ->
tokyo.network.105.39:111
Mar 7 00:22:28 ALID/111 TCP portmap Scan detected: 148.243.136.7:2321 ->
tokyo.network.105.6:111
Mar 7 00:22:28 ALID/111 TCP portmap Scan detected: 148.243.136.7:2346 ->
tokyo.network.105.27:111
Mar 7 00:22:28 ALID/111 TCP portmap Scan detected: 148.243.136.7:2353 ->
tokyo.network.105.34:111
Mar 7 00:22:28 ALID/111 TCP portmap Scan detected: 148.243.136.7:2347 ->
tokyo.network.105.28:111
Mar 7 00:22:28 ALID/111 TCP portmap Scan detected: 148.243.136.7:2349 ->
tokyo.network.105.30:111
Mar 7 00:22:28 ALID/111 TCP portmap Scan detected: 148.243.136.7:2375 ->
tokyo.network.105.56:111
Mar 7 00:22:28 ALID/111 TCP portmap Scan detected: 148.243.136.7:2341 ->
tokyo.network.105.23:111
Mar 7 00:22:28 ALID/111 TCP portmap Scan detected: 148.243.136.7:2344 ->
tokyo.network.105.25:111
Mar 7 00:22:28 ALID/111 TCP portmap Scan detected: 148.243.136.7:2343 ->
tokyo.network.105.24:111
Mar 7 00:22:28 ALID/111 TCP portmap Scan detected: 148.243.136.7:2348 ->
tokyo.network.105.29:111
Mar 7 00:22:28 ALID/111 TCP portmap Scan detected: 148.243.136.7:2345 ->
tokyo.network.105.26:111
Mar 7 00:22:28 ALID/111 TCP portmap Scan detected: 148.243.136.7:2383 ->
tokyo.network.105.64:111
Mar 7 00:22:28 ALID/111 TCP portmap Scan detected: 148.243.136.7:2373 ->
tokyo.network.105.54:111
Mar 7 00:22:28 ALID/111 TCP portmap Scan detected: 148.243.136.7:2377 ->

tokyo.network.105.58:111
Mar 7 00:22:28 ALID/111 TCP portmap Scan detected: 148.243.136.7:2382 ->
tokyo.network.105.63:111
Mar 7 00:22:28 ALID/111 TCP portmap Scan detected: 148.243.136.7:2384 ->
tokyo.network.105.65:111
Mar 11 22:17:26 ALID/111 TCP portmap Scan detected: 148.243.136.7:1559 ->
tokyo.network2.242.134:111
Mar 11 22:17:26 ALID/111 TCP portmap Scan detected: 148.243.136.7:1575 ->
tokyo.network2.242.150:111
Mar 11 22:17:26 ALID/111 TCP portmap Scan detected: 148.243.136.7:1564 ->
tokyo.network2.242.139:111
Mar 11 22:17:26 ALID/111 TCP portmap Scan detected: 148.243.136.7:1570 ->
tokyo.network2.242.145:111
Mar 11 22:17:26 ALID/111 TCP portmap Scan detected: 148.243.136.7:1586 ->
tokyo.network2.242.161:111
Mar 12 05:46:25 ALID/111 TCP portmap Scan detected: 148.243.136.7:2969 ->
tokyo.network2.242.147:111
Mar 12 05:46:25 ALID/111 TCP portmap Scan detected: 148.243.136.7:3003 ->
tokyo.network2.242.181:111
Mar 12 05:46:25 ALID/111 TCP portmap Scan detected: 148.243.136.7:3064 ->
tokyo.network2.242.242:111
Mar 12 05:46:25 ALID/111 TCP portmap Scan detected: 148.243.136.7:2960 ->
tokyo.network2.242.138:111

Detect 1 - Forensics Performed On Host

=====

Name: na-148-243-131-7.na.avantel.net.mx
Address: 148.243.136.7

#####

[~]\$ whois 148.243.136.7
nsINIC-Mexico (NETBLK-REDMEX-BNETS)REDMEX-BNETS 148.203.0.0 - 148.250.255.255
Avantel, S.A. (NETBLK-AVANTEL-BL11) AVANTEL-BL11 148.243.0.0 - 148.243.255.255
CAPITEL SYSTEMS SA DE CV (NETBLK-CAPSYS-MX) CAPSYS-MX
148.243.136.0 - 148.243.136.127

#####

na-148-243-131-7.na.avantel.net.mx
NIC-Mexico (NETBLK-REDMEX-BNETS)
Av. Eugenio Garza Sada #2501 Sur
Monterrey, Nuevo Leon 64849
MX

Netname: REDMEX-BNETS
Netblock: 148.203.0.0 - 148.250.255.255

Maintainer: MEX

Coordinator:

Mexico, Administrador Ip (AIM4-ARIN) ipmaster@NIC.MX
+52(8) 3875346 (FAX) (8) 3284208

Record last updated on 18-May-1999.

Database last updated on 5-Mar-2001 06:33:54 EDT.

#####

Avantel, S.A. (NETBLK-AVANTEL-BL11)
Vasconcelos 130 ote
San Pedro, Nuevo Leon 66267
MX

Netname: AVANTEL-BL11

Netblock: 148.243.0.0 - 148.243.255.255

Maintainer: AVAN

Coordinator:

Administrator, Noc (NA83-ARIN) noc@AVANTEL.NET.MX
(8) 156 3065

Domain System inverse mapping provided by:

DNS1.AVANTEL.NET.MX 200.33.213.66

DNS2.AVANTEL.NET.MX 200.33.209.66

Record last updated on 23-Sep-1999.

Database last updated on 5-Mar-2001 06:33:54 EDT.

#####

CAPITEL SYSTEMS SA DE CV (NETBLK-CAPSYS-MX)
SOR JUANA INES DE LA CRUZ 400 NTE
TAMPICO, TM 89000
Mexico

Netname: CAPSYS-MX

Netblock: 148.243.136.0 - 148.243.136.127

Coordinator:

RODRIGUEZ, ADOLFO (AR324-ARIN) arlatt@hotmail.com
(1) 2186416

Record last updated on 04-May-2000.

Database last updated on 5-Mar-2001 06:33:54 EDT.

=====

###border###

last reveals that /var/log/wtmp was apparently not tampered with.

```
reboot system boot 2.2.14-5.0smp Mon Mar 5 12:46 (1+06:28) =
=20
sdr pts/1 200.51.210.75 Mon Mar 5 06:20 - 06:20 (00:00) =
=20
sdr pts/0 200.51.210.75 Mon Mar 5 06:16 - 06:20 (00:04) =
=20
sdr pts/0 200.51.210.75 Mon Mar 5 03:32 - 03:32 (00:00) =
=20
root tty2 Thu Mar 1 08:42 - 09:56 (4+01:13) =
=20
```

wtmp begins Thu Mar 1 08:42:54 2001

###border###

/var/log/messages shows the initial buffer overflow of rpc statd. At this point the intruder had root privilege on the system.

```
Mar 5 03:21:54 vpn1 rpc.statd[433]: gethostbyname error for ^X=F7=FF=BF^X=
=F7=FF=BF^Y=F7=FF=BF^Y=F7=FF=BF^Z=F7=FF=BF^Z=F7=FF=BF^ [=F7=FF=BF^ [=F7=FF=BF^ [=F7=FF=BF^
bffff750 8049710 8052c18687465676274736f6d616e797265206520726f7220726f66 =
=
=
=
bffff718 =
bffff719 bff=
ff71a =
=
bffff71b=90=90=90=90=90=90=90=90=90=90=
=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=
=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=
=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=
=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=
=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=
=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=
=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=
=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=
=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=
=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=
=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=90=
Mar 5 03:32:04 vpn1 PAM_pwdb[11238]: password for (sdr/503) changed by ((n=
ull)/0)
Mar 5 03:32:35 vpn1 PAM_pwdb[11240]: (login) session opened for user sdr b=
y (uid=3D0)
Mar 5 03:32:42 vpn1 inetd[763]: pid 11239: exit status 1
Mar 5 04:02:00 vpn1 anacron[11276]: Updated timestamp for job `cron.daily`=
to 2001-03-05
Mar 5 04:02:05 vpn1 PAM_pwdb[11389]: (su) session opened for user news by =
```

```
(uid=3D0)
Mar 5 04:02:05 vpn1 PAM_pwdb[11389]: (su) session closed for user news
Mar 5 06:16:01 vpn1 PAM_pwdb[11473]: (login) session opened for user sdr b=
y (uid=3D0)
Mar 5 06:20:01 vpn1 PAM_pwdb[14152]: (login) session opened for user sdr b=
y (uid=3D0)
Mar 5 06:20:12 vpn1 PAM_pwdb[14152]: (login) session closed for user sdr
Mar 5 06:20:12 vpn1 inetd[763]: pid 14151: exit status 1
Mar 5 06:20:16 vpn1 PAM_pwdb[11473]: (login) session closed for user sdr
Mar 5 06:20:16 vpn1 inetd[763]: pid 11472: exit status 1
```

###border###

The intruder then added a regular user account to /etc/passwd. I estimate = that /etc/passwd- shows what the /etc/passwd file temporarily looked like. = It is important to note that the new user sdr's home directory is /tmp. T= he reason for this will become clear later.

```
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:
daemon:x:2:2:daemon:/sbin:
adm:x:3:4:adm:/var/adm:
lp:x:4:7:lp:/var/spool/lpd:
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:
news:x:9:13:news:/var/spool/news:
uucp:x:10:14:uucp:/var/spool/uucp:
operator:x:11:0:operator:/root:
games:x:12:100:games:/usr/games:
gopher:x:13:30:gopher:/usr/lib/gopher-data:
ftp:x:14:50:FTP User:/home/ftp:
nobody:x:99:99:Nobody:/:
xfs:x:43:43:X Font Server:/etc/X11/fs:/bin/false
named:x:25:25:Named:/var/named:/bin/false
gdm:x:42:42:./home/gdm:/bin/bash
piranha:x:60:60:./home/httpd/html/piranha:/dev/null
postgres:x:26:26:PostgreSQL Server:/var/lib/pgsql:/bin/bash
pvm:x:24:24:./usr/share/pvm3:/bin/bash
squid:x:23:23:./var/spool/squid:/dev/null
sdr:x:503:503:./tmp:/bin/bash
```

###border###

The intruder then altered the passwd file to obscure the new account by placing it in the middle of the passwd file. Also, the sdr user's uid and gid are changed, home directory is changed to more closely emulate the home directory of other system daemons, and finally the shell is altered to /sbin/false. The shell change is apparently to suggest that this user can't login, as it has no shell.

```
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:
daemon:x:2:2:daemon:/sbin:
adm:x:3:4:adm:/var/adm:
lp:x:4:7:lp:/var/spool/lpd:
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:
sdr:x:20:20:sdr:/var/spool/sdr:/sbin/false
news:x:9:13:news:/var/spool/news:
uucp:x:10:14:uucp:/var/spool/uucp:
operator:x:11:0:operator:/root:
games:x:12:100:games:/usr/games:
gopher:x:13:30:gopher:/usr/lib/gopher-data:
ftp:x:14:50:FTP User:/home/ftp:
nobody:x:99:99:Nobody:/:
xfs:x:43:43:X Font Server:/etc/X11/fs:/bin/false
named:x:25:25:Named:/var/named:/bin/false
gdm:x:42:42:./home/gdm:/bin/bash
piranha:x:60:60:./home/httpd/html/piranha:/dev/null
postgres:x:26:26:PostgreSQL Server:/var/lib/pgsql:/bin/bash
pvm:x:24:24:./usr/share/pvm3:/bin/bash
squid:x:23:23:./var/spool/squid:/dev/null
```

###border###

Having recalled that the user sdr's home directory was initially /tmp, we examine /tmp/.bash_history. It is becoming clear that either the intruder did not have time, interest, or ability to hide his/her tracks very well. It struck me as odd the attention to detail given to the /etc/passwd file, while leaving other obvious log and other accounting information was not sanitized. This .bash_history file is very revealing. It shows an interesting event in that a file called 'sush' (something like superuser shell?) is copied to /sbin/false. Also the 'cat > tmp' is apparently where he/she entered a script from stdin. This file no longer exists, so some of the intruder's actions are lost. While a full forensic effort might have revealed the data still on the drive platters, grave robbing such as this could have occupied days - weeks. Also since we initially decided to not remove the hard drives, it is likely that we have altered the disks sufficiently to reduce our ability to retrieve this information. Given the criticality and limited damage, the value of this host and therefore its forensic/evidentiary value, I don't believe we've sacrificed any useful data.

```
w
ls
uname -a
cp sush /sbin/false
pico /etc/passwd
pico /etc/passwd
pico /etc/passwd
pico /etc/shadow
ls
rm *
exit
```

```
w
cd /var/tmp
ls
ls -al
cd /tmp
ls -al
cd /var/tmp
ls
cat > tmp
chmod +x tmp
=2E/tmp
exit
```

###border###

Running 'strings' on the binary /sbin/false file reveals the following. While not complete information, it certainly does appear to turn over a rootshell to sdr when he/she logs in.

```
/lib/ld-linux.so.2
__gmon_start__
libc.so.6
system
__deregister_frame_info
setgid
_IO_stdin_used
__libc_start_main
setuid
__register_frame_info
GLIBC_2.0
PTRh
QVh8
/bin/bash
```

###border###

A comparison of the compromised system's md5sums and file stat information to that of a new full RedHat 6.2 install reveal changes in keeping with what activity we have witnessed thus far. While evidence of backdoors/trojans/rootkits is limited, this box is still compromised and no longer trustworthy. It will be wiped clean and re-installed by XXXXXXXXXX with XXXXXXXXXXXXX's new internal Linux distribution (which specifically designed for our company's needs, with security in mind). This will give XXXXXXXXXX a strong base image from which to build. Note that the integrity check results below were collected trusting only PERL on the compromised system. The md5 implementation was completely self contained in the forensic perl script. (Special thanks go to Dan Farmer for posting his EMT script to the TCT mailing list. This was the tool used to do this quick and dirty file integrity test).

The following SxID files are new
/sbin/false

The following Executable files changed from the trusted dist:

/usr/doc/libtool-1.3.4/demo/configure

The following non-executable files changed from the trusted dist:

/boot/kernel.h
/etc/X11/fs/config
/etc/conf.linuxconf
/etc/group
/etc/httpd/conf/access.conf
/etc/httpd/conf/httpd.conf
/etc/httpd/php3.ini
/etc/inetd.conf
/etc/info-dir
/etc/localtime
/etc/mime.types
/etc/pam.d/login
/etc/pam.d/passwd
/etc/pam.d/rlogin
/etc/passwd
/etc/services
/etc/sysconfig/pcmcia
/etc/sysctl.conf
/etc/syslog.conf
/lib/modules/2.2.14-5.0/block/DAC960.o
/lib/modules/2.2.14-5.0/block/cpqarray.o
/lib/modules/2.2.14-5.0/block/ide-floppy.o
/lib/modules/2.2.14-5.0/block/ide-tape.o
/lib/modules/2.2.14-5.0/block/linear.o
/lib/modules/2.2.14-5.0/block/loop.o
/lib/modules/2.2.14-5.0/block/nbd.o
/lib/modules/2.2.14-5.0/block/raid0.o
/lib/modules/2.2.14-5.0/block/raid1.o
/lib/modules/2.2.14-5.0/block/raid5.o
/lib/modules/2.2.14-5.0/block/xd.o
/lib/modules/2.2.14-5.0/cdrom/aztcd.o
/lib/modules/2.2.14-5.0/cdrom/cdu31a.o
/lib/modules/2.2.14-5.0/cdrom/cm206.o
/lib/modules/2.2.14-5.0/cdrom/gscd.o
/lib/modules/2.2.14-5.0/cdrom/isp16.o
/lib/modules/2.2.14-5.0/cdrom/mcd.o
/lib/modules/2.2.14-5.0/cdrom/mcdx.o
/lib/modules/2.2.14-5.0/cdrom/optcd.o
/lib/modules/2.2.14-5.0/cdrom/sbpcd.o
/lib/modules/2.2.14-5.0/cdrom/sjcd.o
/lib/modules/2.2.14-5.0/cdrom/sonycd535.o
/lib/modules/2.2.14-5.0/fs/autofs.o
/lib/modules/2.2.14-5.0/fs/binfmt_aout.o
/lib/modules/2.2.14-5.0/fs/binfmt_java.o
/lib/modules/2.2.14-5.0/fs/binfmt_misc.o
/lib/modules/2.2.14-5.0/fs/coda.o
/lib/modules/2.2.14-5.0/fs/fat.o
/lib/modules/2.2.14-5.0/fs/hfs.o

/lib/modules/2.2.14-5.0/fs/hpfs.o
/lib/modules/2.2.14-5.0/fs/lockd.o
/lib/modules/2.2.14-5.0/fs/minix.o
/lib/modules/2.2.14-5.0/fs/msdos.o
/lib/modules/2.2.14-5.0/fs/ncpfs.o
/lib/modules/2.2.14-5.0/fs/nfs.o
/lib/modules/2.2.14-5.0/fs/nfsd.o
/lib/modules/2.2.14-5.0/fs/nls_cp437.o
/lib/modules/2.2.14-5.0/fs/nls_cp737.o
/lib/modules/2.2.14-5.0/fs/nls_cp775.o
/lib/modules/2.2.14-5.0/fs/nls_cp850.o
/lib/modules/2.2.14-5.0/fs/nls_cp852.o
/lib/modules/2.2.14-5.0/fs/nls_cp855.o
/lib/modules/2.2.14-5.0/fs/nls_cp857.o
/lib/modules/2.2.14-5.0/fs/nls_cp860.o
/lib/modules/2.2.14-5.0/fs/nls_cp861.o
/lib/modules/2.2.14-5.0/fs/nls_cp862.o
/lib/modules/2.2.14-5.0/fs/nls_cp863.o
/lib/modules/2.2.14-5.0/fs/nls_cp864.o
/lib/modules/2.2.14-5.0/fs/nls_cp865.o
/lib/modules/2.2.14-5.0/fs/nls_cp866.o
/lib/modules/2.2.14-5.0/fs/nls_cp869.o
/lib/modules/2.2.14-5.0/fs/nls_cp874.o
/lib/modules/2.2.14-5.0/fs/nls_iso8859-1.o
/lib/modules/2.2.14-5.0/fs/nls_iso8859-14.o
/lib/modules/2.2.14-5.0/fs/nls_iso8859-15.o
/lib/modules/2.2.14-5.0/fs/nls_iso8859-2.o
/lib/modules/2.2.14-5.0/fs/nls_iso8859-3.o
/lib/modules/2.2.14-5.0/fs/nls_iso8859-4.o
/lib/modules/2.2.14-5.0/fs/nls_iso8859-5.o
/lib/modules/2.2.14-5.0/fs/nls_iso8859-6.o
/lib/modules/2.2.14-5.0/fs/nls_iso8859-7.o
/lib/modules/2.2.14-5.0/fs/nls_iso8859-8.o
/lib/modules/2.2.14-5.0/fs/nls_iso8859-9.o
/lib/modules/2.2.14-5.0/fs/nls_koi8-r.o
/lib/modules/2.2.14-5.0/fs/romfs.o
/lib/modules/2.2.14-5.0/fs/smbfs.o
/lib/modules/2.2.14-5.0/fs/sysv.o
/lib/modules/2.2.14-5.0/fs/ufs.o
/lib/modules/2.2.14-5.0/fs/umsdos.o
/lib/modules/2.2.14-5.0/fs/vfat.o
/lib/modules/2.2.14-5.0/ipv4/ip_masq_autofw.o
/lib/modules/2.2.14-5.0/ipv4/ip_masq_cuseeme.o
/lib/modules/2.2.14-5.0/ipv4/ip_masq_ftp.o
/lib/modules/2.2.14-5.0/ipv4/ip_masq_irc.o
/lib/modules/2.2.14-5.0/ipv4/ip_masq_mfw.o
/lib/modules/2.2.14-5.0/ipv4/ip_masq_portfw.o
/lib/modules/2.2.14-5.0/ipv4/ip_masq_quake.o
/lib/modules/2.2.14-5.0/ipv4/ip_masq_raudio.o
/lib/modules/2.2.14-5.0/ipv4/ip_masq_user.o
/lib/modules/2.2.14-5.0/ipv4/ip_masq_vdolive.o
/lib/modules/2.2.14-5.0/ipv4/ip_vs_lc.o

/lib/modules/2.2.14-5.0/ipv4/ip_vs_rr.o
/lib/modules/2.2.14-5.0/ipv4/ip_vs_wlc.o
/lib/modules/2.2.14-5.0/ipv4/ip_vs_wrr.o
/lib/modules/2.2.14-5.0/ipv4/rarp.o
/lib/modules/2.2.14-5.0/misc/aci.o
/lib/modules/2.2.14-5.0/misc/acquirewdt.o
/lib/modules/2.2.14-5.0/misc/actisys.o
/lib/modules/2.2.14-5.0/misc/ad1816.o
/lib/modules/2.2.14-5.0/misc/ad1848.o
/lib/modules/2.2.14-5.0/misc/adlib_card.o
/lib/modules/2.2.14-5.0/misc/aedsp16.o
/lib/modules/2.2.14-5.0/misc/aten.o
/lib/modules/2.2.14-5.0/misc/atixlmouse.o
/lib/modules/2.2.14-5.0/misc/awe_wave.o
/lib/modules/2.2.14-5.0/misc/b1.o
/lib/modules/2.2.14-5.0/misc/b1isa.o
/lib/modules/2.2.14-5.0/misc/b1pci.o
/lib/modules/2.2.14-5.0/misc/bpck.o
/lib/modules/2.2.14-5.0/misc/bttv.o
/lib/modules/2.2.14-5.0/misc/busmouse.o
/lib/modules/2.2.14-5.0/misc/buz.o
/lib/modules/2.2.14-5.0/misc/bw-qcam.o
/lib/modules/2.2.14-5.0/misc/c-qcam.o
/lib/modules/2.2.14-5.0/misc/capi.o
/lib/modules/2.2.14-5.0/misc/capidrv.o
/lib/modules/2.2.14-5.0/misc/capiutil.o
/lib/modules/2.2.14-5.0/misc/cmpci.o
/lib/modules/2.2.14-5.0/misc/comm.o
/lib/modules/2.2.14-5.0/misc/cs4232.o
/lib/modules/2.2.14-5.0/misc/cyclades.o
/lib/modules/2.2.14-5.0/misc/dstr.o
/lib/modules/2.2.14-5.0/misc/dtlk.o
/lib/modules/2.2.14-5.0/misc/emu10k1.o
/lib/modules/2.2.14-5.0/misc/epat.o
/lib/modules/2.2.14-5.0/misc/epca.o
/lib/modules/2.2.14-5.0/misc/epia.o
/lib/modules/2.2.14-5.0/misc/es1370.o
/lib/modules/2.2.14-5.0/misc/es1371.o
/lib/modules/2.2.14-5.0/misc/esi.o
/lib/modules/2.2.14-5.0/misc/esp.o
/lib/modules/2.2.14-5.0/misc/esssolo1.o
/lib/modules/2.2.14-5.0/misc/fit2.o
/lib/modules/2.2.14-5.0/misc/fit3.o
/lib/modules/2.2.14-5.0/misc/friq.o
/lib/modules/2.2.14-5.0/misc/frpw.o
/lib/modules/2.2.14-5.0/misc/ftape.o
/lib/modules/2.2.14-5.0/misc/girbil.o
/lib/modules/2.2.14-5.0/misc/gus.o
/lib/modules/2.2.14-5.0/misc/hisax.o
/lib/modules/2.2.14-5.0/misc/i2c.o
/lib/modules/2.2.14-5.0/misc/icn.o
/lib/modules/2.2.14-5.0/misc/ip2.o

/lib/modules/2.2.14-5.0/misc/ip2main.o
/lib/modules/2.2.14-5.0/misc/iph5526.o
/lib/modules/2.2.14-5.0/misc/ircomm_tty.o
/lib/modules/2.2.14-5.0/misc/irlpt.o
/lib/modules/2.2.14-5.0/misc/irlpt_client.o
/lib/modules/2.2.14-5.0/misc/irlpt_server.o
/lib/modules/2.2.14-5.0/misc/irport.o
/lib/modules/2.2.14-5.0/misc/irtty.o
/lib/modules/2.2.14-5.0/misc/isdn.o
/lib/modules/2.2.14-5.0/misc/isdn_bsdcomp.o
/lib/modules/2.2.14-5.0/misc/isdnloop.o
/lib/modules/2.2.14-5.0/misc/isicom.o
/lib/modules/2.2.14-5.0/misc/istallion.o
/lib/modules/2.2.14-5.0/misc/ixj.o
/lib/modules/2.2.14-5.0/misc/joy-analog.o
/lib/modules/2.2.14-5.0/misc/joy-assassin.o
/lib/modules/2.2.14-5.0/misc/joy-console.o
/lib/modules/2.2.14-5.0/misc/joy-creative.o
/lib/modules/2.2.14-5.0/misc/joy-db9.o
/lib/modules/2.2.14-5.0/misc/joy-gravis.o
/lib/modules/2.2.14-5.0/misc/joy-lightning.o
/lib/modules/2.2.14-5.0/misc/joy-logitech.o
/lib/modules/2.2.14-5.0/misc/joy-magellan.o
/lib/modules/2.2.14-5.0/misc/joy-pci.o
/lib/modules/2.2.14-5.0/misc/joy-sidewinder.o
/lib/modules/2.2.14-5.0/misc/joy-spaceball.o
/lib/modules/2.2.14-5.0/misc/joy-spaceorb.o
/lib/modules/2.2.14-5.0/misc/joy-thrustmaster.o
/lib/modules/2.2.14-5.0/misc/joy-turbografx.o
/lib/modules/2.2.14-5.0/misc/joy-warrior.o
/lib/modules/2.2.14-5.0/misc/joystick.o
/lib/modules/2.2.14-5.0/misc/kbic.o
/lib/modules/2.2.14-5.0/misc/kernelcapi.o
/lib/modules/2.2.14-5.0/misc/ktti.o
/lib/modules/2.2.14-5.0/misc/litelink.o
/lib/modules/2.2.14-5.0/misc/lp.o
/lib/modules/2.2.14-5.0/misc/mad16.o
/lib/modules/2.2.14-5.0/misc/maestro.o
/lib/modules/2.2.14-5.0/misc/maui.o
/lib/modules/2.2.14-5.0/misc/moxa.o
/lib/modules/2.2.14-5.0/misc/mpu401.o
/lib/modules/2.2.14-5.0/misc/msbusmouse.o
/lib/modules/2.2.14-5.0/misc/msnd.o
/lib/modules/2.2.14-5.0/misc/msnd_classic.o
/lib/modules/2.2.14-5.0/misc/msnd_pinnacle.o
/lib/modules/2.2.14-5.0/misc/msp3400.o
/lib/modules/2.2.14-5.0/misc/mxser.o
/lib/modules/2.2.14-5.0/misc/n_hdlc.o
/lib/modules/2.2.14-5.0/misc/nm256.o
/lib/modules/2.2.14-5.0/misc/nvram.o
/lib/modules/2.2.14-5.0/misc/on20.o
/lib/modules/2.2.14-5.0/misc/on26.o

/lib/modules/2.2.14-5.0/misc/opl3.o
/lib/modules/2.2.14-5.0/misc/opl3sa.o
/lib/modules/2.2.14-5.0/misc/opl3sa2.o
/lib/modules/2.2.14-5.0/misc/paride.o
/lib/modules/2.2.14-5.0/misc/parport.o
/lib/modules/2.2.14-5.0/misc/parport_pc.o
/lib/modules/2.2.14-5.0/misc/parport_probe.o
/lib/modules/2.2.14-5.0/misc/pas2.o
/lib/modules/2.2.14-5.0/misc/pc110pad.o
/lib/modules/2.2.14-5.0/misc/pc87108.o
/lib/modules/2.2.14-5.0/misc/pcbit.o
/lib/modules/2.2.14-5.0/misc/pcd.o
/lib/modules/2.2.14-5.0/misc/pcwd.o
/lib/modules/2.2.14-5.0/misc/pd.o
/lib/modules/2.2.14-5.0/misc/pf.o
/lib/modules/2.2.14-5.0/misc/pg.o
/lib/modules/2.2.14-5.0/misc/phonedev.o
/lib/modules/2.2.14-5.0/misc/pms.o
/lib/modules/2.2.14-5.0/misc/pss.o
/lib/modules/2.2.14-5.0/misc/pt.o
/lib/modules/2.2.14-5.0/misc/qpmouse.o
/lib/modules/2.2.14-5.0/misc/radio-aimslab.o
/lib/modules/2.2.14-5.0/misc/radio-aztech.o
/lib/modules/2.2.14-5.0/misc/radio-cadet.o
/lib/modules/2.2.14-5.0/misc/radio-gemtek.o
/lib/modules/2.2.14-5.0/misc/radio-miropcm20.o
/lib/modules/2.2.14-5.0/misc/radio-rtrack2.o
/lib/modules/2.2.14-5.0/misc/radio-sf16fmi.o
/lib/modules/2.2.14-5.0/misc/radio-trust.o
/lib/modules/2.2.14-5.0/misc/radio-typhoon.o
/lib/modules/2.2.14-5.0/misc/radio-zoltrix.o
/lib/modules/2.2.14-5.0/misc/riscom8.o
/lib/modules/2.2.14-5.0/misc/rocket.o
/lib/modules/2.2.14-5.0/misc/saa5249.o
/lib/modules/2.2.14-5.0/misc/saa7111.o
/lib/modules/2.2.14-5.0/misc/saa7185.o
/lib/modules/2.2.14-5.0/misc/sb.o
/lib/modules/2.2.14-5.0/misc/sgalaxy.o
/lib/modules/2.2.14-5.0/misc/smc-ircc.o
/lib/modules/2.2.14-5.0/misc/softdog.o
/lib/modules/2.2.14-5.0/misc/softoss2.o
/lib/modules/2.2.14-5.0/misc/sonicvibes.o
/lib/modules/2.2.14-5.0/misc/sound.o
/lib/modules/2.2.14-5.0/misc/soundcore.o
/lib/modules/2.2.14-5.0/misc/soundlow.o
/lib/modules/2.2.14-5.0/misc/specialix.o
/lib/modules/2.2.14-5.0/misc/sscape.o
/lib/modules/2.2.14-5.0/misc/stallion.o
/lib/modules/2.2.14-5.0/misc/sunrpc.o
/lib/modules/2.2.14-5.0/misc/sx.o
/lib/modules/2.2.14-5.0/misc/synclink.o
/lib/modules/2.2.14-5.0/misc/t1isa.o

/lib/modules/2.2.14-5.0/misc/t1pci.o
/lib/modules/2.2.14-5.0/misc/tekram.o
/lib/modules/2.2.14-5.0/misc/toshoboe.o
/lib/modules/2.2.14-5.0/misc/trix.o
/lib/modules/2.2.14-5.0/misc/tuner.o
/lib/modules/2.2.14-5.0/misc/uart401.o
/lib/modules/2.2.14-5.0/misc/uircc.o
/lib/modules/2.2.14-5.0/misc/v_midi.o
/lib/modules/2.2.14-5.0/misc/via82cxxx.o
/lib/modules/2.2.14-5.0/misc/videodev.o
/lib/modules/2.2.14-5.0/misc/w83977af_ir.o
/lib/modules/2.2.14-5.0/misc/wanrouter.o
/lib/modules/2.2.14-5.0/misc/wavefront.o
/lib/modules/2.2.14-5.0/misc/wdt.o
/lib/modules/2.2.14-5.0/misc/zft-compressor.o
/lib/modules/2.2.14-5.0/misc/zftape.o
/lib/modules/2.2.14-5.0/net/3c501.o
/lib/modules/2.2.14-5.0/net/3c503.o
/lib/modules/2.2.14-5.0/net/3c505.o
/lib/modules/2.2.14-5.0/net/3c507.o
/lib/modules/2.2.14-5.0/net/3c509.o
/lib/modules/2.2.14-5.0/net/3c515.o
/lib/modules/2.2.14-5.0/net/3c59x.o
/lib/modules/2.2.14-5.0/net/3c90x.o
/lib/modules/2.2.14-5.0/net/82596.o
/lib/modules/2.2.14-5.0/net/8390.o
/lib/modules/2.2.14-5.0/net/ac3200.o
/lib/modules/2.2.14-5.0/net/acenic.o
/lib/modules/2.2.14-5.0/net/arian-proc.o
/lib/modules/2.2.14-5.0/net/arian.o
/lib/modules/2.2.14-5.0/net/at1700.o
/lib/modules/2.2.14-5.0/net/bonding.o
/lib/modules/2.2.14-5.0/net/bsd_comp.o
/lib/modules/2.2.14-5.0/net/cosa.o
/lib/modules/2.2.14-5.0/net/cs89x0.o
/lib/modules/2.2.14-5.0/net/de4x5.o
/lib/modules/2.2.14-5.0/net/de600.o
/lib/modules/2.2.14-5.0/net/de620.o
/lib/modules/2.2.14-5.0/net/depca.o
/lib/modules/2.2.14-5.0/net/dgrs.o
/lib/modules/2.2.14-5.0/net/dlci.o
/lib/modules/2.2.14-5.0/net/dmfe.o
/lib/modules/2.2.14-5.0/net/dummy.o
/lib/modules/2.2.14-5.0/net/e2100.o
/lib/modules/2.2.14-5.0/net/eeopro.o
/lib/modules/2.2.14-5.0/net/eeopro100.o
/lib/modules/2.2.14-5.0/net/eexpress.o
/lib/modules/2.2.14-5.0/net/epic100.o
/lib/modules/2.2.14-5.0/net/eql.o
/lib/modules/2.2.14-5.0/net/es3210.o
/lib/modules/2.2.14-5.0/net/eth16i.o
/lib/modules/2.2.14-5.0/net/ethertap.o

/lib/modules/2.2.14-5.0/net/ewrk3.o
/lib/modules/2.2.14-5.0/net/fmv18x.o
/lib/modules/2.2.14-5.0/net/hostess_sv11.o
/lib/modules/2.2.14-5.0/net/hp-plus.o
/lib/modules/2.2.14-5.0/net/hp.o
/lib/modules/2.2.14-5.0/net/hp100.o
/lib/modules/2.2.14-5.0/net/ibmtr.o
/lib/modules/2.2.14-5.0/net/ircomm.o
/lib/modules/2.2.14-5.0/net/irda.o
/lib/modules/2.2.14-5.0/net/irda_deflate.o
/lib/modules/2.2.14-5.0/net/irlan.o
/lib/modules/2.2.14-5.0/net/lance.o
/lib/modules/2.2.14-5.0/net/lne390.o
/lib/modules/2.2.14-5.0/net/ne.o
/lib/modules/2.2.14-5.0/net/ne2k-pci.o
/lib/modules/2.2.14-5.0/net/ne3210.o
/lib/modules/2.2.14-5.0/net/ni5010.o
/lib/modules/2.2.14-5.0/net/ni52.o
/lib/modules/2.2.14-5.0/net/ni65.o
/lib/modules/2.2.14-5.0/net/old_tulip.o
/lib/modules/2.2.14-5.0/net/olympic.o
/lib/modules/2.2.14-5.0/net/pcnet32.o
/lib/modules/2.2.14-5.0/net/plip.o
/lib/modules/2.2.14-5.0/net/ppp.o
/lib/modules/2.2.14-5.0/net/ppp_deflate.o
/lib/modules/2.2.14-5.0/net/rcpci.o
/lib/modules/2.2.14-5.0/net/rtl8139.o
/lib/modules/2.2.14-5.0/net/sb1000.o
/lib/modules/2.2.14-5.0/net/sbni.o
/lib/modules/2.2.14-5.0/net/sdla.o
/lib/modules/2.2.14-5.0/net/sdladv.o
/lib/modules/2.2.14-5.0/net/sealevel.o
/lib/modules/2.2.14-5.0/net/shaper.o
/lib/modules/2.2.14-5.0/net/sis900.o
/lib/modules/2.2.14-5.0/net/sk98lin.o
/lib/modules/2.2.14-5.0/net/sktr.o
/lib/modules/2.2.14-5.0/net/slhc.o
/lib/modules/2.2.14-5.0/net/slip.o
/lib/modules/2.2.14-5.0/net/smc-ultra.o
/lib/modules/2.2.14-5.0/net/smc-ultra32.o
/lib/modules/2.2.14-5.0/net/smc9194.o
/lib/modules/2.2.14-5.0/net/strip.o
/lib/modules/2.2.14-5.0/net/syncppp.o
/lib/modules/2.2.14-5.0/net/tlan.o
/lib/modules/2.2.14-5.0/net/tulip.o
/lib/modules/2.2.14-5.0/net/via-rhine.o
/lib/modules/2.2.14-5.0/net/wanpipe.o
/lib/modules/2.2.14-5.0/net/wavelan.o
/lib/modules/2.2.14-5.0/net/wd.o
/lib/modules/2.2.14-5.0/net/yellowfin.o
/lib/modules/2.2.14-5.0/net/z85230.o
/lib/modules/2.2.14-5.0/scsi/53c7,8xx.o

/lib/modules/2.2.14-5.0/scsi/AM53C974.o
/lib/modules/2.2.14-5.0/scsi/BusLogic.o
/lib/modules/2.2.14-5.0/scsi/NCR53c406a.o
/lib/modules/2.2.14-5.0/scsi/a100u2w.o
/lib/modules/2.2.14-5.0/scsi/advansys.o
/lib/modules/2.2.14-5.0/scsi/aha152x.o
/lib/modules/2.2.14-5.0/scsi/aha1542.o
/lib/modules/2.2.14-5.0/scsi/aha1740.o
/lib/modules/2.2.14-5.0/scsi/aic7xxx.o
/lib/modules/2.2.14-5.0/scsi/atp870u.o
/lib/modules/2.2.14-5.0/scsi/dtc.o
/lib/modules/2.2.14-5.0/scsi/eata.o
/lib/modules/2.2.14-5.0/scsi/eata_dma.o
/lib/modules/2.2.14-5.0/scsi/eata_pio.o
/lib/modules/2.2.14-5.0/scsi/fdomain.o
/lib/modules/2.2.14-5.0/scsi/g_NCR5380.o
/lib/modules/2.2.14-5.0/scsi/gdth.o
/lib/modules/2.2.14-5.0/scsi/ide-scsi.o
/lib/modules/2.2.14-5.0/scsi/imm.o
/lib/modules/2.2.14-5.0/scsi/in2000.o
/lib/modules/2.2.14-5.0/scsi/initio.o
/lib/modules/2.2.14-5.0/scsi/ips.o
/lib/modules/2.2.14-5.0/scsi/megaraid.o
/lib/modules/2.2.14-5.0/scsi/ncr53c8xx.o
/lib/modules/2.2.14-5.0/scsi/pas16.o
/lib/modules/2.2.14-5.0/scsi/pci2000.o
/lib/modules/2.2.14-5.0/scsi/pci2220i.o
/lib/modules/2.2.14-5.0/scsi/ppa.o
/lib/modules/2.2.14-5.0/scsi/psi240i.o
/lib/modules/2.2.14-5.0/scsi/qlogicfas.o
/lib/modules/2.2.14-5.0/scsi/qlogicfc.o
/lib/modules/2.2.14-5.0/scsi/qlogicisp.o
/lib/modules/2.2.14-5.0/scsi/scsi_debug.o
/lib/modules/2.2.14-5.0/scsi/seagate.o
/lib/modules/2.2.14-5.0/scsi/sg.o
/lib/modules/2.2.14-5.0/scsi/sim710.o
/lib/modules/2.2.14-5.0/scsi/st.o
/lib/modules/2.2.14-5.0/scsi/sym53c416.o
/lib/modules/2.2.14-5.0/scsi/sym53c8xx.o
/lib/modules/2.2.14-5.0/scsi/t128.o
/lib/modules/2.2.14-5.0/scsi/tmscsim.o
/lib/modules/2.2.14-5.0/scsi/u14-34f.o
/lib/modules/2.2.14-5.0/scsi/ultrastor.o
/lib/modules/2.2.14-5.0/scsi/wd7000.o
/lib/modules/2.2.14-5.0/video/matroxfb.o
/lib/modules/2.2.14-5.0/video/mdacon.o
/usr/X11R6/lib/X11/fonts/misc/fonts.dir
/usr/doc/libtool-1.3.4/demo/Makefile.in
/usr/doc/libtool-1.3.4/demo/aclocal.m4
/usr/lib/umb-scheme/slibcat
/usr/share/fonts/ISO8859-2/100dpi/fonts.dir
/usr/share/fonts/ISO8859-2/75dpi/fonts.dir

/usr/share/fonts/ISO8859-7/100dpi/fonts.dir
/usr/share/fonts/ISO8859-7/75dpi/fonts.dir
/usr/share/fonts/fontmap
/usr/share/texmf/ls-R
/usr/src/linux-2.2.14/Documentation/Configure.help
/usr/src/linux-2.2.14/arch/i386/defconfig
/usr/src/linux-2.2.14/drivers/scsi/.depend
/usr/src/linux-2.2.14/drivers/sound/.depend
/usr/src/linux-2.2.14/include/linux/autoconf.h
/usr/src/linux-2.2.14/net/Config.in
/usr/src/linux-2.2.14/net/Makefile
/usr/src/linux-2.2.14/net/ipv4/.depend
/usr/src/linux-2.2.14/net/ipv4/af_inet.c
/var/log/lastlog
/var/log/sendmail.st

=====
Date: Tue, Mar 6 2001 19:55:04

Comments added by wes

The IP address in /var/log/wtmp that was connecting as the user sdr is:

200.51.210.75

#####

According to ARIN:

TELINTAR UOS (NETBLK-TELINTAR-UOS4-AR) TELINTAR-UOS4-AR

200.51.0.0 - 200.51.255.255

Advance Telecomunicaciones S.A. (NETBLK-ADVANCE-INTERACTIV-AR)

ADVANCE-INTERACTIV-AR

200.51.208.0 - 200.51.217.255

#####

And Telintar:

TELINTAR UOS (NETBLK-TELINTAR-UOS4-AR)

Tucuman 1 4th floor

Buenos Aires, Capital Federal 1001

AR

Netname: TELINTAR-UOS4-AR

Netblock: 200.51.0.0 - 200.51.255.255

Maintainer: TLAR

Coordinator:

Tld, Poc (PT92-ARIN) noc@telintar.net.ar

54-11-4370-1555 (FAX) 54-11-4373-9341

Domain System inverse mapping provided by:

VENUS.SUR.TELINTAR.COM.AR 200.0.193.100
XANADU.SUR.TELINTAR.COM.AR 200.0.193.98

ADDRESSES WITHIN THIS BLOCK ARE NON-PORTABLE

Record last updated on 24-Feb-2000.
Database last updated on 6-Mar-2001 19:02:59 EDT.

#####

And Advance Telecommunications:

Advance Telecomunicaciones S.A. (NETBLK-ADVANCE-INTERACTIV-AR)
Tucuman 1,Piso 8
Buenos Aires, Buenos Aires
AR

Netname: ADVANCE-INTERACTIV-AR
Netblock: 200.51.208.0 - 200.51.217.255
Maintainer: AVTC

Coordinator:
Tld, Poc (PT92-ARIN) noc@telintar.net.ar
54-11-4370-1555 (FAX) 54-11-4373-9341

Domain System inverse mapping provided by:

DNS1.INTERNET-MRSE-SOLUTIONS.COM 200.51.254.254
DNS2.INTERNET-MRSE-SOLUTIONS.COM 200.51.254.251

Record last updated on 17-Nov-2000.
Database last updated on 6-Mar-2001 19:02:59 EDT.

Appendix 1.2

Appendix 1.3

Detect 3 - Full List of Alerts

Nov 9 14:31:53 WBID001 Connect to Back Orifice port: 131.107.3.74:31337 ->
sanjose.network3.83.204:80
Nov 12 15:11:46 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.76:49549 ->
seattle.network.3.254:80

Nov 12 15:11:46 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.74:35837 -> seattle.network.3.254:80
Nov 12 15:11:46 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.76:49550 -> seattle.network.3.254:80
Nov 12 15:11:46 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.74:35839 -> seattle.network.3.254:80
Nov 12 15:11:46 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.76:49551 -> seattle.network.3.254:80
Nov 12 15:11:47 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.76:49552 -> seattle.network.3.254:80
Nov 12 15:11:47 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.76:49553 -> seattle.network.3.254:80
Nov 12 15:11:47 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.87:3390 -> seattle.network.3.254:80
Nov 12 15:11:47 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.76:49554 -> seattle.network.3.254:80
Nov 12 15:11:47 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.76:49555 -> seattle.network.3.254:80
Nov 12 15:11:47 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.76:49556 -> seattle.network.3.254:80
Nov 12 15:11:47 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.76:49557 -> seattle.network.3.254:80
Nov 12 15:11:47 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.74:35842 -> seattle.network.3.254:80
Nov 12 15:11:47 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.76:49558 -> seattle.network.3.254:80
Nov 12 15:11:47 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.76:49559 -> seattle.network.3.254:80
Nov 12 15:11:47 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.76:49560 -> seattle.network.3.254:80
Nov 12 15:11:47 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.76:49561 -> seattle.network.3.254:80
Nov 12 15:11:47 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.76:49564 -> seattle.network.3.254:80
Nov 12 15:11:47 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.76:49566 -> seattle.network.3.254:80
Nov 12 15:11:47 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.76:49567 -> seattle.network.3.254:80
Nov 12 15:11:47 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.76:49568 -> seattle.network.3.254:80
Nov 12 15:11:47 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.76:49570 -> seattle.network.3.254:80
Nov 12 15:11:47 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.76:49572 -> seattle.network.3.254:80
Nov 12 15:11:47 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.76:49574 -> seattle.network.3.254:80
Nov 12 15:11:47 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.76:49575 -> seattle.network.3.254:80
Nov 12 15:11:47 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.76:49576 -> seattle.network.3.254:80
Nov 12 15:11:47 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.76:49577 -> seattle.network.3.254:80

Nov 12 15:11:47 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.76:49578 ->
seattle.network.3.254:80
Nov 12 15:11:47 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.76:49581 ->
seattle.network.3.254:80
Nov 12 15:11:47 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.76:49582 ->
seattle.network.3.254:80
Nov 12 15:11:47 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.76:49583 ->
seattle.network.3.254:80
Nov 12 15:11:47 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.76:49584 ->
seattle.network.3.254:80
Nov 12 15:11:47 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.76:49585 ->
seattle.network.3.254:80
Nov 12 15:11:47 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.76:49586 ->
seattle.network.3.254:80
Nov 12 15:11:47 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.76:49588 ->
seattle.network.3.254:80
Nov 12 15:11:47 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.76:49589 ->
seattle.network.3.254:80
Nov 12 15:11:47 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.76:49590 ->
seattle.network.3.254:80
Nov 12 15:11:47 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.76:49591 ->
seattle.network.3.254:80
Nov 12 15:11:47 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.76:49592 ->
seattle.network.3.254:80
Nov 12 15:11:47 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.76:49593 ->
seattle.network.3.254:80
Nov 12 15:11:47 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.76:49595 ->
seattle.network.3.254:80
Nov 12 15:11:47 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.76:49596 ->
seattle.network.3.254:80
Nov 12 15:11:47 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.74:35856 ->
seattle.network.3.254:80
Nov 12 15:11:48 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.76:49599 ->
seattle.network.3.254:80
Nov 13 10:23:17 WBID001 Connect to Back Orifice port: 131.107.3.83:31337 ->
seattle.network.3.254:80
Nov 13 17:20:55 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.89:56668 ->
seattle.network.3.254:80
Nov 13 17:20:55 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.84:19150 ->
seattle.network.3.254:80
Nov 15 06:24:41 WBID001 Connect to Back Orifice port: 131.107.3.85:31337 ->
seattle.network.3.254:80
Nov 15 18:15:11 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.77:62335 ->
seattle.network.3.254:80
Nov 15 18:15:11 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.77:62498 ->
seattle.network.3.254:80
Nov 15 18:15:11 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.77:62500 ->
seattle.network.3.254:80
Nov 15 18:15:11 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.77:62503 ->
seattle.network.3.254:80
Nov 15 18:15:11 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.77:62505 ->
seattle.network.3.254:80

Nov 15 18:15:11 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.77:62513 ->
seattle.network.3.254:80
Nov 15 18:15:11 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.77:62514 ->
seattle.network.3.254:80
Nov 15 18:15:11 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.77:62519 ->
seattle.network.3.254:80
Nov 15 18:15:11 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.77:62520 ->
seattle.network.3.254:80
Nov 15 18:15:12 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.77:62523 ->
seattle.network.3.254:80
Nov 15 18:15:12 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.77:62524 ->
seattle.network.3.254:80
Nov 15 18:15:12 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.77:62529 ->
seattle.network.3.254:80
Nov 15 18:15:12 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.77:62530 ->
seattle.network.3.254:80
Nov 15 18:15:12 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.77:62534 ->
seattle.network.3.254:80
Nov 15 18:15:12 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.77:62535 ->
seattle.network.3.254:80
Nov 15 18:15:12 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.77:62538 ->
seattle.network.3.254:80
Nov 15 18:15:12 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.77:62545 ->
seattle.network.3.254:80
Nov 15 18:15:12 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.77:62547 ->
seattle.network.3.254:80
Nov 15 18:15:12 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.77:62552 ->
seattle.network.3.254:80
Nov 15 18:15:12 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.77:62553 ->
seattle.network.3.254:80
Nov 16 15:51:25 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.76:31282 ->
miami.network.166.62:80
Nov 16 16:34:17 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.76:46957 ->
miami.network.166.62:80
Nov 16 16:40:56 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.76:61113 ->
miami.network.166.62:80
Nov 16 17:57:13 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.76:64012 ->
miami.network.166.62:80
Nov 16 21:29:52 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.77:25820 ->
seattle.network.3.254:80
Nov 16 21:29:52 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.77:25828 ->
seattle.network.3.254:80
Nov 16 21:29:52 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.77:25831 ->
seattle.network.3.254:80
Nov 16 21:29:53 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.77:25834 ->
seattle.network.3.254:80
Nov 18 04:37:40 DDoS - mstream client to handler: sanjose.network3.83.204:80 ->
131.107.3.87:12754
Nov 22 12:21:30 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.88:5091 ->
seattle.network.3.254:80
Nov 22 12:21:33 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.77:52553 ->
seattle.network.3.254:80

Nov 22 12:21:33 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.77:52566 ->
seattle.network.3.254:80
Nov 22 12:21:33 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.77:52568 ->
seattle.network.3.254:80
Nov 22 12:21:33 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.77:52572 ->
seattle.network.3.254:80
Nov 22 12:21:33 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.77:52573 ->
seattle.network.3.254:80
Nov 22 12:21:33 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.77:52575 ->
seattle.network.3.254:80
Nov 22 12:21:33 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.77:52576 ->
seattle.network.3.254:80
Nov 22 12:21:33 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.77:52578 ->
seattle.network.3.254:80
Nov 22 12:21:33 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.77:52580 ->
seattle.network.3.254:80
Nov 22 12:21:33 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.77:52581 ->
seattle.network.3.254:80
Nov 22 12:21:33 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.77:52583 ->
seattle.network.3.254:80
Nov 22 12:21:33 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.77:52586 ->
seattle.network.3.254:80
Nov 22 12:21:33 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.77:52588 ->
seattle.network.3.254:80
Nov 22 12:21:33 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.77:52591 ->
seattle.network.3.254:80
Nov 22 12:21:33 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.77:52593 ->
seattle.network.3.254:80
Nov 22 12:21:34 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.77:52596 ->
seattle.network.3.254:80
Nov 22 12:21:34 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.77:52598 ->
seattle.network.3.254:80
Nov 22 12:21:34 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.77:52617 ->
seattle.network.3.254:80
Nov 25 03:58:44 IDS290 - WEB-CGI - infosearch fname: 131.107.3.89:22120 ->
seattle.network.3.254:80
Nov 28 17:37:22 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.77:38122 ->
seattle.network.3.254:80
Nov 28 17:37:55 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.77:39336 ->
seattle.network.3.254:80
Nov 29 00:26:16 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.88:60081 ->
seattle.network.3.254:80
Nov 29 00:26:17 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.85:12898 ->
seattle.network.3.254:80
Nov 29 00:27:07 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.77:36714 ->
seattle.network.3.254:80
Nov 29 00:27:07 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.77:36724 ->
seattle.network.3.254:80
Nov 29 10:19:22 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.76:26590 ->
seattle.network.3.254:80
Nov 29 10:19:22 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.76:26591 ->
seattle.network.3.254:80

Nov 29 10:19:22 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.76:26592 -> seattle.network.3.254:80
Nov 29 10:19:22 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.86:37274 -> seattle.network.3.254:80
Nov 29 10:19:22 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.76:26601 -> seattle.network.3.254:80
Nov 29 10:19:22 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.76:26602 -> seattle.network.3.254:80
Nov 29 10:19:22 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.76:26603 -> seattle.network.3.254:80
Nov 29 10:19:22 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.78:8885 -> seattle.network.3.254:80
Nov 29 10:19:23 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.86:37290 -> seattle.network.3.254:80
Nov 29 10:19:23 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.87:64821 -> seattle.network.3.254:80
Nov 29 10:19:23 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.86:37292 -> seattle.network.3.254:80
Nov 29 17:55:55 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.76:32302 -> seattle.network.3.254:80
Nov 29 17:55:55 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.76:32303 -> seattle.network.3.254:80
Nov 29 17:55:55 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.76:32305 -> seattle.network.3.254:80
Nov 29 17:55:55 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.76:32306 -> seattle.network.3.254:80
Nov 29 17:55:55 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.76:32307 -> seattle.network.3.254:80
Nov 29 17:55:56 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.76:32315 -> seattle.network.3.254:80
Nov 29 17:55:56 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.76:32316 -> seattle.network.3.254:80
Nov 29 17:55:56 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.76:32319 -> seattle.network.3.254:80
Nov 29 17:55:56 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.76:32320 -> seattle.network.3.254:80
Nov 29 17:55:56 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.76:32324 -> seattle.network.3.254:80
Nov 29 17:55:56 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.76:32326 -> seattle.network.3.254:80
Nov 29 17:55:56 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.76:32328 -> seattle.network.3.254:80
Nov 29 17:55:56 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.76:32330 -> seattle.network.3.254:80
Nov 29 17:55:56 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.76:32333 -> seattle.network.3.254:80
Nov 29 17:55:56 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.76:32343 -> seattle.network.3.254:80
Nov 29 17:55:56 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.76:32344 -> seattle.network.3.254:80
Nov 29 17:55:56 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.76:32350 -> seattle.network.3.254:80

Nov 29 17:55:56 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.76:32352 ->
seattle.network.3.254:80
Nov 29 17:55:56 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.76:32361 ->
seattle.network.3.254:80
Nov 29 17:55:56 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.86:8958 ->
seattle.network.3.254:80
Nov 29 17:55:56 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.76:32365 ->
seattle.network.3.254:80
Nov 29 17:55:56 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.76:32368 ->
seattle.network.3.254:80
Nov 29 17:55:57 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.76:32369 ->
seattle.network.3.254:80
Nov 29 17:55:57 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.76:32374 ->
seattle.network.3.254:80
Nov 29 17:55:57 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.76:32376 ->
seattle.network.3.254:80
Nov 29 17:55:57 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.76:32379 ->
seattle.network.3.254:80
Nov 29 17:55:57 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.76:32380 ->
seattle.network.3.254:80
Nov 29 17:55:57 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.76:32381 ->
seattle.network.3.254:80
Nov 29 17:55:57 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.76:32383 ->
seattle.network.3.254:80
Nov 29 17:55:57 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.76:32389 ->
seattle.network.3.254:80
Nov 29 17:55:57 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.86:8997 ->
seattle.network.3.254:80
Nov 29 17:55:57 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.76:32398 ->
seattle.network.3.254:80
Nov 29 17:55:57 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.76:32399 ->
seattle.network.3.254:80
Nov 29 17:55:57 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.76:32400 ->
seattle.network.3.254:80
Nov 29 17:55:57 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.76:32404 ->
seattle.network.3.254:80
Nov 29 17:55:57 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.76:32403 ->
seattle.network.3.254:80
Nov 29 17:55:58 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.76:32406 ->
seattle.network.3.254:80
Nov 29 17:55:58 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.76:32407 ->
seattle.network.3.254:80
Nov 29 17:55:58 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.76:32409 ->
seattle.network.3.254:80
Nov 29 17:55:58 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.86:9018 ->
seattle.network.3.254:80
Nov 29 17:55:58 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.86:9021 ->
seattle.network.3.254:80
Nov 29 17:55:58 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.87:24195 ->
seattle.network.3.254:80
Nov 29 17:55:58 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.86:9026 ->
seattle.network.3.254:80

Nov 30 12:51:58 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.88:24097 -> nyc.network4.21.126:80
Dec 6 11:23:49 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.87:35141 -> nyc.network4.21.126:80
Dec 6 11:38:48 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.70:61060 -> seattle.network.3.254:80
Dec 6 16:41:00 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.76:49629 -> washingtondc.network3.117.254:80
Dec 6 16:41:00 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.76:49631 -> washingtondc.network3.117.254:80
Dec 6 16:41:00 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.76:49638 -> washingtondc.network3.117.254:80
Dec 6 16:41:01 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.76:49648 -> washingtondc.network3.117.254:80
Dec 6 16:41:04 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.76:49738 -> washingtondc.network3.117.254:80
Dec 6 16:41:05 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.74:55970 -> washingtondc.network3.117.254:80
Dec 6 16:41:06 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.76:49802 -> washingtondc.network3.117.254:80
Dec 6 16:41:06 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.76:49807 -> washingtondc.network3.117.254:80
Dec 6 16:41:07 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.76:49869 -> washingtondc.network3.117.254:80
Dec 11 18:04:57 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.88:49529 -> chicago.network4.139.254:80
Dec 12 19:23:02 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.88:5465 -> nyc.network4.21.126:80
Dec 12 19:23:02 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.78:29592 -> nyc.network4.21.126:80
Dec 26 15:49:22 IDS297 - WEB MISC - http-directory-traversal 1: 131.107.3.88:32814 -> philadelphia.network.72.254:80
Dec 26 15:49:54 IDS297 - WEB MISC - http-directory-traversal 1: 131.107.3.88:33379 -> philadelphia.network.72.254:80
Jan 3 13:17:34 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.88:42112 -> losangeles.network.99.254:80
Jan 3 13:17:35 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.73:6508 -> chicago.network4.139.254:80
Jan 3 13:17:35 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.88:42150 -> losangeles.network.99.254:80
Jan 3 13:17:35 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.78:4240 -> sandiego.network.13.126:80
Jan 3 13:17:35 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.88:42174 -> losangeles.network.99.254:80
Jan 3 13:17:36 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.70:50356 -> losangeles.network.99.254:80
Jan 3 13:17:36 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.88:42195 -> losangeles.network.99.254:80
Jan 3 13:17:36 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.89:64920 -> losangeles.network.99.254:80
Jan 3 13:17:36 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.88:42210 -> losangeles.network.99.254:80

Jan 3 13:17:36 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.87:24210 ->
chicago.network4.139.254:80
Jan 3 13:17:37 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.88:42245 ->
losangeles.network.99.254:80
Jan 3 13:17:37 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.87:24220 ->
chicago.network4.139.254:80
Jan 3 13:17:37 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.88:42265 ->
losangeles.network.99.254:80
Jan 3 13:17:37 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.91:59724 ->
chicago.network4.139.254:80
Jan 3 13:17:37 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.88:42278 ->
losangeles.network.99.254:80
Jan 3 13:17:37 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.70:50361 ->
losangeles.network.99.254:80
Jan 3 13:17:37 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.88:42287 ->
losangeles.network.99.254:80
Jan 3 13:17:38 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.75:50826 ->
losangeles.network.99.254:80
Jan 3 13:17:38 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.88:42307 ->
losangeles.network.99.254:80
Jan 3 13:17:38 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.91:59735 ->
chicago.network4.139.254:80
Jan 3 13:17:38 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.88:42316 ->
losangeles.network.99.254:80
Jan 3 13:17:38 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.88:42324 ->
losangeles.network.99.254:80
Jan 3 13:17:38 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.88:42338 ->
losangeles.network.99.254:80
Jan 3 13:17:38 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.78:4412 ->
sandiego.network.13.126:80
Jan 3 13:17:39 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.88:42336 ->
losangeles.network.99.254:80
Jan 3 13:17:39 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.91:59773 ->
chicago.network4.139.254:80
Jan 3 13:17:39 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.88:42402 ->
losangeles.network.99.254:80
Jan 3 13:17:39 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.77:30592 ->
chicago.network4.139.254:80
Jan 3 13:17:40 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.88:42476 ->
losangeles.network.99.254:80
Jan 3 13:17:43 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.83:3220 ->
losangeles.network.99.254:80
Jan 3 13:17:43 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.88:42619 ->
losangeles.network.99.254:80
Jan 3 13:17:46 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.83:3338 ->
losangeles.network.99.254:80
Jan 3 13:17:48 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.88:42923 ->
losangeles.network.99.254:80
Jan 3 13:17:48 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.87:24717 ->
chicago.network4.139.254:80
Jan 3 13:17:48 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.88:42936 ->
losangeles.network.99.254:80

Jan 3 13:17:48 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.70:50447 -> losangeles.network.99.254:80
Jan 3 13:17:49 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.88:42953 -> losangeles.network.99.254:80
Jan 3 13:17:49 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.86:24934 -> seattle.network.3.254:80
Jan 3 13:17:49 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.88:42975 -> losangeles.network.99.254:80
Jan 3 13:17:49 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.87:24723 -> chicago.network4.139.254:80
Jan 3 13:17:49 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.88:42987 -> losangeles.network.99.254:80
Jan 3 13:17:49 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.85:47794 -> losangeles.network.99.254:80
Jan 3 13:17:49 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.88:43010 -> losangeles.network.99.254:80
Jan 3 13:17:49 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.83:3329 -> losangeles.network.99.254:80
Jan 3 13:17:49 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.88:43031 -> losangeles.network.99.254:80
Jan 3 13:17:49 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.70:51097 -> losangeles.network.99.254:80
Jan 3 13:29:15 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.88:20355 -> losangeles.network.99.254:80
Jan 3 13:29:16 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.78:37068 -> sandiego.network.13.126:80
Jan 3 13:29:16 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.88:20400 -> losangeles.network.99.254:80
Jan 3 13:29:16 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.79:20693 -> sanfrancisco.network.247.190:80
Jan 3 13:29:17 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.88:20417 -> losangeles.network.99.254:80
Jan 3 13:29:17 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.84:48101 -> seattle.network.3.254:80
Jan 3 13:29:17 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.88:20422 -> losangeles.network.99.254:80
Jan 3 13:29:17 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.74:17145 -> seattle.network.3.254:80
Jan 3 13:29:17 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.88:20436 -> losangeles.network.99.254:80
Jan 3 13:29:17 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.89:38848 -> seattle.network.3.254:80
Jan 3 13:29:17 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.88:20447 -> losangeles.network.99.254:80
Jan 3 13:29:17 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.89:38852 -> seattle.network.3.254:80
Jan 3 13:29:17 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.88:20455 -> losangeles.network.99.254:80
Jan 3 13:29:17 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.76:19440 -> seattle.network.3.254:80
Jan 3 13:29:18 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.88:20477 -> losangeles.network.99.254:80

Jan 3 13:29:18 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.86:56841 -> seattle.network.3.254:80
Jan 3 13:29:18 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.88:20493 -> losangeles.network.99.254:80
Jan 3 13:29:18 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.78:37154 -> sandiego.network.13.126:80
Jan 3 13:29:19 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.88:20533 -> losangeles.network.99.254:80
Jan 3 13:29:19 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.73:35846 -> chicago.network4.139.254:80
Jan 3 13:29:19 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.88:20552 -> losangeles.network.99.254:80
Jan 3 13:29:19 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.79:20806 -> sanfrancisco.network.247.190:80
Jan 3 13:29:19 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.88:20574 -> losangeles.network.99.254:80
Jan 3 13:29:19 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.78:37221 -> sandiego.network.13.126:80
Jan 3 13:29:19 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.88:20593 -> losangeles.network.99.254:80
Jan 3 13:29:19 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.91:29743 -> seattle.network.3.254:80
Jan 3 13:29:20 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.88:20602 -> losangeles.network.99.254:80
Jan 3 13:29:20 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.70:27986 -> sanfrancisco.network.247.190:80
Jan 3 13:29:20 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.88:20623 -> losangeles.network.99.254:80
Jan 3 13:29:20 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.84:48264 -> seattle.network.3.254:80
Jan 3 13:29:20 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.88:20632 -> losangeles.network.99.254:80
Jan 3 13:29:20 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.83:30473 -> sanfrancisco.network.247.190:80
Jan 3 13:29:23 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.88:20786 -> losangeles.network.99.254:80
Jan 3 13:29:23 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.84:48381 -> seattle.network.3.254:80
Jan 3 13:29:23 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.88:20797 -> losangeles.network.99.254:80
Jan 3 13:29:23 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.86:57088 -> seattle.network.3.254:80
Jan 3 13:29:23 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.88:20815 -> losangeles.network.99.254:80
Jan 3 13:29:24 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.79:21007 -> sanfrancisco.network.247.190:80
Jan 3 13:29:24 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.88:20826 -> losangeles.network.99.254:80
Jan 3 13:29:24 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.78:37437 -> sandiego.network.13.126:80
Jan 3 13:30:16 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.88:23652 -> losangeles.network.99.254:80

Jan 3 13:30:16 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.86:59810 ->
seattle.network.3.254:80
Jan 3 13:30:38 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.77:59377 ->
chicago.network4.139.254:80
Jan 3 13:30:39 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.77:59416 ->
chicago.network4.139.254:80
Jan 3 14:40:47 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.88:22307 ->
losangeles.network.99.254:80
Jan 3 14:40:48 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.88:22378 ->
losangeles.network.99.254:80
Jan 3 14:40:49 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.88:22431 ->
losangeles.network.99.254:80
Jan 3 14:40:49 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.84:61753 ->
nyc.network1.165.116:80
Jan 3 14:40:51 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.88:22618 ->
losangeles.network.99.254:80
Jan 3 14:40:51 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.84:61852 ->
nyc.network1.165.116:80
Jan 3 14:40:51 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.88:22659 ->
losangeles.network.99.254:80
Jan 3 14:40:52 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.88:22701 ->
losangeles.network.99.254:80
Jan 3 14:40:52 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.88:22736 ->
losangeles.network.99.254:80
Jan 3 14:40:53 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.70:13488 ->
seattle.network.3.254:80
Jan 3 14:40:53 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.88:22748 ->
losangeles.network.99.254:80
Jan 3 14:40:53 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.88:22758 ->
losangeles.network.99.254:80
Jan 3 14:40:53 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.88:22771 ->
losangeles.network.99.254:80
Jan 3 14:40:53 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.73:25196 ->
dallas.network.182.204:80
Jan 3 14:40:53 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.88:22792 ->
losangeles.network.99.254:80
Jan 3 14:40:54 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.84:61881 ->
nyc.network1.165.116:80
Jan 3 14:40:54 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.88:22823 ->
losangeles.network.99.254:80
Jan 3 14:40:54 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.74:57214 ->
seattle.network.3.254:80
Jan 3 14:40:54 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.88:22850 ->
losangeles.network.99.254:80
Jan 3 14:40:54 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.70:13574 ->
seattle.network.3.254:80
Jan 3 14:40:54 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.88:22868 ->
losangeles.network.99.254:80
Jan 3 14:40:55 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.83:16379 ->
washingtondc.network3.117.254:80
Jan 3 14:40:55 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.88:22913 ->
losangeles.network.99.254:80

Jan 3 14:40:55 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.88:22955 -> losangeles.network.99.254:80
Jan 3 14:40:55 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.74:57303 -> seattle.network.3.254:80
Jan 3 14:40:55 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.88:22974 -> losangeles.network.99.254:80
Jan 3 14:40:57 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.88:23097 -> losangeles.network.99.254:80
Jan 3 14:40:57 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.88:23131 -> losangeles.network.99.254:80
Jan 3 14:40:57 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.88:23139 -> losangeles.network.99.254:80
Jan 3 14:40:58 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.88:23159 -> losangeles.network.99.254:80
Jan 3 14:40:58 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.88:23174 -> losangeles.network.99.254:80
Jan 3 14:40:58 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.88:23186 -> losangeles.network.99.254:80
Jan 3 14:40:58 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.88:23184 -> losangeles.network.99.254:80
Jan 3 14:41:02 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.73:25525 -> dallas.network.182.204:80
Jan 3 14:41:02 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.88:23528 -> losangeles.network.99.254:80
Jan 3 14:41:02 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.84:61982 -> nyc.network1.165.116:80
Jan 3 14:41:19 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.88:24783 -> losangeles.network.99.254:80
Jan 3 14:57:16 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.88:40825 -> losangeles.network.99.254:80
Jan 3 14:57:16 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.74:42925 -> seattle.network.3.254:80
Jan 3 14:57:17 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.88:40855 -> losangeles.network.99.254:80
Jan 3 14:57:17 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.91:30349 -> losangeles.network.99.254:80
Jan 3 14:57:17 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.88:40904 -> losangeles.network.99.254:80
Jan 3 14:57:17 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.75:39783 -> dallas.network.182.204:80
Jan 3 14:57:18 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.88:40953 -> losangeles.network.99.254:80
Jan 3 14:57:18 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.73:7109 -> dallas.network.182.204:80
Jan 3 14:57:18 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.88:41034 -> losangeles.network.99.254:80
Jan 3 14:57:18 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.77:64221 -> losangeles.network.99.254:80
Jan 3 14:57:19 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.88:41079 -> losangeles.network.99.254:80
Jan 3 14:57:19 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.75:39796 -> dallas.network.182.204:80

Jan 3 14:57:19 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.88:41113 -> losangeles.network.99.254:80
Jan 3 14:57:19 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.91:30356 -> losangeles.network.99.254:80
Jan 3 14:57:19 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.88:41149 -> losangeles.network.99.254:80
Jan 3 14:57:19 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.78:47160 -> sandiego.network.13.126:80
Jan 3 14:57:20 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.88:41189 -> losangeles.network.99.254:80
Jan 3 14:57:20 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.79:39109 -> dallas.network.182.204:80
Jan 3 14:57:20 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.88:41264 -> losangeles.network.99.254:80
Jan 3 14:57:21 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.78:47220 -> sandiego.network.13.126:80
Jan 3 14:57:21 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.88:41318 -> losangeles.network.99.254:80
Jan 3 14:57:21 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.91:30448 -> losangeles.network.99.254:80
Jan 3 14:57:21 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.88:41355 -> losangeles.network.99.254:80
Jan 3 14:57:21 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.73:7198 -> dallas.network.182.204:80
Jan 3 14:57:21 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.88:41417 -> losangeles.network.99.254:80
Jan 3 14:57:22 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.88:41442 -> losangeles.network.99.254:80
Jan 3 14:57:22 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.88:41459 -> losangeles.network.99.254:80
Jan 3 14:57:22 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.70:9651 -> dallas.network.182.204:80
Jan 3 14:57:22 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.88:41458 -> losangeles.network.99.254:80
Jan 3 14:57:22 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.83:55793 -> washingtondc.network.3.117.254:80
Jan 3 14:57:23 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.88:41601 -> losangeles.network.99.254:80
Jan 3 14:57:23 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.88:41622 -> losangeles.network.99.254:80
Jan 3 14:57:46 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.88:44003 -> losangeles.network.99.254:80
Jan 3 14:57:46 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.89:46528 -> seattle.network.3.254:80
Jan 3 14:57:46 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.88:44002 -> losangeles.network.99.254:80
Jan 3 14:57:46 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.89:46531 -> seattle.network.3.254:80
Jan 3 14:57:47 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.88:44049 -> losangeles.network.99.254:80
Jan 3 14:57:47 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.89:46541 -> seattle.network.3.254:80

Jan 4 21:05:19 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.88:41944 -> chicago.network4.139.254:80

Jan 4 21:07:38 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.88:45074 -> chicago.network4.139.254:80

Jan 4 21:07:38 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.84:47462 -> nyc.network1.165.116:80

Jan 4 21:07:39 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.88:45088 -> chicago.network4.139.254:80

Jan 4 21:07:39 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.84:47507 -> nyc.network1.165.116:80

Jan 4 21:07:40 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.88:45122 -> chicago.network4.139.254:80

Jan 4 21:10:00 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.77:38859 -> nyc.network2.42.51:80

Jan 4 21:10:00 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.84:55052 -> nyc.network1.165.116:80

Jan 15 15:28:01 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.79:50275 -> chicago.network2.97.204:80

Jan 15 15:28:01 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.79:50277 -> chicago.network2.97.204:80

Jan 15 15:28:01 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.79:50283 -> chicago.network2.97.204:80

Jan 15 15:28:01 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.79:50295 -> chicago.network2.97.204:80

Jan 15 15:28:02 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.79:50300 -> chicago.network2.97.204:80

Jan 15 15:28:02 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.79:50322 -> chicago.network2.97.204:80

Jan 15 15:28:02 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.79:50326 -> chicago.network2.97.204:80

Jan 15 15:28:02 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.79:50329 -> chicago.network2.97.204:80

Jan 15 15:28:02 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.79:50333 -> chicago.network2.97.204:80

Jan 15 15:28:02 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.79:50350 -> chicago.network2.97.204:80

Jan 15 15:28:02 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.79:50352 -> chicago.network2.97.204:80

Jan 15 15:28:03 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.79:50354 -> chicago.network2.97.204:80

Jan 15 15:28:03 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.79:50355 -> chicago.network2.97.204:80

Jan 15 15:28:03 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.79:50360 -> chicago.network2.97.204:80

Jan 15 15:28:03 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.79:50366 -> chicago.network2.97.204:80

Jan 15 15:28:03 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.79:50367 -> chicago.network2.97.204:80

Jan 15 15:28:03 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.79:50368 -> chicago.network2.97.204:80

Jan 15 15:28:03 IDS305 - WEB IIS - View Source via Translate Header: 131.107.3.79:50382 -> chicago.network2.97.204:80

*****SF Seq: 0x792D8220 Ack: 0x17767A09 Win: 0x404 TcpLen: 20

+++++

03/11-18:15:14.249497 0:1:42:BB:E8:41 -> 0:6:29:DE:63:EF type:0x800 len:0x3C
216.63.85.125:53 -> nyc.network2.42.36:53 TCP TTL:22 TOS:0x0 ID:39426 IpLen:20 DgmLen:40

*****SF Seq: 0x278A37BA Ack: 0xD341115 Win: 0x404 TcpLen: 20

+++++

03/11-18:15:14.352073 0:1:42:BB:E8:41 -> 0:6:29:DE:60:69 type:0x800 len:0x3C
216.63.85.125:53 -> nyc.network2.42.41:53 TCP TTL:22 TOS:0x0 ID:39426 IpLen:20 DgmLen:40

*****SF Seq: 0x278A37BA Ack: 0xD341115 Win: 0x404 TcpLen: 20

+++++

03/11-18:15:14.366134 0:1:42:BB:E8:41 -> 0:6:29:DE:64:5A type:0x800 len:0x3C
216.63.85.125:53 -> nyc.network2.42.42:53 TCP TTL:22 TOS:0x0 ID:39426 IpLen:20 DgmLen:40

*****SF Seq: 0x278A37BA Ack: 0xD341115 Win: 0x404 TcpLen: 20

+++++

03/11-18:15:14.386605 0:1:42:BB:E8:41 -> 0:6:29:DE:66:EC type:0x800 len:0x3C
216.63.85.125:53 -> nyc.network2.42.43:53 TCP TTL:22 TOS:0x0 ID:39426 IpLen:20 DgmLen:40

*****SF Seq: 0x278A37BA Ack: 0xD341115 Win: 0x404 TcpLen: 20

+++++

03/11-18:15:14.404925 0:1:42:BB:E8:41 -> 0:6:29:DE:66:F8 type:0x800 len:0x3C
216.63.85.125:53 -> nyc.network2.42.44:53 TCP TTL:22 TOS:0x0 ID:39426 IpLen:20 DgmLen:40

*****SF Seq: 0x278A37BA Ack: 0xD341115 Win: 0x404 TcpLen: 20

+++++

03/11-18:15:14.421792 0:1:42:BB:E8:41 -> 0:6:29:DE:67:8E type:0x800 len:0x3C
216.63.85.125:53 -> nyc.network2.42.45:53 TCP TTL:22 TOS:0x0 ID:39426 IpLen:20 DgmLen:40

*****SF Seq: 0x278A37BA Ack: 0xD341115 Win: 0x404 TcpLen: 20

+++++

03/11-18:15:14.442158 0:1:42:BB:E8:41 -> 0:6:29:DE:65:ED type:0x800 len:0x3C
216.63.85.125:53 -> nyc.network2.42.46:53 TCP TTL:22 TOS:0x0 ID:39426 IpLen:20 DgmLen:40

*****SF Seq: 0x278A37BA Ack: 0xD341115 Win: 0x404 TcpLen: 20

+++++

03/11-18:15:14.465084 0:1:42:BB:E8:41 -> 0:6:29:DE:60:37 type:0x800 len:0x3C
216.63.85.125:53 -> nyc.network2.42.47:53 TCP TTL:22 TOS:0x0 ID:39426 IpLen:20 DgmLen:40

*****SF Seq: 0x278A37BA Ack: 0xD341115 Win: 0x404 TcpLen: 20

+++++

03/11-18:15:14.482268 0:1:42:BB:E8:41 -> 0:6:29:DE:60:5F type:0x800 len:0x3C
216.63.85.125:53 -> nyc.network2.42.48:53 TCP TTL:22 TOS:0x0 ID:39426 IpLen:20 DgmLen:40
*****SF Seq: 0x278A37BA Ack: 0xD341115 Win: 0x404 TcpLen: 20

=====
=====

03/11-18:15:14.525256 0:1:42:BB:E8:41 -> 2:E0:52:7:8B:90 type:0x800 len:0x3C
216.63.85.125:53 -> nyc.network2.42.50:53 TCP TTL:22 TOS:0x0 ID:39426 IpLen:20 DgmLen:40
*****SF Seq: 0x278A37BA Ack: 0xD341115 Win: 0x404 TcpLen: 20

=====
=====

03/11-18:15:14.540303 0:1:42:BB:E8:41 -> 2:E0:52:7:8B:8E type:0x800 len:0x3C
216.63.85.125:53 -> nyc.network2.42.51:53 TCP TTL:22 TOS:0x0 ID:39426 IpLen:20 DgmLen:40
*****SF Seq: 0x278A37BA Ack: 0xD341115 Win: 0x404 TcpLen: 20

=====
=====

03/11-18:15:14.581266 0:1:42:BB:E8:41 -> 2:E0:52:7:8B:8E type:0x800 len:0x3C
216.63.85.125:53 -> nyc.network2.42.54:53 TCP TTL:22 TOS:0x0 ID:39426 IpLen:20 DgmLen:40
*****SF Seq: 0x278A37BA Ack: 0xD341115 Win: 0x404 TcpLen: 20

=====
=====

03/11-18:15:14.690664 0:1:42:BB:E8:41 -> 2:E0:52:7:8B:8E type:0x800 len:0x3C
216.63.85.125:53 -> nyc.network2.42.58:53 TCP TTL:22 TOS:0x0 ID:39426 IpLen:20 DgmLen:40
*****SF Seq: 0x278A37BA Ack: 0xD341115 Win: 0x404 TcpLen: 20

=====
=====

03/11-18:15:14.716722 0:1:42:BB:E8:41 -> 2:E0:52:7:8B:8E type:0x800 len:0x3C
216.63.85.125:53 -> nyc.network2.42.60:53 TCP TTL:22 TOS:0x0 ID:39426 IpLen:20 DgmLen:40
*****SF Seq: 0x278A37BA Ack: 0xD341115 Win: 0x404 TcpLen: 20

=====
=====

03/11-18:15:14.809077 0:1:42:BB:E8:41 -> 2:E0:52:7:8B:8E type:0x800 len:0x3C
216.63.85.125:53 -> nyc.network2.42.65:53 TCP TTL:22 TOS:0x0 ID:39426 IpLen:20 DgmLen:40
*****SF Seq: 0x278A37BA Ack: 0xD341115 Win: 0x404 TcpLen: 20

=====
=====

03/11-18:15:14.828097 0:1:42:BB:E8:41 -> 2:E0:52:7:8B:8E type:0x800 len:0x3C
216.63.85.125:53 -> nyc.network2.42.66:53 TCP TTL:22 TOS:0x0 ID:39426 IpLen:20 DgmLen:40
*****SF Seq: 0x278A37BA Ack: 0xD341115 Win: 0x404 TcpLen: 20

=====
=====

03/11-18:15:14.854092 0:1:42:BB:E8:41 -> 2:E0:52:7:8B:8E type:0x800 len:0x3C
216.63.85.125:53 -> nyc.network2.42.67:53 TCP TTL:22 TOS:0x0 ID:39426 IpLen:20 DgmLen:40
*****SF Seq: 0x278A37BA Ack: 0xD341115 Win: 0x404 TcpLen: 20

====+

=====

Snort processed 28 packets.

Breakdown by protocol:

TCP: 28 (100.000%)

UDP: 0 (0.000%)

ICMP: 0 (0.000%)

FRAGS: 0 (0.000%)

REBUILT: 0 (0.000%)

ARP: 0 (0.000%)

IPv6: 0 (0.000%)

IPX: 0 (0.000%)

OTHER: 0 (0.000%)

=====

ALERTS: 0

LOGGED: 0

PASSED: 0

=====

Appendix 2

Appendix 3

Script: wes.alerts.ipaddrs.sh

#!/bin/bash

#Variable declarations

outfile="wes.alerts.ipaddrs.results"

#All source addresses

echo "Top 10 alert source addresses by occurrence" >> "\$outfile"

cat data_alert* | grep ' -> ' | awk -F ' -> ' '{print \$1}' | awk '{print \$NF}' | awk -F ':' '{print \$1}' | sort | uniq

-c | sort -r -n -k 1 | head --lines=10 >> "\$outfile"

echo "" >> "\$outfile"

#All source ports

echo "Top 10 alert source ports by occurrence" >> "\$outfile"

cat data_alert* | grep ' -> ' | awk -F ' -> ' '{print \$1}' | awk '{print \$NF}' | awk -F ':' '{print \$2}' | sort | uniq

```
-c | sort -r -n -k 1 | head --lines=10 >> "$outfile"  
echo "" >> "$outfile"
```

```
#All destination addresses
```

```
echo "Top 10 alert destination addresses by occurrence" >> "$outfile"  
cat data_alert* | grep ' -> ' | awk -F ' -> ' '{print $2}' | awk '{print $1}' | awk -F ':' '{print $1}' | sort | uniq -c  
| sort -r -n -k 1 | head --lines=10 >> "$outfile"  
echo "" >> "$outfile"
```

```
#All destination ports
```

```
echo "Top 10 alert destination ports by occurrence" >> "$outfile"  
cat data_alert* | grep ' -> ' | awk -F ' -> ' '{print $2}' | awk '{print $1}' | awk -F ':' '{print $2}' | sort | uniq -c  
| sort -r -n -k 1 | head --lines=10 >> "$outfile"  
echo "" >> "$outfile"
```

```
echo "" >> "$outfile"  
echo "" >> "$outfile"
```

```
#Create file with all IP addresses, regardless of source or destination
```

```
cat data* | grep ' -> ' | awk -F ' -> ' '{print $1}' | awk '{print $NF}' | awk -F ':' '{print $1}' | sort | uniq >  
/tmp/junkfile  
cat data* | grep ' -> ' | awk -F ' -> ' '{print $2}' | awk '{print $1}' | awk -F ':' '{print $1}' | sort | uniq >>  
/tmp/junkfile  
cat /tmp/junkfile | sort | uniq -c | sort -r -n -k 1 > wes.allip.tally  
rm -f /tmp/junkfile
```

```
#All source addresses
```

```
echo "Top 10 scan source addresses by occurrence" >> "$outfile"  
cat data_scans* | grep ' -> ' | awk -F ' -> ' '{print $1}' | awk '{print $NF}' | awk -F ':' '{print $1}' | sort |  
uniq -c | sort -r -n -k 1 | head --lines=10 >> "$outfile"  
echo "" >> "$outfile"
```

```
echo "" >> "$outfile"  
echo "" >> "$outfile"
```

```
Script: wes.correlationcounts.sh
```

```
#!/bin/bash
```

```
#Variable declarations
```

```
outfile="wes.correlationcounts.results"
```

```
#All source addresses
```

```
echo "Top 10 alert source addresses" >> "$outfile"  
cat data_alert* | grep ' -> ' | awk -F ' -> ' '{print $1}' | awk '{print $NF}' | awk -F ':' '{print $1}' | sort | uniq  
-c | sort -r -n -k 1 | head --lines=10 >> "$outfile"  
echo "" >> "$outfile"
```

```
#All destination addresses
```

```
echo "Top 10 alert destination addresses" >> "$outfile"
cat data_alert* | grep ' -> ' | awk -F ' -> ' '{print $2}' | awk '{print $1}' | awk -F ':' '{print $1}' | sort | uniq -c
| sort -r -n -k 1 | head --lines=10 >> "$outfile"
echo "" >> "$outfile"
```

#All source addresses

```
echo "Top 10 scan source addresses" >> "$outfile"
cat data_scan* | grep ' -> ' | awk -F ' -> ' '{print $1}' | awk '{print $NF}' | awk -F ':' '{print $1}' | sort | uniq
-c | sort -r -n -k 1 | head --lines=10 >> "$outfile"
echo "" >> "$outfile"
```

#All destination addresses

```
echo "Top 10 scan destination addresses" >> "$outfile"
cat data_scan* | grep ' -> ' | awk -F ' -> ' '{print $2}' | awk '{print $1}' | awk -F ':' '{print $1}' | sort | uniq -
c | sort -r -n -k 1 | head --lines=10 >> "$outfile"
echo "" >> "$outfile"
```

```
echo "" >> "$outfile"
echo "" >> "$outfile"
echo "" >> "$outfile"
```

Script: wes.counts.sh

```
#!/bin/bash
```

#Variable declarations

```
outfile="wes.counts.results"
```

#From all data files in directory

```
echo "##### From all data files #####" >> "$outfile"
echo "" >> "$outfile"
```

```
echo "What preprocessor events seen" >> "$outfile"
```

```
cat data* | grep '\[*\*]' | grep spp | awk -F '\[\*\*\]' '{print $2}' | awk -F ":" '{print $1}' | sort -u >>
"$outfile"
echo "" >> "$outfile"
```

```
echo "What preprocessor events seen and how many" >> "$outfile"
```

```
cat data* | grep '\[*\*]' | grep spp | awk -F '\[\*\*\]' '{print $2}' | awk -F ":" '{print $1}' | sort | uniq -c |
sort -r -n -k 1 >> "$outfile"
echo "" >> "$outfile"
```

```
echo "All non-spp events" >> "$outfile"
```

```
cat data* | grep '\[*\*]' | grep -v spp | awk -F '\[\*\*\]' '{print $2}' | awk -F ":" '{print $1}' | sort -u >>
"$outfile"
echo "" >> "$outfile"
```

```
echo "All non-spp events and how many" >> "$outfile"
```

```
cat data* | grep '\[*\*]' | grep -v spp | awk -F '\[\*\*\]' '{print $2}' | awk -F ":" '{print $1}' | sort | uniq -c |
```

```
sort -r -n -k 1 >> "$outfile"
echo "" >> "$outfile"
```

```
echo "" >> "$outfile"
echo "" >> "$outfile"
```

```
#From all alert data files in directory
echo "##### From alert files only #####" >> "$outfile"
echo "" >> "$outfile"
```

```
echo "What preprocessor events seen" >> "$outfile"
cat data_alert.* | grep '\[*\*]' | grep spp | awk -F '\\[*\*]' '{print $2}' | awk -F ":" '{print $1}' | sort -u >>
"$outfile"
echo "" >> "$outfile"
```

```
echo "What preprocessor events seen and how many" >> "$outfile"
cat data_alert.* | grep '\[*\*]' | grep spp | awk -F '\\[*\*]' '{print $2}' | awk -F ":" '{print $1}' | sort | uniq
-c | sort -r -n -k 1 >> "$outfile"
echo "" >> "$outfile"
```

```
echo "All non-spp events" >> "$outfile"
cat data_alert.* | grep '\[*\*]' | grep -v spp | awk -F '\\[*\*]' '{print $2}' | awk -F ":" '{print $1}' | sort -u
>> "$outfile"
echo "" >> "$outfile"
```

```
echo "All non-spp events and how many" >> "$outfile"
cat data_alert.* | grep '\[*\*]' | grep -v spp | awk -F '\\[*\*]' '{print $2}' | awk -F ":" '{print $1}' | sort |
uniq -c | sort -r -n -k 1 >> "$outfile"
echo "" >> "$outfile"
```

```
echo "" >> "$outfile"
echo "" >> "$outfile"
```

Script: wes.ipaddrs.sh

```
#!/bin/bash
```

```
#Variable declarations
outfile="wes.ipaddrs.results"
```

```
#Determine which files have the [*] pattern in them
echo "Which files have the [*] pattern in them" >> "$outfile"
grep -l '\[*\*]' data* >> "$outfile"
echo "" >> "$outfile"
```

```
#Determine which files DO NOT have the [*] pattern in them
echo "Which files DO NOT have the [*] pattern in them" >> "$outfile"
grep -L '\[*\*]' data* >> "$outfile"
echo "" >> "$outfile"
```

```
#Determine which files have the '->' pattern in them
echo "Which files have the -> pattern in them" >> "$outfile"
grep -l '->' data* >> "$outfile"
echo "" >> "$outfile"
```

```
#Determine which files DO NOT have the '->' pattern in them
echo "Which files DO NOT have the -> pattern in them" >> "$outfile"
grep -L '->' data* >> "$outfile"
echo "" >> "$outfile"
```

```
#All source addresses
```

```
echo "Top 100 source addresses by occurrence" >> "$outfile"
cat data* | grep '->' | awk -F '->' '{print $1}' | awk '{print $NF}' | awk -F ':' '{print $1}' | sort | uniq -c |
sort -r -n -k 1 | head --lines=100 >> "$outfile"
cat data* | grep '->' | awk -F '->' '{print $1}' | awk '{print $NF}' | awk -F ':' '{print $1}' | sort | uniq -c |
sort -r -n -k 1 >> wes.srcip.tally
echo "" >> "$outfile"
```

```
#All source ports
```

```
echo "Top 100 source ports by occurrence" >> "$outfile"
cat data* | grep '->' | awk -F '->' '{print $1}' | awk '{print $NF}' | awk -F ':' '{print $2}' | sort | uniq -c |
sort -r -n -k 1 | head --lines=100 >> "$outfile"
cat data* | grep '->' | awk -F '->' '{print $1}' | awk '{print $NF}' | awk -F ':' '{print $2}' | sort | uniq -c |
sort -r -n -k 1 >> wes.srcport.tally
echo "" >> "$outfile"
```

```
#All destination addresses
```

```
echo "Top 100 destination addresses by occurrence" >> "$outfile"
cat data* | grep '->' | awk -F '->' '{print $2}' | awk '{print $1}' | awk -F ':' '{print $1}' | sort | uniq -c | sort
-r -n -k 1 | head --lines=100 >> "$outfile"
cat data* | grep '->' | awk -F '->' '{print $2}' | awk '{print $1}' | awk -F ':' '{print $1}' | sort | uniq -c | sort
-r -n -k 1 >> wes.dstip.tally
echo "" >> "$outfile"
```

```
#All destination ports
```

```
echo "Top 100 destination ports by occurrence" >> "$outfile"
cat data* | grep '->' | awk -F '->' '{print $2}' | awk '{print $1}' | awk -F ':' '{print $2}' | sort | uniq -c | sort
-r -n -k 1 | head --lines=100 >> "$outfile"
cat data* | grep '->' | awk -F '->' '{print $2}' | awk '{print $1}' | awk -F ':' '{print $2}' | sort | uniq -c | sort
-r -n -k 1 >> wes.dstport.tally
echo "" >> "$outfile"
```

```
echo "" >> "$outfile"
echo "" >> "$outfile"
```

```
#Create file with all IP addresses, regardless of source or destination
```

```
cat data* | grep '->' | awk -F '->' '{print $1}' | awk '{print $NF}' | awk -F ':' '{print $1}' | sort | uniq >
/tmp/junkfile
cat data* | grep '->' | awk -F '->' '{print $2}' | awk '{print $1}' | awk -F ':' '{print $1}' | sort | uniq >>
/tmp/junkfile
cat /tmp/junkfile | sort | uniq -c | sort -r -n -k 1 > wes.allip.tally
rm -f /tmp/junkfile
```

```
echo "Total number of IP addresses observed" >> "$outfile"
cat wes.allip.tally | wc -l >> "$outfile"
echo "" >> "$outfile"
```

```
echo "Total number of IP addresses from MY.NET active/observed" >> "$outfile"
cat wes.allip.tally | grep 'MY\.NET' > wes.mynetip.tally
cat wes.mynetip.tally | wc -l >> "$outfile"
echo "" >> "$outfile"
```

```
echo "" >> "$outfile"
echo "" >> "$outfile"
```

Script: wes.oos.sh

```
#!/bin/bash
```

```
#Variable declarations
outfile="wes.oos.results"
```

```
#Create wes.oospairs files which contain all source and dest ip pairs both with and then without the
port numbers
```

```
for i in `bin/lis data_oos*` ; do cat "$i" | grep ' -> ' | awk '{print $2,$4}' ; done > wes.oospairs.ports
for i in `bin/lis data_oos*` ; do cat "$i" | grep ' -> ' | awk '{print $2,$4}' | sed s/^[0-9]*//g ; done >
wes.oospairs.addresses
```

```
#Source addresses
```

```
echo "Top 100 oos unique source ip addresses" >> "$outfile"
cat wes.oospairs.addresses | awk '{print $1}' | sort | uniq -c | sort -r -n -k 1 | head --lines=100 >>
"$outfile"
echo "" >> "$outfile"
```

```
#Source ports
```

```
echo "Top 100 oos unique source ports" >> "$outfile"
cat wes.oospairs.ports | awk '{print $1}' | sed s/.*:// | sort | uniq -c | sort -r -n -k 1 | head --lines=100 >>
"$outfile"
echo "" >> "$outfile"
```

```
#Destination addresses
```

```
echo "Top 100 oos unique destination ip addresses" >> "$outfile"
cat wes.oospairs.addresses | awk '{print $2}' | sort | uniq -c | sort -r -n -k 1 | head --lines=100 >>
"$outfile"
echo "" >> "$outfile"
```

```
#Destination ports
```

```
echo "Top 100 oos unique destination ports" >> "$outfile"
cat wes.oospairs.ports | awk '{print $2}' | sed s/.*:// | sort | uniq -c | sort -r -n -k 1 | head --lines=100 >>
"$outfile"
echo "" >> "$outfile"
echo "" >> "$outfile"
```



```
#All oos pairs
echo "Top 100 address pairs" >> "$outfile"
cat wes.oospairs.addresses | sort | uniq -c | sort -r -n -k 1 | head --lines=100 >> "$outfile"
echo "" >> "$outfile"

echo "Top 100 port pairs" >> "$outfile"
cat wes.oospairs.ports | sed s/\ .*:\ / | sed s/\ .*:\ / | sort | uniq -c | sort -r -n -k 1 | head --lines=100 >>
"$outfile"
echo "" >> "$outfile"

echo "" >> "$outfile"
echo "" >> "$outfile"
```

© SANS Institute 2000 - 2005, Author retains full rights.

Upcoming Training

Click Here to
{Get CERTIFIED!}



SANS Security East 2019	New Orleans, LA	Feb 02, 2019 - Feb 09, 2019	Live Event
Security East 2019 - SEC503: Intrusion Detection In-Depth	New Orleans, LA	Feb 04, 2019 - Feb 09, 2019	vLive
SANS Northern VA Spring- Tysons 2019	Tysons, VA	Feb 11, 2019 - Feb 16, 2019	Live Event
SANS London February 2019	London, United Kingdom	Feb 11, 2019 - Feb 16, 2019	Live Event
SANS New York Metro Winter 2019	Jersey City, NJ	Feb 18, 2019 - Feb 23, 2019	Live Event
SANS Scottsdale 2019	Scottsdale, AZ	Feb 18, 2019 - Feb 23, 2019	Live Event
SANS vLive - SEC503: Intrusion Detection In-Depth	SEC503 - 201902,	Feb 27, 2019 - Apr 04, 2019	vLive
SANS San Francisco Spring 2019	San Francisco, CA	Mar 11, 2019 - Mar 16, 2019	Live Event
SANS Madrid March 2019	Madrid, Spain	Mar 25, 2019 - Mar 30, 2019	Live Event
SANS 2019	Orlando, FL	Apr 01, 2019 - Apr 08, 2019	Live Event
Blue Team Summit & Training 2019	Louisville, KY	Apr 11, 2019 - Apr 18, 2019	Live Event
SANS Riyadh April 2019	Riyadh, Kingdom Of Saudi Arabia	Apr 13, 2019 - Apr 18, 2019	Live Event
Community SANS New York SEC503	New York, NY	Apr 29, 2019 - May 04, 2019	Community SANS
SANS Security West 2019	San Diego, CA	May 09, 2019 - May 16, 2019	Live Event
SANS Northern VA Spring- Reston 2019	Reston, VA	May 19, 2019 - May 24, 2019	Live Event
SANS Amsterdam May 2019	Amsterdam, Netherlands	May 20, 2019 - May 25, 2019	Live Event
SANS San Antonio 2019	San Antonio, TX	May 28, 2019 - Jun 02, 2019	Live Event
San Antonio 2019 - SEC503: Intrusion Detection In-Depth	San Antonio, TX	May 28, 2019 - Jun 02, 2019	vLive
SANS London June 2019	London, United Kingdom	Jun 03, 2019 - Jun 08, 2019	Live Event
SANSFIRE 2019	Washington, DC	Jun 15, 2019 - Jun 22, 2019	Live Event
Security Operations Summit & Training 2019	New Orleans, LA	Jun 24, 2019 - Jul 01, 2019	Live Event
SANS Paris July 2019	Paris, France	Jul 01, 2019 - Jul 06, 2019	Live Event
SANS Rocky Mountain 2019	Denver, CO	Jul 15, 2019 - Jul 20, 2019	Live Event
SANS Columbia 2019	Columbia, MD	Jul 15, 2019 - Jul 20, 2019	Live Event
SANS Boston Summer 2019	Boston, MA	Jul 29, 2019 - Aug 03, 2019	Live Event
SANS Chicago 2019	Chicago, IL	Aug 19, 2019 - Aug 24, 2019	Live Event
SANS Copenhagen August 2019	Copenhagen, Denmark	Aug 26, 2019 - Aug 31, 2019	Live Event
SANS Network Security 2019	Las Vegas, NV	Sep 09, 2019 - Sep 16, 2019	Live Event
SANS Oslo September 2019	Oslo, Norway	Sep 09, 2019 - Sep 14, 2019	Live Event
SANS London September 2019	London, United Kingdom	Sep 23, 2019 - Sep 28, 2019	Live Event
SANS OnDemand	Online	Anytime	Self Paced