Global Information Assurance Certification Paper

GCIA Practical for Spencer Allain

1 Table of Contents

2 Introduction

This document contains the 10 detects and analyses required for the GCIA certification practical. All included detects are of results found in the wild, meaning that no network traffic was artificially created. The only thing created in the lab was the evaluation of an attack traces.

Note that I now know that only 5 detects using a different format than my initial submission are required, so these have been re-ordered and re-formatted to fit the current requirements.

All destination addresses to the network that is being monitored have been sanitized. All external addresses have not been.

All junk from Exodus Communications, and things from Global Crossing and UUNET that have exactly the same signature were excluded from any of the traces. Now if somebody could just get all of their traffic excluded from the net as a whole.

The site being monitored is not heavily trafficked, and has some serious network and host defenses in place. As such, almost all of the detects are of the scanning nature, and not actual transmissions of data. Even the traditional cgi-script attacks are absent because of the nature of this site not having a high publicity. For this certification, it has been somewhat of a curse, but for day-to-day maintenance and monitoring it makes for easy filtering of anomalous behavior.

3 Detects

Name	Summary
1. Netbios-ns Scan	Netbios name query requests were sent to each externally visible IP address in ascending order.
2. SunRPC Probe	On two consecutive days, the same machine, from the same port, attempted to connect to the sunrpc portmapper port on one monitored machine.
3. False Netbus Scan	A single scan to port 12345 on a single host machine was detected.
4. Strange Finger	A finger attempt to a single machine being monitored that seems to have the TCP retry set to at least 7 minutes.
5. Remailer Attempt	An unrecognized host hit a low traffic mail server with a flurry of short connects, with very little actual data being transferred between the two systems. Further inspection revealed that the individual was trolling for mail servers that would relay mail from just about anywhere.
6. Source Port 20766 Reset	The only unsolicited traffic to one machine over two days was three resets from different machines, but all from port 20766.
7. Port 32773 Scan	A single machine scanned three monitored machines on port 32773 in less than a second. This is a very likely port for a SunRPC service to be running at.
8. Web Server Scan	The scan begins with 3 ICMP Echo Requests, and if a response comes back, then 3 requests are sent to port 80 (www) and 3 go to the socks port (1080).
9. Unsolicited Host Unreachable	One of the externally visible machines doesn't actually exist, as there is another machine that simply has an arp entry for it. Somehow that virtual machine was sent a host unreachable message.
10. Unsolicited Time Exceeded	The same IP address with just an arp entry, but no real machine listening, somehow received a time exceeded in transit from the @Home Network.

3.1 Detect 1

tcpdump output

```
16:07:40.894521 208.61.141.146.netbios-ns > 256.0.0.1.netbios-ns:
>>> NBT UDP PACKET(137): QUERY; REQUEST; BROADCAST
16:07:42.388994 208.61.141.146.netbios-ns > 256.0.0.1.netbios-ns:
>>> NBT UDP PACKET(137): QUERY; REQUEST; BROADCAST
16:07:43.888781 208.61.141.146.netbios-ns > 256.0.0.1.netbios-ns:
>>> NBT UDP PACKET(137): QUERY; REQUEST; BROADCAST
16:13:28.646185 208.61.141.146.netbios-ns > 256.0.0.2.netbios-ns:
>>> NBT UDP PACKET(137): QUERY; REQUEST; BROADCAST
16:13:30.140836 208.61.141.146.netbios-ns > 256.0.0.2.netbios-ns:
```

```
>>> NBT UDP PACKET(137): QUERY; REQUEST; BROADCAST
16:13:31.640623 208.61.141.146.netbios-ns > 256.0.0.2.netbios-ns:
>>> NBT UDP PACKET(137): QUERY; REQUEST; BROADCAST
16:14:10.827002 208.61.141.146.netbios-ns > 256.0.0.3.netbios-ns:
>>> NBT UDP PACKET(137): QUERY; REQUEST; BROADCAST
16:14:12.326081 208.61.141.146.netbios-ns > 256.0.0.3.netbios-ns:
>>> NBT UDP PACKET(137): QUERY; REQUEST; BROADCAST
16:14:13.825948 208.61.141.146.netbios-ns > 256.0.0.3.netbios-ns:
>>> NBT UDP PACKET(137): QUERY; REQUEST; BROADCAST
16:15:56.183539 208.61.141.146.netbios-ns > 256.0.0.4.netbios-ns:
>>> NBT UDP PACKET(137): QUERY; REQUEST; BROADCAST
16:15:57.680090 208.61.141.146.netbios-ns > 256.0.0.4.netbios-ns:
>>> NBT UDP PACKET(137): QUERY; REQUEST; BROADCAST
16:15:59.179780 208.61.141.146.netbios-ns > 256.0.0.4.netbios-ns:
>>> NBT UDP PACKET(137): QUERY; REQUEST; BROADCAST
16:19:37.513945 208.61.141.146.netbios-ns > 256.0.0.5.netbios-ns:
>>> NBT UDP PACKET(137): QUERY; REQUEST; BROADCAST
16:19:39.008445 208.61.141.146.netbios-ns > 256.0.0.5.netbios-ns:
>>> NBT UDP PACKET(137): QUERY; REQUEST; BROADCAST
16:19:40.508297 208.61.141.146.netbios-ns > 256.0.0.5.netbios-ns:
>>> NBT UDP PACKET(137): QUERY; REQUEST; BROADCAST
16:20:09.169070 208.61.141.146.netbios-ns > 256.0.0.6.netbios-ns:
>>> NBT UDP PACKET(137): QUERY; REQUEST; BROADCAST
16:20:10.666034 208.61.141.146.netbios-ns > 256.0.0.6.netbios-ns:
>>> NBT UDP PACKET(137): QUERY; REQUEST; BROADCAST
16:20:12.165320 208.61.141.146.netbios-ns > 256.0.0.6.netbios-ns:
>>> NBT UDP PACKET(137): QUERY; REQUEST; BROADCAST
16:20:19.694216 208.61.141.146.netbios-ns > 256.0.0.7.netbios-ns:
>>> NBT UDP PACKET(137): QUERY; REQUEST; BROADCAST
16:20:21.188554 208.61.141.146.netbios-ns > 256.0.0.7.netbios-ns:
>>> NBT UDP PACKET(137): QUERY; REQUEST; BROADCAST
16:20:22.688403 208.61.141.146.netbios-ns > 256.0.0.7.netbios-ns:
>>> NBT UDP PACKET(137): QUERY; REQUEST; BROADCAST
16:27:34.811119 208.61.141.146.netbios-ns > 256.0.0.8.netbios-ns:
>>> NBT UDP PACKET(137): QUERY; REQUEST; BROADCAST
16:27:36.310635 208.61.141.146.netbios-ns > 256.0.0.8.netbios-ns:
>>> NBT UDP PACKET(137): QUERY; REQUEST; BROADCAST
16:27:37.810364 208.61.141.146.netbios-ns > 256.0.0.8.netbios-ns:
>>> NBT UDP PACKET(137): QUERY; REQUEST; BROADCAST
16:29:57.841960 208.61.141.146.netbios-ns > 256.0.0.9.netbios-ns:
>>> NBT UDP PACKET(137): QUERY; REQUEST; BROADCAST
16:29:59.341272 208.61.141.146.netbios-ns > 256.0.0.9.netbios-ns:
>>> NBT UDP PACKET(137): QUERY; REQUEST; BROADCAST
16:30:00.840935 208.61.141.146.netbios-ns > 256.0.0.9.netbios-ns:
>>> NBT UDP PACKET(137): QUERY; REQUEST; BROADCAST
16:35:04.015054 208.61.141.146.netbios-ns > 256.0.0.10.netbios-ns:
>>> NBT UDP PACKET(137): QUERY; REQUEST; BROADCAST
16:35:05.512051 208.61.141.146.netbios-ns > 256.0.0.10.netbios-ns:
>>> NBT UDP PACKET(137): QUERY; REQUEST; BROADCAST
16:35:07.011738 208.61.141.146.netbios-ns > 256.0.0.10.netbios-ns:
>>> NBT UDP PACKET(137): QUERY; REQUEST; BROADCAST
16:35:14.550656 208.61.141.146.netbios-ns > 256.0.0.11.netbios-ns:
>>> NBT UDP PACKET(137): QUERY; REQUEST; BROADCAST
16:35:16.048422 208.61.141.146.netbios-ns > 256.0.0.11.netbios-ns:
>>> NBT UDP PACKET(137): QUERY; REQUEST; BROADCAST
16:35:17.548023 208.61.141.146.netbios-ns > 256.0.0.11.netbios-ns:
>>> NBT UDP PACKET(137): QUERY; REQUEST; BROADCAST
```

snort output

```
[**] IDS177/netbios-name-query [**]
07/20-16:07:40.894521 208.61.141.146:137 -> 256.0.0.1:137
UDP TTL:112 TOS:0x0 ID:22685
Len: 58

[**] IDS177/netbios-name-query [**]
07/20-16:07:42.388994 208.61.141.146:137 -> 256.0.0.1:137
UDP TTL:112 TOS:0x0 ID:23197
Len: 58

[**] IDS177/netbios-name-query [**]
07/20-16:07:43.888781 208.61.141.146:137 -> 256.0.0.1:137
UDP TTL:112 TOS:0x0 ID:23709
Len: 58

[**] IDS177/netbios-name-query [**]
07/20-16:13:28.646185 208.61.141.146:137 -> 256.0.0.2:137
UDP TTL:112 TOS:0x0 ID:41886
Len: 58

[**] IDS177/netbios-name-query [**]
07/20-16:13:30.140836 208.61.141.146:137 -> 256.0.0.2:137
UDP TTL:112 TOS:0x0 ID:42398
Len: 58

[**] IDS177/netbios-name-query [**]
07/20-16:13:31.640623 208.61.141.146:137 -> 256.0.0.2:137
UDP TTL:112 TOS:0x0 ID:42910
Len: 58

[**] IDS177/netbios-name-query [**]
07/20-16:14:10.827002 208.61.141.146:137 -> 256.0.0.3:137
UDP TTL:112 TOS:0x0 ID:52126
Len: 58

[**] IDS177/netbios-name-query [**]
07/20-16:14:12.326081 208.61.141.146:137 -> 256.0.0.3:137
UDP TTL:112 TOS:0x0 ID:52638
Len: 58

[**] IDS177/netbios-name-query [**]
07/20-16:14:13.825948 208.61.141.146:137 -> 256.0.0.3:137
UDP TTL:112 TOS:0x0 ID:53150
Len: 58

[**] IDS177/netbios-name-query [**]
07/20-16:15:56.183539 208.61.141.146:137 -> 256.0.0.4:137
UDP TTL:112 TOS:0x0 ID:13215
```

```
Len: 58

[**] IDS177/netbios-name-query [**]
07/20-16:15:57.680090 208.61.141.146:137 -> 256.0.0.4:137
UDP TTL:112 TOS:0x0 ID:13727
Len: 58

[**] IDS177/netbios-name-query [**]
07/20-16:15:59.179780 208.61.141.146:137 -> 256.0.0.4:137
UDP TTL:112 TOS:0x0 ID:14239
Len: 58

[**] IDS177/netbios-name-query [**]
07/20-16:19:37.513945 208.61.141.146:137 -> 256.0.0.5:137
UDP TTL:112 TOS:0x0 ID:1440
Len: 58

[**] IDS177/netbios-name-query [**]
07/20-16:19:39.008445 208.61.141.146:137 -> 256.0.0.5:137
UDP TTL:112 TOS:0x0 ID:1952
Len: 58

[**] IDS177/netbios-name-query [**]
07/20-16:19:40.508297 208.61.141.146:137 -> 256.0.0.5:137
UDP TTL:112 TOS:0x0 ID:2464
Len: 58

[**] IDS177/netbios-name-query [**]
07/20-16:20:09.169070 208.61.141.146:137 -> 256.0.0.6:137
UDP TTL:112 TOS:0x0 ID:9120
Len: 58

[**] IDS177/netbios-name-query [**]
07/20-16:20:10.666034 208.61.141.146:137 -> 256.0.0.6:137
UDP TTL:112 TOS:0x0 ID:9632
Len: 58

[**] IDS177/netbios-name-query [**]
07/20-16:20:12.165320 208.61.141.146:137 -> 256.0.0.6:137
UDP TTL:112 TOS:0x0 ID:10144
Len: 58

[**] IDS177/netbios-name-query [**]
07/20-16:20:19.694216 208.61.141.146:137 -> 256.0.0.7:137
UDP TTL:112 TOS:0x0 ID:11680
Len: 58

[**] IDS177/netbios-name-query [**]
07/20-16:20:21.188554 208.61.141.146:137 -> 256.0.0.7:137
UDP TTL:112 TOS:0x0 ID:12192
Len: 58

[**] IDS177/netbios-name-query [**]
07/20-16:20:22.688403 208.61.141.146:137 -> 256.0.0.7:137
UDP TTL:112 TOS:0x0 ID:12704
Len: 58

[**] IDS177/netbios-name-query [**]
07/20-16:25:28.344170 208.61.141.146:137 -> 256.0.0.200:137
UDP TTL:112 TOS:0x0 ID:21921
Len: 58

[**] IDS177/netbios-name-query [**]
07/20-16:25:29.838887 208.61.141.146:137 -> 256.0.0.200:137
UDP TTL:112 TOS:0x0 ID:22433
Len: 58

[**] IDS177/netbios-name-query [**]
07/20-16:25:31.338794 208.61.141.146:137 -> 256.0.0.200:137
UDP TTL:112 TOS:0x0 ID:22945
Len: 58

[**] IDS177/netbios-name-query [**]
07/20-16:27:34.811119 208.61.141.146:137 -> 256.0.0.8:137
UDP TTL:112 TOS:0x0 ID:53665
Len: 58

[**] IDS177/netbios-name-query [**]
07/20-16:27:36.310635 208.61.141.146:137 -> 256.0.0.8:137
UDP TTL:112 TOS:0x0 ID:54177
Len: 58

[**] IDS177/netbios-name-query [**]
07/20-16:27:37.810364 208.61.141.146:137 -> 256.0.0.8:137
UDP TTL:112 TOS:0x0 ID:54689
Len: 58

[**] IDS177/netbios-name-query [**]
07/20-16:29:57.841960 208.61.141.146:137 -> 256.0.0.9:137
UDP TTL:112 TOS:0x0 ID:22178
Len: 58

[**] IDS177/netbios-name-query [**]
07/20-16:29:59.341272 208.61.141.146:137 -> 256.0.0.9:137
UDP TTL:112 TOS:0x0 ID:22690
Len: 58

[**] IDS177/netbios-name-query [**]
07/20-16:30:00.840935 208.61.141.146:137 -> 256.0.0.9:137
UDP TTL:112 TOS:0x0 ID:23458
Len: 58

[**] IDS177/netbios-name-query [**]
07/20-16:35:04.015054 208.61.141.146:137 -> 256.0.0.10:137
UDP TTL:112 TOS:0x0 ID:54947
Len: 58

[**] IDS177/netbios-name-query [**]
07/20-16:35:05.512051 208.61.141.146:137 -> 256.0.0.10:137
UDP TTL:112 TOS:0x0 ID:55459
Len: 58
```

```
[**] IDS177/netbios-name-query [**]
07/20-16:35:07.011738 208.61.141.146:137 -> 256.0.0.10:137
UDP TTL:112 TOS:0x0 ID:55971
Len: 58

[**] IDS177/netbios-name-query [**]
07/20-16:35:14.550656 208.61.141.146:137 -> 256.0.0.11:137
UDP TTL:112 TOS:0x0 ID:57763
Len: 58

[**] IDS177/netbios-name-query [**]
07/20-16:35:16.048422 208.61.141.146:137 -> 256.0.0.11:137
UDP TTL:112 TOS:0x0 ID:58275
Len: 58

[**] IDS177/netbios-name-query [**]
07/20-16:35:17.548023 208.61.141.146:137 -> 256.0.0.11:137
UDP TTL:112 TOS:0x0 ID:58787
Len: 58
```

3.1.1 Source of trace

A monitored network where the sensor is located between all external firewalls and the router to the ISP.

3.1.2 Detect was generated by

Detected by both tcpdump and snort filters. tcpdump has a filter to check for any Netbios traffic since it should never be passing outside of any of the firewalls. Snort detected the fact that it was a name query request.

tcpdump format

```
hh:mm:ss.SSSSSS Source.SrcPort > Destination.DstPort: ProtocolSpecificData
```

snort format

```
[**] ShortIdentifier/VerboseIdentifier [**]
MM/DD-hh:mm:ss.SSSSSS Source:SrcPort -> Destination:DstPort
Protocol TTL:TimeToLive TOS:TypeOfService ID:IP_Identifier
Len: LengthOfProtocolDataInBytes
```

3.1.3 Probability the source address was spoofed

It is almost certain that the source address is not spoofed, as the person is probing for information.

3.1.4 Description of attack

The entire IP range of machines is scanned in ascending order on UDP port 137 precisely 3 times from the same source address, also using source port 137.

3.1.5 Attack mechanism

This is less of an attack, but more of a pre-cursor for one. The person is doing reconnaissance, and in a very noisy fashion also.

3.1.6 Correlations

This points to a mass UDP scan discussion, and the important context is that there apparently is no known standard Windows software that scans an entire network range on UDP port 137, as was done on the network being monitored.

http://archives.neohapsis.com/archives/incidents/2000-05/0054.html

3.1.7 Evidence of active targeting

This was a general scan of the entire network.

3.1.8 Severity

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

(3 + 5) - (5 + 3) = 0

System Criticality	3	The only machines that could possibly respond to such a query are only semi-critical machines.
Attack Lethality	5	NetBIOS frightens me. It gives out so much information, and lets you do so many things with minimal or no security restrictions that I always consider those kinds of attacks or scans potentially lethal.
Network Countermeasures	5	All Netbios traffic is blocked in all directions, in and out.
System Countermeasures	3	Since it is scanning a variety of machines, the lowest score will cover all machines. Since all of the potentially vulnerable machines are patched very well (and none are running SAMBA) there is little they could except maybe get some information.

3.1.9 Defensive recommendation

Unless absolutely necessary, all Netbios ports 135-139 and 445 for Windows 2000 should be blocked incoming and outgoing from all firewalls. Even if some have to be left open, make sure that only the allowed traffic between trusted sites is allowed (if you don't have a VPN) to at least limit the chance of attack.

3.1.10 Multiple choice test question

```
16:07:40.894521 208.61.141.146.netbios-ns > 256.0.0.1.netbios-ns:
>>> NBT UDP PACKET(137): QUERY; REQUEST; BROADCAST
16:07:42.388994 208.61.141.146.netbios-ns > 256.0.0.1.netbios-ns:
>>> NBT UDP PACKET(137): QUERY; REQUEST; BROADCAST
16:07:43.888781 208.61.141.146.netbios-ns > 256.0.0.1.netbios-ns:
>>> NBT UDP PACKET(137): QUERY; REQUEST; BROADCAST
16:13:28.646185 208.61.141.146.netbios-ns > 256.0.0.2.netbios-ns:
>>> NBT UDP PACKET(137): QUERY; REQUEST; BROADCAST
16:13:30.140836 208.61.141.146.netbios-ns > 256.0.0.2.netbios-ns:
>>> NBT UDP PACKET(137): QUERY; REQUEST; BROADCAST
16:13:31.640623 208.61.141.146.netbios-ns > 256.0.0.2.netbios-ns:
>>> NBT UDP PACKET(137): QUERY; REQUEST; BROADCAST

The attacker is most likely:

     a) conducting a denial of service attack.
     b) attempting to directly accessing a file share on a Windows machine.
     c) a misconfigured machine looking for its domain server.
     d) probing to find information about Windows machines.

answer: d
```

Start of Detect TOC

3.2 Detect 2

tcpdump output day 1

```
19:19:25.645787 207.8.127.251.1023 > 256.0.0.1.sunrpc:  udp 84
19:19:25.894351 207.8.127.251.1023 > 256.0.0.1.sunrpc:  udp 84
19:19:26.406348 207.8.127.251.1023 > 256.0.0.1.sunrpc:  udp 84
19:19:27.403450 207.8.127.251.1023 > 256.0.0.1.sunrpc:  udp 84
19:19:28.404856 207.8.127.251.1023 > 256.0.0.1.sunrpc:  udp 84
19:19:30.403533 207.8.127.251.1023 > 256.0.0.1.sunrpc:  udp 84
19:19:34.394810 207.8.127.251.1023 > 256.0.0.1.sunrpc:  udp 84
```

tcpdump output day 2

```
20:38:57.328466 207.8.127.169.1023 > 256.0.0.1.sunrpc:  udp 84
20:38:57.576253 207.8.127.169.1023 > 256.0.0.1.sunrpc:  udp 84
20:38:58.076211 207.8.127.169.1023 > 256.0.0.1.sunrpc:  udp 84
20:38:59.096665 207.8.127.169.1023 > 256.0.0.1.sunrpc:  udp 84
20:39:00.075754 207.8.127.169.1023 > 256.0.0.1.sunrpc:  udp 84
20:39:02.077393 207.8.127.169.1023 > 256.0.0.1.sunrpc:  udp 84
20:39:06.096211 207.8.127.169.1023 > 256.0.0.1.sunrpc:  udp 84
20:39:10.086466 207.8.127.169.1023 > 256.0.0.1.sunrpc:  udp 84
20:39:18.077254 207.8.127.169.1023 > 256.0.0.1.sunrpc:  udp 84
20:39:34.118056 207.8.127.169.1023 > 256.0.0.1.sunrpc:  udp 84
20:39:50.119161 207.8.127.169.1023 > 256.0.0.1.sunrpc:  udp 84
```

3.2.1 Source of trace

A monitored network where the sensor is located between all external firewalls and the router to the ISP.

3.2.2 Detect was generated by

Detected by a tcpdump filter to check for any incoming requests directly to the portmapper port, meaning port 111.

tcpdump format

```
hh:mm:ss.SSSSSS Source.SrcPort > Destination.DstPort: ProtocolSpecificData
```

3.2.3 Probability the source address was spoofed

Unlikely, as the person is looking for information, and unless they have control of a router between the monitored site and the source site, it must be the real IP. Furthermore, it is coming from two distinct addresses within the same range of IPs for an ISP dialup connection, so they probably are not too worried about anyone seeing their current IP address.

3.2.4 Description of attack

An individual Solaris machine was scanned on the portmapper port over two separate days. Each time the probe used the same source port (a reserved port for a Unix system), and the IP addresses both fell within those assigned to an ISP in Texas.

3.2.5 Attack mechanism

The probe of a portmapper is to discover what ports potentially vulnerable RPC applications are running at. A site that leaves their portmapper port open to the world is a very good target, as they probably have left the other ports open as well.

3.2.6 Correlations

UDP scans to port 111 are so common, that it is pretty much outright recommended that such access be denied under all circumstances http://pandonia.canberra.edu.au/ClientServer/week3/security.sgml-039.html.

There are even pages dating back to 1997 about alternatives to simply scanning on port 111, since it was such a common process that many people had started blocking it at their routers http://www.infowar.com/iwftp/phrack/Phrack51/P51-10.txt

3.2.7 Evidence of active targeting

Precisely one machine was targeted, and indeed at one time in the distant past, this machine did indeed reside unprotected on the internet and was running an old version of the Solaris operating system.

3.2.8 Severity

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

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

System Criticality	5	The machine targeted is very important, and it would be very bad if it were to be compromised.
Attack Lethality	4	While it was not successful, the attack was targeted at a Solaris machine, and no other machines were scanned at all.
Network Countermeasures	5	All unsolicited packets to port 111 are blocked by a stateful firewall.
System Countermeasures	2	Even though the machine has many patches applied, it still is running a fairly old version of the Solaris operating system.

3.2.9 Defensive recommendation

Block all requests to the portmapper port at a firewall or packet filtering router.

3.2.10 Multiple choice test question

```
Day 1
19:19:25.645787 207.8.127.251.1023 > 256.0.0.1.sunrpc:  udp 84
19:19:25.894351 207.8.127.251.1023 > 256.0.0.1.sunrpc:  udp 84

Day 2
20:38:57.328466 207.8.127.169.1023 > 256.0.0.1.sunrpc:  udp 84
20:38:57.576253 207.8.127.169.1023 > 256.0.0.1.sunrpc:  udp 84


This is probably evidence of a:

   a) concerted denial of service attack.
   b) scan done by the same person.
   c) scan from spoofed IP addresses.
   d) portmapper buffer overflow attempt.

answer: b
```

Start of Detect TOC

3.3 Detect 3

tcpdump output

```
07:40:51.892573 216.223.94.152.3272 > 256.0.0.1.12345:  udp 19
0x0000   4500 002f 10ed 0000 6e11 74df d8df 5e98        E../....n.t...^.
0x0010   XXXX XXXX 0cc8 3039 001b 526a ce63 d1d2        ......09..Rj.c..
0x0020   16e7 13cf 38a5 a586 b275 4b99 aa32 58          ....8....uK..2X
```

3.3.1 Source of trace

A monitored network where the sensor is located between all external firewalls and the router to the ISP.

3.3.2 Detect was generated by

A tcpdump filter that looks for traffic specifically to externally visible machines, and that excludes all traffic allowed in by the specified security policies. In other words, all unexpected traffic is caught.

tcpdump format

```
hh:mm:ss.SSSSSS Source.SrcPort > Destination.DstPort: ProtocolSpecificData
```

3.3.3 Probability the source address was spoofed

Not completely sure, but if the person was really looking for a trojan, they would want to be able to get a response, so I suspect that the source addresses is not spoofed..

3.3.4 Description of attack

A single UDP request to port 12345 was sent to a Windows machine from a host located at a Hydro-Electric Commission in Canada.

3.3.5 Attack mechanism

This is a scan for a process listening to UDP port 12345.

3.3.6 Correlations

This is very interesting as Netbus and variants are known to listen on TCP port 12345, but I have be unable to find anything that normally listens on UDP port 12345. My guess is that this is a scan new hoping to get past sensors that allow access to port 12345 without regard to the protocol. I do not recognize the hex dump output, and neither does any snort rules that I could dig up. Hopefully somebody with more experience can examine this packet in more detail.

3.3.7 Evidence of active targeting

The only machine that was targeted, was one of the few Windows machines that is externally visible.

3.3.8 Severity

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

(3 + 5) - (5 + 2) = 1

System Criticality	3	The machine targeted is somewhat important, but it is not a critical machine.
Attack Lethality	5	Without any knowledge of how lethal the attack could be, I've given it the highest rating.
Network Countermeasures	5	All unsolicited packets to dynamic ports are blocked by a stateful firewall.
System Countermeasures	2	The machine is somewhat hardened, but there is no telling whether a trojan could be placed onto it or not.

3.3.9 Defensive recommendation

Block all unsolicited traffic to all ports except for those absolutely necessary, like DNS and HTTP.

3.3.10 Multiple choice test question

```
07:40:51.892573 216.223.94.152.3272 > 256.0.0.1.12345:  udp 19
0x0000   4500 002f 10ed 0000 6e11 74df d8df 5e98        E../....n.t...^.
0x0010   XXXX XXXX 0cc8 3039 001b 526a ce63 d1d2        ......09..Rj.c..
0x0020   16e7 13cf 38a5 a586 b275 4b99 aa32 58          ....8....uK..2X

This is definitely NOT:

   a) a Netbus scan.
   b) a packet with a spoofed IP address.
   c) a BackOrifice scan.
   d) a UDP packet with payload of size 19 bytes.

answer: a
```

This question is evil because Netbus only operates using TCP, but some sites just label the interesting port as 12345; it could have a spoofed IP address; BackOrifice could be running at that port and uses UDP; and the payload is definitely 19 bytes, but it wastes time checking for sure that 19 doesn't mean the size of the whole UDP portion of the packet.

Start of Detect TOC

3.4 Detect 4

tcpdump output

```
14:53:18.706511 131.215.103.94.1189 > 256.0.0.1.finger: S 784278959:784278959(0) win 32120 <mss 1460,sackOK,timestamp 418650[|tcp]> (DF)
14:53:21.677656 131.215.103.94.1189 > 256.0.0.1.finger: S 784278959:784278959(0) win 32120 <mss 1460,sackOK,timestamp 418950[|tcp]> (DF)
14:53:27.678870 131.215.103.94.1189 > 256.0.0.1.finger: S 784278959:784278959(0) win 32120 <mss 1460,sackOK,timestamp 419550[|tcp]> (DF)
14:53:39.680571 131.215.103.94.1189 > 256.0.0.1.finger: S 784278959:784278959(0) win 32120 <mss 1460,sackOK,timestamp 420750[|tcp]> (DF)
14:54:03.683702 131.215.103.94.1189 > 256.0.0.1.finger: S 784278959:784278959(0) win 32120 <mss 1460,sackOK,timestamp 423150[|tcp]> (DF)
14:54:51.685735 131.215.103.94.1189 > 256.0.0.1.finger: S 784278959:784278959(0) win 32120 <mss 1460,sackOK,timestamp 427950[|tcp]> (DF)
14:56:27.685218 131.215.103.94.1189 > 256.0.0.1.finger: S 784278959:784278959(0) win 32120 <mss 1460,sackOK,timestamp 437550[|tcp]> (DF)
14:58:27.689330 131.215.103.94.1189 > 256.0.0.1.finger: S 784278959:784278959(0) win 32120 <mss 1460,sackOK,timestamp 449550[|tcp]> (DF)
15:00:27.699089 131.215.103.94.1189 > 256.0.0.1.finger: S 784278959:784278959(0) win 32120 <mss 1460,sackOK,timestamp 461550[|tcp]> (DF)
```

3.4.1 Source of trace

A monitored network where the sensor is located between all external firewalls and the router to the ISP.

3.4.2 Detect was generated by

A tcpdump filter that looks for traffic specifically to externally visible machines, and that excludes all traffic allowed in by the specified security policies. In other words, all unexpected traffic is caught.

This site doesn't like its machines being fingered by the outside world.

tcpdump format

```
hh:mm:ss.SSSSSS Source.SrcPort > Destination.DstPort: ProtocolSpecificData
```

3.4.3 Probability the source address was spoofed

It's a TCP connection, in order for the handshake to complete, the address must not be spoofed.

3.4.4 Description of attack

This is a SYN scan to see if the finger port will respond. If it can complete the 3-way handshake, then it can request all kinds of useful information about the accounts on the machine.

3.4.5 Attack mechanism

Attempt to determine as much information about the machine as possible because finger likes to give out tons of information useful to a hacker.

3.4.6 Correlations

There a few exploits associated with finger http://www.cotse.com/exploits/netapps/finger/, but mostly it is used to find information about valid user names on the system and other system information.

This is a very strange finger request, and I've not found anything on the web that quite corresponds to it, as it actually looks like a valid tcp retry, but with an abnormally long retry limit.

The retry rate interval goes from 3 seconds to 6, to 12, to 24, to 48, to 96, and then instead of jumping to 192 it stabilizes at 120 seconds, or exactly two minutes. This is a very distinguishing characteristic, and I wish I could know for certain whether it is the operating system, or some hacker program providing such predictable behavior.

3.4.7 Evidence of active targeting

The single machine targeted used to be a very popular machine, just over 3 years ago, and it is a Solaris box.

3.4.8 Severity

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

(5 + 1) - (5 + 4) = -3

System Criticality	5	The machine targeted is quite important, and it would be noticed quickly if operations were disrupted.
Attack Lethality	1	This is about as soft a probe one could get, unless a trojan were somehow able to be installed and allowed to run at the reserved finger port.
Network Countermeasures	5	All unsolicited packets to the finger port are blocked by a stateful firewall.
System Countermeasures	4	The machine has been configured to not respond to finger requests from anywhere.

3.4.9 Defensive recommendation

Block all incoming requests to the finger port at either a firewall or packet filtering router.

3.4.10 Multiple choice test question

```
14:53:18.706511 131.215.103.94.1189 > 256.0.0.1.finger: S 784278959:784278959(0) win 32120 <mss 1460,sackOK,timestamp 418650[|tcp]> (DF)
14:53:21.677656 131.215.103.94.1189 > 256.0.0.1.finger: S 784278959:784278959(0) win 32120 <mss 1460,sackOK,timestamp 418950[|tcp]> (DF)
14:53:27.678870 131.215.103.94.1189 > 256.0.0.1.finger: S 784278959:784278959(0) win 32120 <mss 1460,sackOK,timestamp 419550[|tcp]> (DF)
14:53:39.680571 131.215.103.94.1189 > 256.0.0.1.finger: S 784278959:784278959(0) win 32120 <mss 1460,sackOK,timestamp 420750[|tcp]> (DF)
14:54:03.683702 131.215.103.94.1189 > 256.0.0.1.finger: S 784278959:784278959(0) win 32120 <mss 1460,sackOK,timestamp 423150[|tcp]> (DF)
14:54:51.685735 131.215.103.94.1189 > 256.0.0.1.finger: S 784278959:784278959(0) win 32120 <mss 1460,sackOK,timestamp 427950[|tcp]> (DF)
14:56:27.685218 131.215.103.94.1189 > 256.0.0.1.finger: S 784278959:784278959(0) win 32120 <mss 1460,sackOK,timestamp 437550[|tcp]> (DF)
14:58:27.689330 131.215.103.94.1189 > 256.0.0.1.finger: S 784278959:784278959(0) win 32120 <mss 1460,sackOK,timestamp 449550[|tcp]> (DF)
15:00:27.699089 131.215.103.94.1189 > 256.0.0.1.finger: S 784278959:784278959(0) win 32120 <mss 1460,sackOK,timestamp 461550[|tcp]> (DF)

What is abnormal about this scan:

   a) the source port is repeated.
   b) the destination port is repeated.
   c) the length of time between the first packet and the last packet.
   d) the sequence number is repeated.

answer: d
```

While this looks totally bogus, aside from the fact that the TCP retry limit is extremely long, it is hard to tell if the packets were crafted or a non-normal operating system is being used. I'd like to note that this finger originates from a CalTech machine.

Start of Detect TOC

3.5 Detect 5

tcpdump output

```
18:22:04.723696 194.178.232.55.2488 > mailhost.smtp: S 4132059863:4132059863(0) win 32120 <mss 1460,sackOK,timestamp 2921939[|tcp]> (DF)
18:22:04.726714 mailhost.smtp > 194.178.232.55.2488: S 448162290:448162290(0) ack 4132059864 win 32120 <mss 1460,sackOK,timestamp 183739012[|tcp]> (DF)
18:22:04.827885 194.178.232.55.2488 > mailhost.smtp: . ack 1 win 32120 <nop,nop,timestamp 2921950 183739012> (DF)
18:22:30.631963 194.178.232.55.2630 > mailhost.smtp: S 4155444067:4155444067(0) win 32120 <mss 1460,sackOK,timestamp 2924530[|tcp]> (DF)
18:22:30.632928 mailhost.smtp > 194.178.232.55.2630: S 474827743:474827743(0) ack 4155444068 win 32120 <mss 1460,sackOK,timestamp 183741603[|tcp]> (DF)
18:22:30.729484 194.178.232.55.2630 > mailhost.smtp: . ack 1 win 32120 <nop,nop,timestamp 2924540 183741603> (DF)
18:22:35.132042 mailhost.smtp > 194.178.232.55.2488: P 1:88(87) ack 1 win 32120 <nop,nop,timestamp 183742053 2921950> (DF)
18:22:35.229776 194.178.232.55.2488 > mailhost.smtp: . ack 88 win 32120 <nop,nop,timestamp 2924990 183742053> (DF)
18:22:35.230284 194.178.232.55.2488 > mailhost.smtp: P 1:34(33) ack 88 win 32120 <nop,nop,timestamp 2924990 183742053> (DF)
18:22:35.230621 mailhost.smtp > 194.178.232.55.2488: . ack 34 win 32120 <nop,nop,timestamp 183742063 2924990> (DF)
18:22:35.230851 mailhost.smtp > 194.178.232.55.2488: P 88:186(98) ack 34 win 32120 <nop,nop,timestamp 183742063 2924990> (DF)
18:22:35.327768 194.178.232.55.2488 > mailhost.smtp: P 34:63(29) ack 186 win 32120 <nop,nop,timestamp 2925000 183742063> (DF)
18:22:35.339609 mailhost.smtp > 194.178.232.55.2488: . ack 63 win 32120 <nop,nop,timestamp 183742074 2925000> (DF)
18:22:35.548410 mailhost.smtp > 194.178.232.55.2488: P 186:222(36) ack 63 win 32120 <nop,nop,timestamp 183742094 2925000> (DF)
18:22:35.645080 194.178.232.55.2488 > mailhost.smtp: P 63:104(41) ack 222 win 32120 <nop,nop,timestamp 2925031 183742094> (DF)
18:22:35.659622 mailhost.smtp > 194.178.232.55.2488: . ack 104 win 32120 <nop,nop,timestamp 183742106 2925031> (DF)
18:22:35.938182 mailhost.smtp > 194.178.232.55.2488: P 222:278(56) ack 104 win 32120 <nop,nop,timestamp 183742133 2925031> (DF)
18:22:36.035017 194.178.232.55.2488 > mailhost.smtp: F 104:104(0) ack 278 win 32120 <nop,nop,timestamp 2925070 183742133> (DF)
18:22:36.035405 mailhost.smtp > 194.178.232.55.2488: . ack 105 win 32120 <nop,nop,timestamp 183742143 2925070> (DF)
18:22:36.040590 mailhost.smtp > 194.178.232.55.2488: F 278:278(0) ack 105 win 32120 <nop,nop,timestamp 183742144 2925070> (DF)
18:22:36.136738 194.178.232.55.2488 > mailhost.smtp: . ack 279 win 32120 <nop,nop,timestamp 2925080 183742144> (DF)
18:23:00.632928 mailhost.smtp > 194.178.232.55.2630: P 1:88(87) ack 1 win 32120 <nop,nop,timestamp 183744603 2924540> (DF)
18:23:00.729559 194.178.232.55.2630 > mailhost.smtp: . ack 88 win 32120 <nop,nop,timestamp 2927540 183744603> (DF)
18:23:00.730495 194.178.232.55.2630 > mailhost.smtp: P 1:34(33) ack 88 win 32120 <nop,nop,timestamp 2927540 183744603> (DF)
18:23:00.730837 mailhost.smtp > 194.178.232.55.2630: . ack 34 win 32120 <nop,nop,timestamp 183744613 2927540> (DF)
18:23:00.731044 mailhost.smtp > 194.178.232.55.2630: P 88:186(98) ack 34 win 32120 <nop,nop,timestamp 183744613 2927540> (DF)
18:23:00.828137 194.178.232.55.2630 > mailhost.smtp: P 34:63(29) ack 186 win 32120 <nop,nop,timestamp 2927550 183744613> (DF)
18:23:00.833515 mailhost.smtp > 194.178.232.55.2630: P 186:222(36) ack 63 win 32120 <nop,nop,timestamp 183744623 2927550> (DF)
18:23:00.931045 194.178.232.55.2630 > mailhost.smtp: P 63:106(43) ack 222 win 32120 <nop,nop,timestamp 2927560 183744623> (DF)
18:23:00.934431 mailhost.smtp > 194.178.232.55.2630: P 222:280(58) ack 106 win 32120 <nop,nop,timestamp 183744633 2927560> (DF)
18:23:01.031827 194.178.232.55.2630 > mailhost.smtp: F 106:106(0) ack 280 win 32120 <nop,nop,timestamp 2927570 183744633> (DF)
18:23:01.032202 mailhost.smtp > 194.178.232.55.2630: . ack 107 win 32120 <nop,nop,timestamp 183744643 2927570> (DF)
18:23:01.037453 mailhost.smtp > 194.178.232.55.2630: F 280:280(0) ack 107 win 32120 <nop,nop,timestamp 183744643 2927570> (DF)
18:23:01.133444 194.178.232.55.2630 > mailhost.smtp: . ack 281 win 32120 <nop,nop,timestamp 2927580 183744643> (DF)
18:24:00.952615 194.178.232.55.3017 > mailhost.smtp: S 4247532536:4247532536(0) win 32120 <mss 1460,sackOK,timestamp 2933562[|tcp]> (DF)
18:24:00.953679 mailhost.smtp > 194.178.232.55.3017: S 558019630:558019630(0) ack 4247532537 win 32120 <mss 1460,sackOK,timestamp 183750635[|tcp]> (DF)
18:24:01.049432 194.178.232.55.3017 > mailhost.smtp: . ack 1 win 32120 <nop,nop,timestamp 2933572 183750635> (DF)
18:24:30.956220 mailhost.smtp > 194.178.232.55.3017: P 1:88(87) ack 1 win 32120 <nop,nop,timestamp 183753635 2933572> (DF)
18:24:31.052675 194.178.232.55.3017 > mailhost.smtp: . ack 88 win 32120 <nop,nop,timestamp 2936572 183753635> (DF)
18:24:31.053435 194.178.232.55.3017 > mailhost.smtp: P 1:34(33) ack 88 win 32120 <nop,nop,timestamp 2936572 183753635> (DF)
18:24:31.053774 mailhost.smtp > 194.178.232.55.3017: . ack 34 win 32120 <nop,nop,timestamp 183753644 2936572> (DF)
18:24:31.053992 mailhost.smtp > 194.178.232.55.3017: P 88:186(98) ack 34 win 32120 <nop,nop,timestamp 183753645 2936572> (DF)
18:24:31.151001 194.178.232.55.3017 > mailhost.smtp: P 34:62(28) ack 186 win 32120 <nop,nop,timestamp 2936582 183753645> (DF)
18:24:31.155593 mailhost.smtp > 194.178.232.55.3017: P 186:220(34) ack 62 win 32120 <nop,nop,timestamp 183753655 2936582> (DF)
18:24:31.252743 194.178.232.55.3017 > mailhost.smtp: P 62:103(41) ack 220 win 32120 <nop,nop,timestamp 2936592 183753655> (DF)
18:24:31.255856 mailhost.smtp > 194.178.232.55.3017: P 220:274(54) ack 103 win 32120 <nop,nop,timestamp 183753665 2936592> (DF)
18:24:31.352043 194.178.232.55.3017 > mailhost.smtp: F 103:103(0) ack 274 win 32120 <nop,nop,timestamp 2936602 183753665> (DF)
18:24:31.352447 mailhost.smtp > 194.178.232.55.3017: . ack 104 win 32120 <nop,nop,timestamp 183753674 2936602> (DF)
18:24:31.357636 mailhost.smtp > 194.178.232.55.3017: F 274:274(0) ack 104 win 32120 <nop,nop,timestamp 183753675 2936602> (DF)
18:24:31.454183 194.178.232.55.3017 > mailhost.smtp: . ack 275 win 32120 <nop,nop,timestamp 2936612 183753675> (DF)
18:24:31.744025 194.178.232.55.3160 > mailhost.smtp: S 4266826584:4266826584(0) win 32120 <mss 1460,sackOK,timestamp 2936641[|tcp]> (DF)
18:24:31.744899 mailhost.smtp > 194.178.232.55.3160: S 598489660:598489660(0) ack 4266826585 win 32120 <mss 1460,sackOK,timestamp 183753714[|tcp]> (DF)
18:24:31.840784 194.178.232.55.3160 > mailhost.smtp: . ack 1 win 32120 <nop,nop,timestamp 2936651 183753714> (DF)
18:24:56.592284 194.178.232.55.3312 > mailhost.smtp: S 10900948:10900948(0) win 32120 <mss 1460,sackOK,timestamp 2939126[|tcp]> (DF)
18:24:56.593249 mailhost.smtp > 194.178.232.55.3312: S 623835070:623835070(0) ack 10900949 win 32120 <mss 1460,sackOK,timestamp 183756198[|tcp]> (DF)
18:24:56.688510 194.178.232.55.3312 > mailhost.smtp: . ack 1 win 32120 <nop,nop,timestamp 2939135 183756198> (DF)
18:25:01.747315 mailhost.smtp > 194.178.232.55.3160: P 1:88(87) ack 1 win 32120 <nop,nop,timestamp 183756714 2936651> (DF)
18:25:01.844116 194.178.232.55.3160 > mailhost.smtp: . ack 88 win 32120 <nop,nop,timestamp 2939651 183756714> (DF)
18:25:01.844851 194.178.232.55.3160 > mailhost.smtp: P 1:34(33) ack 88 win 32120 <nop,nop,timestamp 2939651 183756714> (DF)
18:25:01.845197 mailhost.smtp > 194.178.232.55.3160: . ack 34 win 32120 <nop,nop,timestamp 183756724 2939651> (DF)
18:25:01.845405 mailhost.smtp > 194.178.232.55.3160: P 88:186(98) ack 34 win 32120 <nop,nop,timestamp 183756724 2939651> (DF)
18:25:01.942123 194.178.232.55.3160 > mailhost.smtp: P 34:54(20) ack 186 win 32120 <nop,nop,timestamp 2939661 183756724> (DF)
18:25:01.944441 mailhost.smtp > 194.178.232.55.3160: P 186:224(38) ack 54 win 32120 <nop,nop,timestamp 183756733 2939661> (DF)
18:25:02.040284 194.178.232.55.3160 > mailhost.smtp: F 54:54(0) ack 224 win 32120 <nop,nop,timestamp 2939670 183756733> (DF)
18:25:02.040697 mailhost.smtp > 194.178.232.55.3160: . ack 55 win 32120 <nop,nop,timestamp 183756743 2939670> (DF)
18:25:02.040823 mailhost.smtp > 194.178.232.55.3160: F 224:224(0) ack 55 win 32120 <nop,nop,timestamp 183756743 2939670> (DF)
18:25:02.136864 194.178.232.55.3160 > mailhost.smtp: . ack 225 win 32120 <nop,nop,timestamp 2939680 183756743> (DF)
18:25:26.598206 mailhost.smtp > 194.178.232.55.3312: P 1:88(87) ack 1 win 32120 <nop,nop,timestamp 183759199 2939135> (DF)
18:25:26.694750 194.178.232.55.3312 > mailhost.smtp: . ack 88 win 32120 <nop,nop,timestamp 2942136 183759199> (DF)
18:25:26.695255 194.178.232.55.3312 > mailhost.smtp: P 1:34(33) ack 88 win 32120 <nop,nop,timestamp 2942136 183759199> (DF)
18:25:26.695598 mailhost.smtp > 194.178.232.55.3312: . ack 34 win 32120 <nop,nop,timestamp 183759208 2942136> (DF)
18:25:26.695804 mailhost.smtp > 194.178.232.55.3312: P 88:186(98) ack 34 win 32120 <nop,nop,timestamp 183759208 2942136> (DF)
18:25:26.793260 194.178.232.55.3312 > mailhost.smtp: P 34:63(29) ack 186 win 32120 <nop,nop,timestamp 2942146 183759208> (DF)
18:25:26.797878 mailhost.smtp > 194.178.232.55.3312: P 186:222(36) ack 63 win 32120 <nop,nop,timestamp 183759219 2942146> (DF)
18:25:26.895030 194.178.232.55.3312 > mailhost.smtp: P 63:122(59) ack 222 win 32120 <nop,nop,timestamp 2942156 183759219> (DF)
18:25:26.899252 mailhost.smtp > 194.178.232.55.3312: P 222:296(74) ack 122 win 32120 <nop,nop,timestamp 183759229 2942156> (DF)
18:25:26.996206 194.178.232.55.3312 > mailhost.smtp: F 122:122(0) ack 296 win 32120 <nop,nop,timestamp 2942166 183759229> (DF)
18:25:26.996599 mailhost.smtp > 194.178.232.55.3312: . ack 123 win 32120 <nop,nop,timestamp 183759239 2942166> (DF)
18:25:27.001775 mailhost.smtp > 194.178.232.55.3312: F 296:296(0) ack 123 win 32120 <nop,nop,timestamp 183759239 2942166> (DF)
18:25:27.097035 194.178.232.55.3312 > mailhost.smtp: . ack 297 win 32120 <nop,nop,timestamp 2942176 183759239> (DF)
18:25:29.664153 194.178.232.55.3502 > mailhost.smtp: S 40464102:40464102(0) win 32120 <mss 1460,sackOK,timestamp 2942433[|tcp]> (DF)
18:25:29.665092 mailhost.smtp > 194.178.232.55.3502: S 641253155:641253155(0) ack 40464103 win 32120 <mss 1460,sackOK,timestamp 183759505[|tcp]> (DF)
18:25:29.761065 194.178.232.55.3502 > mailhost.smtp: . ack 1 win 32120 <nop,nop,timestamp 2942442 183759505> (DF)
18:25:56.980423 194.178.232.55.3676 > mailhost.smtp: S 63888661:63888661(0) win 32120 <mss 1460,sackOK,timestamp 2945165[|tcp]> (DF)
18:25:56.981402 mailhost.smtp > 194.178.232.55.3676: S 681192119:681192119(0) ack 63888662 win 32120 <mss 1460,sackOK,timestamp 183762238[|tcp]> (DF)
18:25:57.076595 194.178.232.55.3676 > mailhost.smtp: . ack 1 win 32120 <nop,nop,timestamp 2945175 183762238> (DF)
18:25:59.659427 mailhost.smtp > 194.178.232.55.3502: P 1:88(87) ack 1 win 32120 <nop,nop,timestamp 183762506 2942442> (DF)
18:25:59.755831 194.178.232.55.3502 > mailhost.smtp: . ack 88 win 32120 <nop,nop,timestamp 2945443 183762506> (DF)
18:25:59.756607 194.178.232.55.3502 > mailhost.smtp: P 1:34(33) ack 88 win 32120 <nop,nop,timestamp 2945443 183762506> (DF)
18:25:59.756965 mailhost.smtp > 194.178.232.55.3502: . ack 34 win 32120 <nop,nop,timestamp 183762515 2945443> (DF)
18:25:59.757169 mailhost.smtp > 194.178.232.55.3502: P 88:186(98) ack 34 win 32120 <nop,nop,timestamp 183762516 2945443> (DF)
18:25:59.854270 194.178.232.55.3502 > mailhost.smtp: P 34:63(29) ack 186 win 32120 <nop,nop,timestamp 2945453 183762516> (DF)
18:25:59.858921 mailhost.smtp > 194.178.232.55.3502: P 186:222(36) ack 63 win 32120 <nop,nop,timestamp 183762526 2945453> (DF)
18:25:59.956031 194.178.232.55.3502 > mailhost.smtp: P 63:128(65) ack 222 win 32120 <nop,nop,timestamp 2945463 183762526> (DF)
18:25:59.960159 mailhost.smtp > 194.178.232.55.3502: P 222:302(80) ack 128 win 32120 <nop,nop,timestamp 183762536 2945463> (DF)
18:26:00.056535 194.178.232.55.3502 > mailhost.smtp: F 128:128(0) ack 302 win 32120 <nop,nop,timestamp 2945473 183762536> (DF)
18:26:00.056931 mailhost.smtp > 194.178.232.55.3502: . ack 129 win 32120 <nop,nop,timestamp 183762545 2945473> (DF)
18:26:00.062130 mailhost.smtp > 194.178.232.55.3502: F 302:302(0) ack 129 win 32120 <nop,nop,timestamp 183762546 2945473> (DF)
18:26:00.157630 194.178.232.55.3502 > mailhost.smtp: . ack 303 win 32120 <nop,nop,timestamp 2945483 183762546> (DF)
18:26:22.236347 194.178.232.55.3824 > mailhost.smtp: S 110878168:110878168(0) win 32120 <mss 1460,sackOK,timestamp 2947691[|tcp]> (DF)
18:26:22.237278 mailhost.smtp > 194.178.232.55.3824: S 694096234:694096234(0) ack 110878169 win 32120 <mss 1460,sackOK,timestamp 183764763[|tcp]> (DF)
18:26:22.333052 194.178.232.55.3824 > mailhost.smtp: . ack 1 win 32120 <nop,nop,timestamp 2947701 183764763> (DF)
18:26:26.980382 mailhost.smtp > 194.178.232.55.3676: P 1:88(87) ack 1 win 32120 <nop,nop,timestamp 183765238 2945175> (DF)
```

```
18:26:27.076600 194.178.232.55.3676 > mailhost.smtp: . ack 88 win 32120 <nop,nop,timestamp 2948175 183765238> (DF)
18:26:27.077377 194.178.232.55.3676 > mailhost.smtp: P 1:34(33) ack 88 win 32120 <nop,nop,timestamp 2948175 183765238> (DF)
18:26:27.077717 mailhost.smtp > 194.178.232.55.3676: . ack 34 win 32120 <nop,nop,timestamp 183765247 2948175> (DF)
18:26:27.077924 mailhost.smtp > 194.178.232.55.3676: P 88:186(98) ack 34 win 32120 <nop,nop,timestamp 183765247 2948175> (DF)
18:26:27.176191 194.178.232.55.3676 > mailhost.smtp: P 34:63(29) ack 186 win 32120 <nop,nop,timestamp 2948185 183765247> (DF)
18:26:27.180826 mailhost.smtp > 194.178.232.55.3676: P 186:222(36) ack 63 win 32120 <nop,nop,timestamp 183765258 2948185> (DF)
18:26:27.277642 194.178.232.55.3676 > mailhost.smtp: P 63:122(59) ack 222 win 32120 <nop,nop,timestamp 2948195 183765258> (DF)
18:26:27.281866 mailhost.smtp > 194.178.232.55.3676: P 222:296(74) ack 122 win 32120 <nop,nop,timestamp 183765268 2948195> (DF)
18:26:27.379000 194.178.232.55.3676 > mailhost.smtp: F 122:122(0) ack 296 win 32120 <nop,nop,timestamp 2948205 183765268> (DF)
18:26:27.379402 mailhost.smtp > 194.178.232.55.3676: . ack 123 win 32120 <nop,nop,timestamp 183765278 2948205> (DF)
18:26:27.384576 mailhost.smtp > 194.178.232.55.3676: F 296:296(0) ack 123 win 32120 <nop,nop,timestamp 183765278 2948205> (DF)
18:26:27.480103 194.178.232.55.3676 > mailhost.smtp: . ack 297 win 32120 <nop,nop,timestamp 2948215 183765278> (DF)
18:26:49.927330 194.178.232.55.3987 > mailhost.smtp: S 129968352:129968352(0) win 32120 <mss 1460,sackOK,timestamp 2950460[|tcp]> (DF)
18:26:49.928292 mailhost.smtp > 194.178.232.55.3987: S 726646089:726646089(0) ack 129968353 win 32120 <mss 1460,sackOK,timestamp 183767532[|tcp]> (DF)
18:26:50.024456 194.178.232.55.3987 > mailhost.smtp: . ack 1 win 32120 <nop,nop,timestamp 2950470 183767532> (DF)
18:26:52.241311 mailhost.smtp > 194.178.232.55.3824: P 1:88(87) ack 1 win 32120 <nop,nop,timestamp 183767764 2947701> (DF)
18:26:52.337856 194.178.232.55.3824 > mailhost.smtp: . ack 88 win 32120 <nop,nop,timestamp 2950701 183767764> (DF)
18:26:52.338620 194.178.232.55.3824 > mailhost.smtp: P 1:34(33) ack 88 win 32120 <nop,nop,timestamp 2950701 183767764> (DF)
18:26:52.338969 mailhost.smtp > 194.178.232.55.3824: . ack 34 win 32120 <nop,nop,timestamp 183767774 2950701> (DF)
18:26:52.339178 mailhost.smtp > 194.178.232.55.3824: P 88:186(98) ack 34 win 32120 <nop,nop,timestamp 183767774 2950701> (DF)
18:26:52.436316 194.178.232.55.3824 > mailhost.smtp: P 34:63(29) ack 186 win 32120 <nop,nop,timestamp 2950711 183767774> (DF)
18:26:52.440924 mailhost.smtp > 194.178.232.55.3824: P 186:222(36) ack 63 win 32120 <nop,nop,timestamp 183767784 2950711> (DF)
18:26:52.537814 194.178.232.55.3824 > mailhost.smtp: P 63:128(65) ack 222 win 32120 <nop,nop,timestamp 2950721 183767784> (DF)
18:26:52.541911 mailhost.smtp > 194.178.232.55.3824: P 222:302(80) ack 128 win 32120 <nop,nop,timestamp 183767794 2950721> (DF)
18:26:52.638459 194.178.232.55.3824 > mailhost.smtp: F 128:128(0) ack 302 win 32120 <nop,nop,timestamp 2950731 183767794> (DF)
18:26:52.638857 mailhost.smtp > 194.178.232.55.3824: . ack 129 win 32120 <nop,nop,timestamp 183767803 2950731> (DF)
18:26:52.644048 mailhost.smtp > 194.178.232.55.3824: F 302:302(0) ack 129 win 32120 <nop,nop,timestamp 183767804 2950731> (DF)
18:26:52.739437 194.178.232.55.3824 > mailhost.smtp: . ack 303 win 32120 <nop,nop,timestamp 2950741 183767804> (DF)
18:27:15.572243 194.178.232.55.4132 > mailhost.smtp: S 164587681:164587681(0) win 32120 <mss 1460,sackOK,timestamp 2953024[|tcp]> (DF)
18:27:15.573294 mailhost.smtp > 194.178.232.55.4132: S 765930278:765930278(0) ack 164587682 win 32120 <mss 1460,sackOK,timestamp 183770097[|tcp]> (DF)
18:27:15.669766 194.178.232.55.4132 > mailhost.smtp: . ack 1 win 32120 <nop,nop,timestamp 2953034 183770097> (DF)
18:27:19.932302 mailhost.smtp > 194.178.232.55.3987: P 1:88(87) ack 1 win 32120 <nop,nop,timestamp 183770533 2950470> (DF)
18:27:20.028620 194.178.232.55.3987 > mailhost.smtp: . ack 88 win 32120 <nop,nop,timestamp 2953470 183770533> (DF)
18:27:20.029407 194.178.232.55.3987 > mailhost.smtp: P 1:34(33) ack 88 win 32120 <nop,nop,timestamp 2953470 183770533> (DF)
18:27:20.029747 mailhost.smtp > 194.178.232.55.3987: . ack 34 win 32120 <nop,nop,timestamp 183770542 2953470> (DF)
18:27:20.029956 mailhost.smtp > 194.178.232.55.3987: P 88:186(98) ack 34 win 32120 <nop,nop,timestamp 183770542 2953470> (DF)
18:27:20.127265 194.178.232.55.3987 > mailhost.smtp: P 34:63(29) ack 186 win 32120 <nop,nop,timestamp 2953480 183770542> (DF)
18:27:20.131917 mailhost.smtp > 194.178.232.55.3987: P 186:222(36) ack 63 win 32120 <nop,nop,timestamp 183770553 2953480> (DF)
18:27:20.230519 194.178.232.55.3987 > mailhost.smtp: P 63:122(59) ack 222 win 32120 <nop,nop,timestamp 2953490 183770553> (DF)
18:27:20.235089 mailhost.smtp > 194.178.232.55.3987: P 222:296(74) ack 122 win 32120 <nop,nop,timestamp 183770563 2953490> (DF)
18:27:20.332156 194.178.232.55.3987 > mailhost.smtp: F 122:122(0) ack 296 win 32120 <nop,nop,timestamp 2953500 183770563> (DF)
18:27:20.332549 mailhost.smtp > 194.178.232.55.3987: . ack 123 win 32120 <nop,nop,timestamp 183770573 2953500> (DF)
18:27:20.337726 mailhost.smtp > 194.178.232.55.3987: F 296:296(0) ack 123 win 32120 <nop,nop,timestamp 183770573 2953500> (DF)
18:27:20.433556 194.178.232.55.3987 > mailhost.smtp: . ack 297 win 32120 <nop,nop,timestamp 2953510 183770573> (DF)
18:27:41.983885 194.178.232.55.4286 > mailhost.smtp: S 180219493:180219493(0) win 32120 <mss 1460,sackOK,timestamp 2955665[|tcp]> (DF)
18:27:41.984841 mailhost.smtp > 194.178.232.55.4286: S 796962529:796962529(0) ack 180219494 win 32120 <mss 1460,sackOK,timestamp 183772738[|tcp]> (DF)
18:27:42.080229 194.178.232.55.4286 > mailhost.smtp: . ack 1 win 32120 <nop,nop,timestamp 2955675 183772738> (DF)
18:27:45.573218 mailhost.smtp > 194.178.232.55.4132: P 1:88(87) ack 1 win 32120 <nop,nop,timestamp 183773097 2953034> (DF)
18:27:45.670371 194.178.232.55.4132 > mailhost.smtp: . ack 88 win 32120 <nop,nop,timestamp 2956034 183773097> (DF)
18:27:45.670878 194.178.232.55.4132 > mailhost.smtp: P 1:34(33) ack 88 win 32120 <nop,nop,timestamp 2956034 183773097> (DF)
18:27:45.671219 mailhost.smtp > 194.178.232.55.4132: . ack 34 win 32120 <nop,nop,timestamp 183773107 2956034> (DF)
18:27:45.671427 mailhost.smtp > 194.178.232.55.4132: P 88:186(98) ack 34 win 32120 <nop,nop,timestamp 183773107 2956034> (DF)
18:27:45.768863 194.178.232.55.4132 > mailhost.smtp: P 34:63(29) ack 186 win 32120 <nop,nop,timestamp 2956044 183773107> (DF)
18:27:45.773491 mailhost.smtp > 194.178.232.55.4132: P 186:222(36) ack 63 win 32120 <nop,nop,timestamp 183773117 2956044> (DF)
18:27:45.875544 194.178.232.55.4132 > mailhost.smtp: P 63:128(65) ack 222 win 32120 <nop,nop,timestamp 2956054 183773117> (DF)
18:27:45.879972 mailhost.smtp > 194.178.232.55.4132: P 222:302(80) ack 128 win 32120 <nop,nop,timestamp 183773127 2956054> (DF)
18:27:45.977509 194.178.232.55.4132 > mailhost.smtp: F 128:128(0) ack 302 win 32120 <nop,nop,timestamp 2956065 183773127> (DF)
18:27:45.977940 mailhost.smtp > 194.178.232.55.4132: . ack 129 win 32120 <nop,nop,timestamp 183773137 2956065> (DF)
18:27:45.983152 mailhost.smtp > 194.178.232.55.4132: F 302:302(0) ack 129 win 32120 <nop,nop,timestamp 183773138 2956065> (DF)
18:27:46.079050 194.178.232.55.4132 > mailhost.smtp: . ack 303 win 32120 <nop,nop,timestamp 2956075 183773138> (DF)
18:28:08.617633 194.178.232.55.4454 > mailhost.smtp: S 207811540:207811540(0) win 32120 <mss 1460,sackOK,timestamp 2958329[|tcp]> (DF)
18:28:08.618569 mailhost.smtp > 194.178.232.55.4454: S 820993144:820993144(0) ack 207811541 win 32120 <mss 1460,sackOK,timestamp 183775401[|tcp]> (DF)
18:28:08.714265 194.178.232.55.4454 > mailhost.smtp: . ack 1 win 32120 <nop,nop,timestamp 2958338 183775401> (DF)
18:28:11.984213 mailhost.smtp > 194.178.232.55.4286: P 1:88(87) ack 1 win 32120 <nop,nop,timestamp 183775738 2955675> (DF)
18:28:12.080360 194.178.232.55.4286 > mailhost.smtp: . ack 88 win 32120 <nop,nop,timestamp 2958675 183775738> (DF)
18:28:12.081391 194.178.232.55.4286 > mailhost.smtp: P 1:34(33) ack 88 win 32120 <nop,nop,timestamp 2958675 183775738> (DF)
18:28:12.081732 mailhost.smtp > 194.178.232.55.4286: . ack 34 win 32120 <nop,nop,timestamp 183775747 2958675> (DF)
18:28:12.081943 mailhost.smtp > 194.178.232.55.4286: P 88:186(98) ack 34 win 32120 <nop,nop,timestamp 183775747 2958675> (DF)
18:28:12.178868 194.178.232.55.4286 > mailhost.smtp: P 34:63(29) ack 186 win 32120 <nop,nop,timestamp 2958685 183775747> (DF)
18:28:12.183502 mailhost.smtp > 194.178.232.55.4286: P 186:222(36) ack 63 win 32120 <nop,nop,timestamp 183775758 2958685> (DF)
18:28:12.279808 194.178.232.55.4286 > mailhost.smtp: P 63:123(60) ack 222 win 32120 <nop,nop,timestamp 2958695 183775758> (DF)
18:28:12.284723 mailhost.smtp > 194.178.232.55.4286: P 222:297(75) ack 123 win 32120 <nop,nop,timestamp 183775768 2958695> (DF)
18:28:12.382749 194.178.232.55.4286 > mailhost.smtp: F 123:123(0) ack 297 win 32120 <nop,nop,timestamp 2958705 183775768> (DF)
18:28:12.383154 mailhost.smtp > 194.178.232.55.4286: . ack 124 win 32120 <nop,nop,timestamp 183775778 2958705> (DF)
18:28:12.388337 mailhost.smtp > 194.178.232.55.4286: F 297:297(0) ack 124 win 32120 <nop,nop,timestamp 183775778 2958705> (DF)
18:28:12.484704 194.178.232.55.4286 > mailhost.smtp: . ack 298 win 32120 <nop,nop,timestamp 2958715 183775778> (DF)
18:28:32.835907 194.178.232.55.4613 > mailhost.smtp: S 242462077:242462077(0) win 32120 <mss 1460,sackOK,timestamp 2960750[|tcp]> (DF)
18:28:32.836853 mailhost.smtp > 194.178.232.55.4613: S 842764798:842764798(0) ack 242462078 win 32120 <mss 1460,sackOK,timestamp 183777823[|tcp]> (DF)
18:28:32.933450 194.178.232.55.4613 > mailhost.smtp: . ack 1 win 32120 <nop,nop,timestamp 2960760 183777823> (DF)
18:28:38.615173 mailhost.smtp > 194.178.232.55.4454: P 1:88(87) ack 1 win 32120 <nop,nop,timestamp 183778401 2958338> (DF)
18:28:38.711497 194.178.232.55.4454 > mailhost.smtp: . ack 88 win 32120 <nop,nop,timestamp 2961338 183778401> (DF)
18:28:38.712250 194.178.232.55.4454 > mailhost.smtp: P 1:34(33) ack 88 win 32120 <nop,nop,timestamp 2961338 183778401> (DF)
18:28:38.712594 mailhost.smtp > 194.178.232.55.4454: . ack 34 win 32120 <nop,nop,timestamp 183778410 2961338> (DF)
18:28:38.712801 mailhost.smtp > 194.178.232.55.4454: P 88:186(98) ack 34 win 32120 <nop,nop,timestamp 183778410 2961338> (DF)
18:28:38.809820 194.178.232.55.4454 > mailhost.smtp: P 34:63(29) ack 186 win 32120 <nop,nop,timestamp 2961348 183778410> (DF)
18:28:38.814430 mailhost.smtp > 194.178.232.55.4454: P 186:222(36) ack 63 win 32120 <nop,nop,timestamp 183778421 2961348> (DF)
18:28:38.910784 194.178.232.55.4454 > mailhost.smtp: P 63:129(66) ack 222 win 32120 <nop,nop,timestamp 2961358 183778421> (DF)
18:28:38.915358 mailhost.smtp > 194.178.232.55.4454: P 222:303(81) ack 129 win 32120 <nop,nop,timestamp 183778431 2961358> (DF)
18:28:39.011925 194.178.232.55.4454 > mailhost.smtp: F 129:129(0) ack 303 win 32120 <nop,nop,timestamp 2961368 183778431> (DF)
18:28:39.012342 mailhost.smtp > 194.178.232.55.4454: . ack 130 win 32120 <nop,nop,timestamp 183778440 2961368> (DF)
18:28:39.017548 mailhost.smtp > 194.178.232.55.4454: F 303:303(0) ack 130 win 32120 <nop,nop,timestamp 183778441 2961368> (DF)
18:28:39.113138 194.178.232.55.4454 > mailhost.smtp: . ack 304 win 32120 <nop,nop,timestamp 2961378 183778441> (DF)
18:28:55.835667 194.178.232.55.4754 > mailhost.smtp: S 262962991:262962991(0) win 32120 <mss 1460,sackOK,timestamp 2963050[|tcp]> (DF)
18:28:55.836616 mailhost.smtp > 194.178.232.55.4754: S 869897125:869897125(0) ack 262962992 win 32120 <mss 1460,sackOK,timestamp 183780123[|tcp]> (DF)
18:28:55.934125 194.178.232.55.4754 > mailhost.smtp: . ack 1 win 32120 <nop,nop,timestamp 2963060 183780123> (DF)
18:29:02.836062 mailhost.smtp > 194.178.232.55.4613: P 1:88(87) ack 1 win 32120 <nop,nop,timestamp 183780823 2960760> (DF)
18:29:02.933549 194.178.232.55.4613 > mailhost.smtp: . ack 88 win 32120 <nop,nop,timestamp 2963760 183780823> (DF)
18:29:02.934055 194.178.232.55.4613 > mailhost.smtp: P 1:34(33) ack 88 win 32120 <nop,nop,timestamp 2963760 183780823> (DF)
18:29:02.934400 mailhost.smtp > 194.178.232.55.4613: . ack 34 win 32120 <nop,nop,timestamp 183780833 2963760> (DF)
18:29:02.934603 mailhost.smtp > 194.178.232.55.4613: P 88:186(98) ack 34 win 32120 <nop,nop,timestamp 183780833 2963760> (DF)
18:29:03.031752 194.178.232.55.4613 > mailhost.smtp: P 34:63(29) ack 186 win 32120 <nop,nop,timestamp 2963770 183780833> (DF)
18:29:03.036370 mailhost.smtp > 194.178.232.55.4613: P 186:222(36) ack 63 win 32120 <nop,nop,timestamp 183780843 2963770> (DF)
18:29:03.133186 194.178.232.55.4613 > mailhost.smtp: P 63:104(41) ack 222 win 32120 <nop,nop,timestamp 2963780 183780843> (DF)
18:29:03.136397 mailhost.smtp > 194.178.232.55.4613: P 222:278(56) ack 104 win 32120 <nop,nop,timestamp 183780853 2963780> (DF)
18:29:03.233516 194.178.232.55.4613 > mailhost.smtp: F 104:104(0) ack 278 win 32120 <nop,nop,timestamp 2963790 183780853> (DF)
18:29:03.233925 mailhost.smtp > 194.178.232.55.4613: . ack 105 win 32120 <nop,nop,timestamp 183780863 2963790> (DF)
18:29:03.239100 mailhost.smtp > 194.178.232.55.4613: F 278:278(0) ack 105 win 32120 <nop,nop,timestamp 183780863 2963790> (DF)
18:29:03.335447 194.178.232.55.4613 > mailhost.smtp: . ack 279 win 32120 <nop,nop,timestamp 2963800 183780863> (DF)
18:29:21.548193 194.178.232.55.4901 > mailhost.smtp: S 290954911:290954911(0) win 32120 <mss 1460,sackOK,timestamp 2965622[|tcp]> (DF)
18:29:21.549196 mailhost.smtp > 194.178.232.55.4901: S 899609476:899609476(0) ack 290954912 win 32120 <mss 1460,sackOK,timestamp 183782694[|tcp]> (DF)
18:29:21.646849 194.178.232.55.4901 > mailhost.smtp: . ack 1 win 32120 <nop,nop,timestamp 2965631 183782694> (DF)
18:29:25.836896 mailhost.smtp > 194.178.232.55.4754: P 1:88(87) ack 1 win 32120 <nop,nop,timestamp 183783123 2963060> (DF)
```

```
18:29:25.933750 194.178.232.55.4754 > mailhost.smtp: . ack 88 win 32120 <nop,nop,timestamp 2966060 183783123> (DF)
18:29:25.934324 194.178.232.55.4754 > mailhost.smtp: P 1:34(33) ack 88 win 32120 <nop,nop,timestamp 2966060 183783123> (DF)
18:29:25.934664 mailhost.smtp > 194.178.232.55.4754: . ack 34 win 32120 <nop,nop,timestamp 183783133 2966060> (DF)
18:29:25.934870 mailhost.smtp > 194.178.232.55.4754: P 88:186(98) ack 34 win 32120 <nop,nop,timestamp 183783133 2966060> (DF)
18:29:26.032717 194.178.232.55.4754 > mailhost.smtp: P 34:54(20) ack 186 win 32120 <nop,nop,timestamp 2966070 183783133> (DF)
18:29:26.035076 mailhost.smtp > 194.178.232.55.4754: P 186:224(38) ack 54 win 32120 <nop,nop,timestamp 183783143 2966070> (DF)
18:29:26.130939 194.178.232.55.4754 > mailhost.smtp: F 54:54(0) ack 224 win 32120 <nop,nop,timestamp 2966080 183783143> (DF)
18:29:26.131375 mailhost.smtp > 194.178.232.55.4754: . ack 55 win 32120 <nop,nop,timestamp 183783152 2966080> (DF)
18:29:26.131494 mailhost.smtp > 194.178.232.55.4754: F 224:224(0) ack 55 win 32120 <nop,nop,timestamp 183783152 2966080> (DF)
18:29:26.227282 194.178.232.55.4754 > mailhost.smtp: . ack 225 win 32120 <nop,nop,timestamp 2966090 183783152> (DF)
18:29:51.547900 mailhost.smtp > 194.178.232.55.4901: P 1:88(87) ack 1 win 32120 <nop,nop,timestamp 183785694 2965631> (DF)
18:29:51.644259 194.178.232.55.4901 > mailhost.smtp: . ack 88 win 32120 <nop,nop,timestamp 2968631 183785694> (DF)
18:29:51.644989 194.178.232.55.4901 > mailhost.smtp: P 1:34(33) ack 88 win 32120 <nop,nop,timestamp 2968631 183785694> (DF)
18:29:51.645331 mailhost.smtp > 194.178.232.55.4901: . ack 34 win 32120 <nop,nop,timestamp 183785703 2968631> (DF)
18:29:51.645541 mailhost.smtp > 194.178.232.55.4901: P 88:186(98) ack 34 win 32120 <nop,nop,timestamp 183785703 2968631> (DF)
18:29:51.742937 194.178.232.55.4901 > mailhost.smtp: P 34:48(14) ack 186 win 32120 <nop,nop,timestamp 2968641 183785703> (DF)
18:29:51.744416 mailhost.smtp > 194.178.232.55.4901: P 186:207(21) ack 48 win 32120 <nop,nop,timestamp 183785713 2968641> (DF)
18:29:51.841544 194.178.232.55.4901 > mailhost.smtp: P 48:89(41) ack 207 win 32120 <nop,nop,timestamp 2968651 183785713> (DF)
18:29:51.845341 mailhost.smtp > 194.178.232.55.4901: P 207:263(56) ack 89 win 32120 <nop,nop,timestamp 183785723 2968651> (DF)
18:29:51.942979 194.178.232.55.4901 > mailhost.smtp: F 89:89(0) ack 263 win 32120 <nop,nop,timestamp 2968661 183785723> (DF)
18:29:51.943396 mailhost.smtp > 194.178.232.55.4901: . ack 90 win 32120 <nop,nop,timestamp 183785733 2968661> (DF)
18:29:51.948556 mailhost.smtp > 194.178.232.55.4901: F 263:263(0) ack 90 win 32120 <nop,nop,timestamp 183785734 2968661> (DF)
18:29:52.044480 194.178.232.55.4901 > mailhost.smtp: . ack 264 win 32120 <nop,nop,timestamp 2968671 183785734> (DF)
18:29:58.050685 194.178.232.55.1065 > mailhost.smtp: S 322567430:322567430(0) win 32120 <mss 1460,sackOK,timestamp 2969272[|tcp]> (DF)
18:29:58.051604 mailhost.smtp > 194.178.232.55.1065: S 938726431:938726431(0) ack 322567431 win 32120 <mss 1460,sackOK,timestamp 183786344[|tcp]> (DF)
18:29:58.147998 194.178.232.55.1065 > mailhost.smtp: . ack 1 win 32120 <nop,nop,timestamp 2969282 183786344> (DF)
18:30:17.249822 194.178.232.55.1233 > mailhost.smtp: S 340945192:340945192(0) win 32120 <mss 1460,sackOK,timestamp 2971192[|tcp]> (DF)
18:30:17.250791 mailhost.smtp > 194.178.232.55.1233: S 948911590:948911590(0) ack 340945193 win 32120 <mss 1460,sackOK,timestamp 183788264[|tcp]> (DF)
18:30:17.346594 194.178.232.55.1233 > mailhost.smtp: . ack 1 win 32120 <nop,nop,timestamp 2971201 183788264> (DF)
18:30:28.049221 mailhost.smtp > 194.178.232.55.1065: P 1:88(87) ack 1 win 32120 <nop,nop,timestamp 183789344 2969282> (DF)
18:30:28.145274 194.178.232.55.1065 > mailhost.smtp: . ack 88 win 32120 <nop,nop,timestamp 2972281 183789344> (DF)
18:30:28.146093 194.178.232.55.1065 > mailhost.smtp: P 1:34(33) ack 88 win 32120 <nop,nop,timestamp 2972281 183789344> (DF)
18:30:28.146441 mailhost.smtp > 194.178.232.55.1065: . ack 34 win 32120 <nop,nop,timestamp 183789353 2972281> (DF)
18:30:28.146648 mailhost.smtp > 194.178.232.55.1065: P 88:186(98) ack 34 win 32120 <nop,nop,timestamp 183789353 2972281> (DF)
18:30:28.243471 194.178.232.55.1065 > mailhost.smtp: P 34:70(36) ack 186 win 32120 <nop,nop,timestamp 2972291 183789353> (DF)
18:30:28.255592 mailhost.smtp > 194.178.232.55.1065: P 186:229(43) ack 70 win 32120 <nop,nop,timestamp 183789364 2972291> (DF)
18:30:28.352076 194.178.232.55.1065 > mailhost.smtp: P 70:111(41) ack 229 win 32120 <nop,nop,timestamp 2972302 183789364> (DF)
18:30:28.355283 mailhost.smtp > 194.178.232.55.1065: P 229:285(56) ack 111 win 32120 <nop,nop,timestamp 183789374 2972302> (DF)
18:30:28.451976 194.178.232.55.1065 > mailhost.smtp: F 111:111(0) ack 285 win 32120 <nop,nop,timestamp 2972312 183789374> (DF)
18:30:28.452360 mailhost.smtp > 194.178.232.55.1065: . ack 112 win 32120 <nop,nop,timestamp 183789384 2972312> (DF)
18:30:28.457539 mailhost.smtp > 194.178.232.55.1065: F 285:285(0) ack 112 win 32120 <nop,nop,timestamp 183789385 2972312> (DF)
18:30:28.553466 194.178.232.55.1065 > mailhost.smtp: . ack 286 win 32120 <nop,nop,timestamp 2972322 183789385> (DF)
18:30:47.249927 mailhost.smtp > 194.178.232.55.1233: P 1:88(87) ack 1 win 32120 <nop,nop,timestamp 183791264 2971201> (DF)
18:30:47.346076 194.178.232.55.1233 > mailhost.smtp: . ack 88 win 32120 <nop,nop,timestamp 2974201 183791264> (DF)
18:30:47.346720 194.178.232.55.1233 > mailhost.smtp: P 1:34(33) ack 88 win 32120 <nop,nop,timestamp 2974201 183791264> (DF)
18:30:47.347062 mailhost.smtp > 194.178.232.55.1233: . ack 34 win 32120 <nop,nop,timestamp 183791273 2974201> (DF)
18:30:47.347269 mailhost.smtp > 194.178.232.55.1233: P 88:186(98) ack 34 win 32120 <nop,nop,timestamp 183791273 2974201> (DF)
18:30:47.444587 194.178.232.55.1233 > mailhost.smtp: P 34:72(38) ack 186 win 32120 <nop,nop,timestamp 2974211 183791273> (DF)
18:30:47.447689 mailhost.smtp > 194.178.232.55.1233: P 186:231(45) ack 72 win 32120 <nop,nop,timestamp 183791284 2974211> (DF)
18:30:47.544661 194.178.232.55.1233 > mailhost.smtp: P 72:113(41) ack 231 win 32120 <nop,nop,timestamp 2974221 183791284> (DF)
18:30:47.548173 mailhost.smtp > 194.178.232.55.1233: P 231:287(56) ack 113 win 32120 <nop,nop,timestamp 183791294 2974221> (DF)
18:30:47.644576 194.178.232.55.1233 > mailhost.smtp: F 113:113(0) ack 287 win 32120 <nop,nop,timestamp 2974231 183791294> (DF)
18:30:47.645002 mailhost.smtp > 194.178.232.55.1233: . ack 114 win 32120 <nop,nop,timestamp 183791303 2974231> (DF)
18:30:47.650199 mailhost.smtp > 194.178.232.55.1233: F 287:287(0) ack 114 win 32120 <nop,nop,timestamp 183791304 2974231> (DF)
18:30:47.746160 194.178.232.55.1233 > mailhost.smtp: . ack 288 win 32120 <nop,nop,timestamp 2974241 183791304> (DF)
18:30:48.618077 194.178.232.55.1392 > mailhost.smtp: S 370289671:370289671(0) win 32120 <mss 1460,sackOK,timestamp 2974328[|tcp]> (DF)
18:30:48.618966 mailhost.smtp > 194.178.232.55.1392: S 987485004:987485004(0) ack 370289672 win 32120 <mss 1460,sackOK,timestamp 183791401[|tcp]> (DF)
18:30:48.714535 194.178.232.55.1392 > mailhost.smtp: . ack 1 win 32120 <nop,nop,timestamp 2974338 183791401> (DF)
18:31:18.621091 mailhost.smtp > 194.178.232.55.1392: P 1:88(87) ack 1 win 32120 <nop,nop,timestamp 183794401 2974338> (DF)
18:31:18.717437 194.178.232.55.1392 > mailhost.smtp: . ack 88 win 32120 <nop,nop,timestamp 2977338 183794401> (DF)
18:31:18.718217 194.178.232.55.1392 > mailhost.smtp: P 1:34(33) ack 88 win 32120 <nop,nop,timestamp 2977338 183794401> (DF)
18:31:18.718555 mailhost.smtp > 194.178.232.55.1392: . ack 34 win 32120 <nop,nop,timestamp 183794410 2977338> (DF)
18:31:18.718773 mailhost.smtp > 194.178.232.55.1392: P 88:186(98) ack 34 win 32120 <nop,nop,timestamp 183794411 2977338> (DF)
18:31:18.816158 194.178.232.55.1392 > mailhost.smtp: P 34:78(44) ack 186 win 32120 <nop,nop,timestamp 2977348 183794411> (DF)
18:31:18.820604 mailhost.smtp > 194.178.232.55.1392: P 186:237(51) ack 78 win 32120 <nop,nop,timestamp 183794421 2977348> (DF)
18:31:18.917482 194.178.232.55.1392 > mailhost.smtp: P 78:119(41) ack 237 win 32120 <nop,nop,timestamp 2977358 183794421> (DF)
18:31:18.920806 mailhost.smtp > 194.178.232.55.1392: P 237:293(56) ack 119 win 32120 <nop,nop,timestamp 183794431 2977358> (DF)
18:31:19.018101 194.178.232.55.1392 > mailhost.smtp: F 119:119(0) ack 293 win 32120 <nop,nop,timestamp 2977368 183794431> (DF)
18:31:19.018499 mailhost.smtp > 194.178.232.55.1392: . ack 120 win 32120 <nop,nop,timestamp 183794440 2977368> (DF)
18:31:19.023638 mailhost.smtp > 194.178.232.55.1392: F 293:293(0) ack 120 win 32120 <nop,nop,timestamp 183794441 2977368> (DF)
18:31:19.119857 194.178.232.55.1392 > mailhost.smtp: . ack 294 win 32120 <nop,nop,timestamp 2977379 183794441> (DF)
```

maillog output

```
Jul 17 18:27:58 mailhost sendmail[16917]: WAA16917: ruleset=check_rcpt, arg1=<orbs-relaytest@manawatu.co.nz>, relay=relaytest.orbs.vuurwerk.nl [194.178.232.55], reje
Jul 17 18:27:58 mailhost sendmail[16917]: WAA16917: lost input channel from relaytest.orbs.vuurwerk.nl [194.178.232.55]
Jul 17 18:27:58 mailhost sendmail[16917]: WAA16917: from=<sender@orbs.org>, size=0, class=0, pri=0, nrcpts=0, proto=SMTP, relay=relaytest.orbs.vuurwerk.nl [194.178.2
Jul 17 18:28:23 mailhost sendmail[16918]: WAA16918: ruleset=check_rcpt, arg1=<"orbs-relaytest@manawatu.co.nz">, relay=relaytest.orbs.vuurwerk.nl [194.178.232.55], re
Jul 17 18:28:23 mailhost sendmail[16918]: WAA16918: lost input channel from relaytest.orbs.vuurwerk.nl [194.178.232.55]
Jul 17 18:28:23 mailhost sendmail[16918]: WAA16918: from=<sender@orbs.org>, size=0, class=0, pri=0, nrcpts=0, proto=SMTP, relay=relaytest.orbs.vuurwerk.nl [194.178.2
Jul 17 18:29:53 mailhost sendmail[16920]: WAA16920: ruleset=check_rcpt, arg1=orbs-relaytest@manawatu.co.nz, relay=relaytest.orbs.vuurwerk.nl [194.178.232.55], reject
Jul 17 18:29:53 mailhost sendmail[16920]: WAA16920: lost input channel from relaytest.orbs.vuurwerk.nl [194.178.232.55]
Jul 17 18:29:53 mailhost sendmail[16920]: WAA16920: from=sender@orbs.org, size=0, class=0, pri=0, nrcpts=0, proto=SMTP, relay=relaytest.orbs.vuurwerk.nl [194.178.232
Jul 17 18:30:24 mailhost sendmail[16925]: WAA16925: ruleset=check_mail, arg1=<sender>, relay=relaytest.orbs.vuurwerk.nl [194.178.232.55], reject=553 <sender>... Doma
Jul 17 18:30:24 mailhost sendmail[16925]: WAA16925: from=<sender>, size=0, class=0, pri=0, nrcpts=0, proto=SMTP, relay=relaytest.orbs.vuurwerk.nl [194.178.232.55]
Jul 17 18:30:49 mailhost sendmail[16926]: WAA16926: ruleset=check_rcpt, arg1=<orbs-relaytest%manawatu.co.nz@[256.0.0.1]>, relay=relaytest.orbs.vuurwerk.nl [194.178.2
Jul 17 18:30:49 mailhost sendmail[16926]: WAA16926: lost input channel from relaytest.orbs.vuurwerk.nl [194.178.232.55]
Jul 17 18:30:49 mailhost sendmail[16926]: WAA16926: from=<sender@orbs.org>, size=0, class=0, pri=0, nrcpts=0, proto=SMTP, relay=relaytest.orbs.vuurwerk.nl [194.178.2
Jul 17 18:31:22 mailhost sendmail[16929]: WAA16929: ruleset=check_rcpt, arg1=<orbs-relaytest%manawatu.co.nz@mailhost.my.domain>, relay=relaytest.orbs.vuurwerk.nl [19
Jul 17 18:31:22 mailhost sendmail[16929]: WAA16929: lost input channel from relaytest.orbs.vuurwerk.nl [194.178.232.55]
Jul 17 18:31:22 mailhost sendmail[16929]: WAA16929: from=<sender@orbs.org>, size=0, class=0, pri=0, nrcpts=0, proto=SMTP, relay=relaytest.orbs.vuurwerk.nl [194.178.2
Jul 17 18:31:49 mailhost sendmail[16931]: WAA16931: ruleset=check_rcpt, arg1=<orbs-relaytest@manawatu.co.nz@[256.0.0.1]>, relay=relaytest.orbs.vuurwerk.nl [194.178.2
Jul 17 18:31:49 mailhost sendmail[16931]: WAA16931: lost input channel from relaytest.orbs.vuurwerk.nl [194.178.232.55]
Jul 17 18:31:49 mailhost sendmail[16931]: WAA16931: from=<sender@orbs.org>, size=0, class=0, pri=0, nrcpts=0, proto=SMTP, relay=relaytest.orbs.vuurwerk.nl [194.178.2
Jul 17 18:32:14 mailhost sendmail[16933]: WAA16933: ruleset=check_rcpt, arg1=<orbs-relaytest@manawatu.co.nz@mailhost.my.domain>, relay=relaytest.orbs.vuurwerk.nl [19
Jul 17 18:32:15 mailhost sendmail[16933]: WAA16933: lost input channel from relaytest.orbs.vuurwerk.nl [194.178.232.55]
Jul 17 18:32:15 mailhost sendmail[16933]: WAA16933: from=<sender@orbs.org>, size=0, class=0, pri=0, nrcpts=0, proto=SMTP, relay=relaytest.orbs.vuurwerk.nl [194.178.2
Jul 17 18:32:42 mailhost sendmail[16935]: WAA16935: ruleset=check_rcpt, arg1=<manawatu.co.nz!orbs-relaytest@[256.0.0.1]>, relay=relaytest.orbs.vuurwerk.nl [194.178.2
Jul 17 18:32:42 mailhost sendmail[16935]: WAA16935: lost input channel from relaytest.orbs.vuurwerk.nl [194.178.232.55]
Jul 17 18:32:42 mailhost sendmail[16935]: WAA16935: from=<sender@orbs.org>, size=0, class=0, pri=0, nrcpts=0, proto=SMTP, relay=relaytest.orbs.vuurwerk.nl [194.178.2
Jul 17 18:33:08 mailhost sendmail[16937]: WAA16937: ruleset=check_rcpt, arg1=<manawatu.co.nz!orbs-relaytest@mailhost.my.domain>, relay=relaytest.orbs.vuurwerk.nl [19
Jul 17 18:33:08 mailhost sendmail[16937]: WAA16937: lost input channel from relaytest.orbs.vuurwerk.nl [194.178.232.55]
Jul 17 18:33:08 mailhost sendmail[16937]: WAA16937: from=<sender@orbs.org>, size=0, class=0, pri=0, nrcpts=0, proto=SMTP, relay=relaytest.orbs.vuurwerk.nl [194.178.2
Jul 17 18:33:34 mailhost sendmail[16939]: WAA16939: ruleset=check_rcpt, arg1=<@[256.0.0.1]:orbs-relaytest@manawatu.co.nz>, relay=relaytest.orbs.vuurwerk.nl [194.178.
Jul 17 18:33:34 mailhost sendmail[16939]: WAA16939: lost input channel from relaytest.orbs.vuurwerk.nl [194.178.232.55]
Jul 17 18:33:34 mailhost sendmail[16939]: WAA16939: from=<sender@orbs.org>, size=0, class=0, pri=0, nrcpts=0, proto=SMTP, relay=relaytest.orbs.vuurwerk.nl [194.178.2
Jul 17 18:34:01 mailhost sendmail[16941]: WAA16941: ruleset=check_rcpt, arg1=<@mailhost.my.domain:orbs-relaytest@manawatu.co.nz>, relay=relaytest.orbs.vuurwerk.nl [1
Jul 17 18:34:01 mailhost sendmail[16941]: WAA16941: lost input channel from relaytest.orbs.vuurwerk.nl [194.178.232.55]
```

```
Jul 17 18:34:01 mailhost sendmail[16941]: WAA16941: from=<sender@orbs.org>, size=0, class=0, pri=0, nrcpts=0, proto=SMTP, relay=relaytest.orbs.vuurwerk.nl [194.178.2
Jul 17 18:34:25 mailhost sendmail[16943]: WAA16943: ruleset=check_rcpt, arg1=<manawatu.co.nz!orbs-relaytest>, relay=relaytest.orbs.vuurwerk.nl [194.178.232.55], reje
Jul 17 18:34:25 mailhost sendmail[16943]: WAA16943: lost input channel from relaytest.orbs.vuurwerk.nl [194.178.232.55]
Jul 17 18:34:25 mailhost sendmail[16943]: WAA16943: from=<orbs.org!sender>, size=0, class=0, pri=0, nrcpts=0, proto=SMTP, relay=relaytest.orbs.vuurwerk.nl [194.178.2
Jul 17 18:34:48 mailhost sendmail[16945]: WAA16945: ruleset=check_mail, arg1=<sender>, relay=relaytest.orbs.vuurwerk.nl [194.178.232.55], reject=553 <sender>... Doma
Jul 17 18:34:48 mailhost sendmail[16945]: WAA16945: from=<sender>, size=0, class=0, pri=0, nrcpts=0, proto=SMTP, relay=relaytest.orbs.vuurwerk.nl [194.178.232.55]
Jul 17 18:35:14 mailhost sendmail[16946]: WAA16946: ruleset=check_rcpt, arg1=<orbs-relaytest@manawatu.co.nz>, relay=relaytest.orbs.vuurwerk.nl [194.178.232.55], reje
Jul 17 18:35:14 mailhost sendmail[16946]: WAA16946: lost input channel from relaytest.orbs.vuurwerk.nl [194.178.232.55]
Jul 17 18:35:14 mailhost sendmail[16946]: WAA16946: from=<>, size=0, class=0, pri=0, nrcpts=0, proto=SMTP, relay=relaytest.orbs.vuurwerk.nl [194.178.232.55]
Jul 17 18:35:50 mailhost sendmail[16949]: WAA16949: ruleset=check_rcpt, arg1=<orbs-relaytest@manawatu.co.nz>, relay=relaytest.orbs.vuurwerk.nl [194.178.232.55], reje
Jul 17 18:35:50 mailhost sendmail[16949]: WAA16949: lost input channel from relaytest.orbs.vuurwerk.nl [194.178.232.55]
Jul 17 18:35:50 mailhost sendmail[16949]: WAA16949: from=<sender@256.0.0.1>, size=0, class=0, pri=0, nrcpts=0, proto=SMTP, relay=relaytest.orbs.vuurwerk.nl [194.178.
Jul 17 18:36:09 mailhost sendmail[16950]: WAA16950: ruleset=check_rcpt, arg1=<orbs-relaytest@manawatu.co.nz>, relay=relaytest.orbs.vuurwerk.nl [194.178.232.55], reje
Jul 17 18:36:10 mailhost sendmail[16950]: WAA16950: lost input channel from relaytest.orbs.vuurwerk.nl [194.178.232.55]
Jul 17 18:36:10 mailhost sendmail[16950]: WAA16950: from=<sender@[256.0.0.1]>, size=0, class=0, pri=0, nrcpts=0, proto=SMTP, relay=relaytest.orbs.vuurwerk.nl [194.17
Jul 17 18:36:41 mailhost sendmail[16952]: WAA16952: ruleset=check_rcpt, arg1=<orbs-relaytest@manawatu.co.nz>, relay=relaytest.orbs.vuurwerk.nl [194.178.232.55], reje
Jul 17 18:36:41 mailhost sendmail[16952]: WAA16952: lost input channel from relaytest.orbs.vuurwerk.nl [194.178.232.55]
Jul 17 18:36:41 mailhost sendmail[16952]: WAA16952: from=<sender@mailhost.my.domain>, size=0, class=0, pri=0, nrcpts=0, proto=SMTP, relay=relaytest.orbs.vuurwerk.nl
```

3.5.1 Source of trace

A monitored network where the sensor is located between all external firewalls and the router to the ISP.

3.5.2 Detect was generated by

A tcpdump filter to watch all traffic going to a low-traffic email server. It was detected manually by noticing the sheer number of connections coming from one site without any traffic in between.

tcpdump format

```
hh:mm:ss.SSSSSS Source.SrcPort > Destination.DstPort: ProtocolSpecificData
```

maillog format

```
MMM DD hh:mm:ss hostname sendmail[ID]: XXXID: SendmailReportingInfo
```

3.5.3 Probability the source address was spoofed

Most definitely not, as it completed multiple full TCP connections.

3.5.4 Description of attack

An automated script is used to try various permutations of "from" addresses, to see if the mail server will remail the message. The fact that the source host resolves to **relaytest.orbs.vuurwerk.nl** is a good indicator as to the motives of the operation.

3.5.5 Attack mechanism

The attack works by taking advantage of mail servers that haven't been properly configured to only relay local mail, or for servers that are deliberately configured to allow relaying from anywhere.

3.5.6 Correlations

This is such a common occurrence that CIAC has a whole page devoted to it. CIAC is the Computer Incident Advisory Capability division of the Department of Energy.

3.5.7 Evidence of active targeting

Hit a mail server with a listed MX record for the domain being monitored.

3.5.8 Severity

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

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

System Criticality	4	This is the incoming mail server for the domain, and people get touchy when mail service is interrupted.
Attack Lethality	2	This attack is not attempting to compromise the system perse, rather it is attempting to get it to do something that it probably shouldn't. It would be lower than a 2 except that it's possible to use a relay under certain conditions to make it look like controversial email is coming from someone within the targeted domain.
Network Countermeasures	1	The network is not filtering out smtp content, and aside from passive monitoring there are no other network countermeasures in place for normal smtp traffic.
System Countermeasures	5	This mail server has a hardened OS, and has been configured to not do remailing from any host, even internal ones. The above type of attack can never be successful on such a system without first modifying the machine via some other attack.

3.5.9 Defensive recommendation

Have two mail servers. One that can only receive mail, and one that can only relay mail, where the relay server is preferably inside a firewall that only allows trusted users access to it.

3.5.10 Multiple choice test question

```
18:22:04.723696 194.178.232.55.2488 > mailhost.smtp: S 4132059863:4132059863(0) win 32120 <mss 1460,sackOK,timestamp 2921939[|tcp]> (DF)
18:22:04.726714 mailhost.smtp > 194.178.232.55.2488: S 448162290:448162290(0) ack 4132059864 win 32120 <mss 1460,sackOK,timestamp 183739012[|tcp
18:22:04.827885 194.178.232.55.2488 > mailhost.smtp: . ack 1 win 32120 <nop,nop,timestamp 2921950 183739012> (DF)

This trace contains

   a) a valid 3-way handshake.
   b) invalid non-matching sequence numbers.
   c) an invalid source port of 2488.
   d) a valid push of TCP payload data.

answer: a
```

Start of Detect TOC

3.6 Detect 6

tcpdump output day 1

```
04:09:08.674921 205.178.5.151.20766 > 256.0.0.1.31125: R 0:0(0) ack 956906616 win 0 (ttl 238, id 2881)
```

tcpdump output day 2

```
17:56:45.152694 193.231.249.2.20766 > 256.0.0.1.46997: R 0:0(0) ack 878029944 win 0 (ttl 116, id 23828)
21:02:27.319179 63.200.206.202.20766 > 256.0.0.1.46741: R 0:0(0) ack 1002890360 win 0 (ttl 48, id 23828)
```

3.6.1 Source of trace

A monitored network where the sensor is located between all external firewalls and the router to the ISP.

3.6.2 Detect was generated by

This trace was detected by a tcpdump filter that searches for the reset flag being set.

tcpdump format

```
hh:mm:ss.SSSSSS Source.SrcPort > Destination.DstPort: ProtocolSpecificData
```

3.6.3 Probability the source address was spoofed

I honestly don't know for sure, but I'm guessing that the source address was not spoofed, and that this is collateral damage from a scan or attack on these other networks and that somehow the monitored host's IP address was associated with port 20766.

3.6.4 Description of attack

I suspect that the monitored host was being used as a decoy in a scan, or attack, being done by somebody else on the source machines. It appears to be a normal reset, so I rather suspect that it was a scan and not an attack, but I may never know.

It is possible that the attacker actually has control of the source machines and is doing a very slow scan of the monitored host, but none of the destination ports are associated with any known trojans, and even the source port isn't associated with anything.

3.6.5 Attack mechanism

This doesn't appear to be an attack, but rather a scan, but I'm still not certain of the purpose of the scan and which machines are the final targets.

3.6.6 Correlations

I could find no correlations that mimic this behavior quite this way. It looks like a normal decoy scan, but then in a way it doesn't. I'm not quite sure what category this falls into, as it could be an attempt to scan the target machine, or as a decoy to the real scan hitting the source machines. There just isn't enough data here to tell for sure.

Here is at least one type of similar attack http://www.sans.org/y2k/053100-1200.htm, but it doesn't quite have the same characteristics.

3.6.7 Evidence of active targeting

Only one machine was targeted, but it's possible that the machine was gleaned from a list of active machines and put into some hacker scanning script.

3.6.8 Severity

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

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

System Criticality	5	This is a primary DNS server, and strange traffic going to it is always a concern.
Attack Lethality	1	Aside from being unsolicited, there were no odd flags set in the packet.
Network Countermeasures	5	All unsolicited packets are blocked to non-DNS ports.
System Countermeasures	5	This machine is very secure, and would be able to hold its own even without additional network protection.

3.6.9 Defensive recommendation

Put a stateful firewall between the Internet and all machines that need to be protected, so that unsolicited traffic of any kind will be blocked.

3.6.10 Multiple choice test question

```
17:56:45.152694 193.231.249.2.20766 > 256.0.0.1.46997: R 0:0(0) ack 878029944 win 0 (ttl 116, id 23828)
21:02:27.319179 63.200.206.202.20766 > 256.0.0.1.46741: R 0:0(0) ack 1002890360 win 0 (ttl 48, id 23828)

What is clearly anomalous about this trace

   a) the time between the packets.
   b) the sequence numbers do not match.
   c) the source hosts are different.
   d) the initiating SYN packets are missing.

answer: d
```

Start of Detect TOC

3.7 Detect 7

tcpdump output

```
05:42:08.941414 61.138.15.98.1877 > 256.0.0.10.32773: S 3332313668:3332313668(0) win 32120 <mss 1460,sackOK,timestamp 45939120[|tcp]> (DF)
05:42:08.975894 61.138.15.98.1909 > 256.0.0.11.32773: S 3327298105:3327298105(0) win 32120 <mss 1460,sackOK,timestamp 45939120[|tcp]> (DF)
05:42:09.023654 61.138.15.98.1942 > 256.0.0.12.32773: S 3328092373:3328092373(0) win 32120 <mss 1460,sackOK,timestamp 45939120[|tcp]> (DF)
```

3.7.1 Source of trace

A monitored network where the sensor is located between all external firewalls and the router to the ISP.

3.7.2 Detect was generated by

A tcpdump filter that checks for all incoming packets that have the SYN flag set detected this.

tcpdump format

```
hh:mm:ss.SSSSSS Source.SrcPort > Destination.DstPort: ProtocolSpecificData
```

3.7.3 Probability the source address was spoofed

Almost nil, as this is a TCP packet attempting to initiate a 3-way handshake.

3.7.4 Description of attack

This appears to be a scan for SunRPC services, as port 32773 has a high probability of getting "randomly" assigned on Solaris system. Apparently Tooltalk often ends up at that port, and it listens for TCP connections.

3.7.5 Attack mechanism

Connect to a Tooltalk service from which there are known exploits that allow the attacker to take over control of the system.

3.7.6 Correlations

As of May 30 of this year, people are still reporting scans to this port, and it's no wondering considering the Stack Overflow in ToolTalk RPC Service problem.

3.7.7 Evidence of active targeting

Semi-active targeting at best. Only three machines were scanned, and none of them would be running SunRPC services at the expected Solaris range.

3.7.8 Severity

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

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

System Criticality	4	The three machines that were attempting to be scanned are all important servers, but it could just be a coincidence as they were scanned

		in ascending IP ordering and the potential attacker may have only picked a small range within which to scan.
Attack Lethality	5	This appears to be a probe for SunRPC services running at port 32773, but it could simply be a scan to attempt to see if hosts are alive so lethality is hard to really tack down, but when in doubt, I've rated it the worst.
Network Countermeasures	5	All unsolicited packets to dynamic ports are blocked by a stateful firewall.
System Countermeasures	4	None of the machines that were scanned are running any variant of the Solaris operating system. I would have given this a 5 if I were absolutely certain that SunRPC services are what this scan was really targeting.

3.7.9 Defensive recommendation

The best defense is to block all unsolicited packets to unserved ports, which this should be considered. Also, all vulnerable machines should have the latest patches applied as you can never tell when something will get through your perimeter defenses.

3.7.10 Multiple choice test question

```
05:42:08.941414 61.138.15.98.1877 > 256.0.0.10.32773: S 3332313668:3332313668(0) win 32120 <mss 1460,sackOK,timestamp 45939120[|tcp]> (DF)
05:42:08.975894 61.138.15.98.1909 > 256.0.0.11.32773: S 3327298105:3327298105(0) win 32120 <mss 1460,sackOK,timestamp 45939120[|tcp]> (DF)
05:42:09.023654 61.138.15.98.1942 > 256.0.0.12.32773: S 3328092373:3328092373(0) win 32120 <mss 1460,sackOK,timestamp 45939120[|tcp]> (DF)
```

This is an example of a) a TCP retry. b) valid non-repeating source port numbers. c) invalid non-repeating sequence numbers. d) a 3-way handshake. answer: b

Start of Detect TOC

3.8 Detect 8

tcpdump output

```
03:28:25.879121 63.253.248.251 > 256.0.0.1: icmp: echo request
03:28:25.879566 256.0.0.1 > 63.253.248.251: icmp: echo reply (DF)
03:28:26.166152 63.253.248.251 > 256.0.0.2: icmp: echo request
03:28:26.195439 63.253.248.251 > 256.0.0.3: icmp: echo request
03:28:27.619888 63.253.248.251.1190 > 256.0.0.1.www: S 434904503:434904503(0) win 16384 <mss 1460,nop,nop,sackOK> (DF)
03:28:27.620270 256.0.0.1.www > 63.253.248.251.1190: R 0:0(0) ack 434904504 win 0 (DF)
03:28:27.651815 63.253.248.251.1191 > 256.0.0.1.socks: S 434946017:434946017(0) win 16384 <mss 1460,nop,nop,sackOK> (DF)
03:28:27.652079 256.0.0.1.socks > 63.253.248.251.1191: R 0:0(0) ack 434946018 win 0 (DF)
03:28:27.679238 63.253.248.251 > 256.0.0.4: icmp: echo request
03:28:27.850348 63.253.248.251 > 256.0.0.5: icmp: echo request
03:28:27.852564 256.0.0.5 > 63.253.248.251: icmp: echo reply
03:28:27.870354 63.253.248.251 > 256.0.0.6: icmp: echo request
03:28:27.890793 63.253.248.251 > 256.0.0.7: icmp: echo request
03:28:27.907045 63.253.248.251 > 256.0.0.8: icmp: echo request
03:28:28.192421 63.253.248.251 > 256.0.0.9: icmp: echo request
03:28:28.369546 63.253.248.251.1190 > 256.0.0.1.www: S 434904503:434904503(0) win 16384 <mss 1460,nop,nop,sackOK> (DF)
03:28:28.369834 256.0.0.1.www > 63.253.248.251.1190: R 0:0(0) ack 1 win 0 (DF)
03:28:28.402543 63.253.248.251 > 256.0.0.10: icmp: echo request
03:28:28.442658 63.253.248.251.1191 > 256.0.0.1.socks: S 434946017:434946017(0) win 16384 <mss 1460,nop,nop,sackOK> (DF)
03:28:28.442908 256.0.0.1.socks > 63.253.248.251.1191: R 0:0(0) ack 1 win 0 (DF)
03:28:28.518844 63.253.248.251 > 256.0.0.11: icmp: echo request
03:28:29.105682 63.253.248.251.1191 > 256.0.0.1.socks: S 434946017:434946017(0) win 16384 <mss 1460,nop,nop,sackOK> (DF)
03:28:29.106077 256.0.0.1.socks > 63.253.248.251.1191: R 0:0(0) ack 1 win 0 (DF)
03:28:29.110614 63.253.248.251.1190 > 256.0.0.1.www: S 434904503:434904503(0) win 16384 <mss 1460,nop,nop,sackOK> (DF)
03:28:29.110859 256.0.0.1.www > 63.253.248.251.1190: R 0:0(0) ack 1 win 0 (DF)
03:28:29.675881 63.253.248.251.1192 > 256.0.0.5.www: S 435525705:435525705(0) win 16384 <mss 1460,nop,nop,sackOK> (DF)
03:28:29.676059 256.0.0.5.www > 63.253.248.251.1192: R 0:0(0) ack 435525706 win 0
03:28:29.676740 63.253.248.251.1193 > 256.0.0.5.socks: S 435571805:435571805(0) win 16384 <mss 1460,nop,nop,sackOK> (DF)
03:28:29.676905 256.0.0.5.socks > 63.253.248.251.1193: R 0:0(0) ack 435571806 win 0
03:28:29.841966 63.253.248.251 > 256.0.0.12: icmp: echo request
03:28:30.431777 63.253.248.251.1192 > 256.0.0.5.www: S 435525705:435525705(0) win 16384 <mss 1460,nop,nop,sackOK> (DF)
03:28:30.431942 256.0.0.5.www > 63.253.248.251.1192: R 0:0(0) ack 1 win 0
03:28:30.473019 63.253.248.251.1193 > 256.0.0.5.socks: S 435571805:435571805(0) win 16384 <mss 1460,nop,nop,sackOK> (DF)
03:28:30.473177 256.0.0.5.socks > 63.253.248.251.1193: R 0:0(0) ack 1 win 0
03:28:31.099371 63.253.248.251.1192 > 256.0.0.5.www: S 435525705:435525705(0) win 16384 <mss 1460,nop,nop,sackOK> (DF)
03:28:31.099543 256.0.0.5.www > 63.253.248.251.1192: R 0:0(0) ack 1 win 0
03:28:31.190132 63.253.248.251.1193 > 256.0.0.5.socks: S 435571805:435571805(0) win 16384 <mss 1460,nop,nop,sackOK> (DF)
03:28:31.190299 256.0.0.5.socks > 63.253.248.251.1193: R 0:0(0) ack 1 win 0
03:29:27.307350 63.253.248.251.1200 > 256.0.0.5.socks: S 450285517:450285517(0) win 16384 <mss 1460,nop,nop,sackOK> (DF)
03:29:27.307554 256.0.0.5.socks > 63.253.248.251.1200: R 0:0(0) ack 450285518 win 0
03:29:28.019622 63.253.248.251.1200 > 256.0.0.5.socks: S 450285517:450285517(0) win 16384 <mss 1460,nop,nop,sackOK> (DF)
03:29:28.019821 256.0.0.5.socks > 63.253.248.251.1200: R 0:0(0) ack 1 win 0
03:29:28.691085 63.253.248.251.1200 > 256.0.0.5.socks: S 450285517:450285517(0) win 16384 <mss 1460,nop,nop,sackOK> (DF)
03:29:28.691253 256.0.0.5.socks > 63.253.248.251.1200: R 0:0(0) ack 1 win 0
03:29:32.319630 63.253.248.251.1208 > 256.0.0.1.socks: S 451990281:451990281(0) win 16384 <mss 1460,nop,nop,sackOK> (DF)
03:29:32.320109 256.0.0.1.socks > 63.253.248.251.1208: R 0:0(0) ack 451990282 win 0 (DF)
03:29:33.018225 63.253.248.251.1208 > 256.0.0.1.socks: S 451990281:451990281(0) win 16384 <mss 1460,nop,nop,sackOK> (DF)
03:29:33.018562 256.0.0.1.socks > 63.253.248.251.1208: R 0:0(0) ack 1 win 0 (DF)
03:29:33.714703 63.253.248.251.1208 > 256.0.0.1.socks: S 451990281:451990281(0) win 16384 <mss 1460,nop,nop,sackOK> (DF)
03:29:33.715035 256.0.0.1.socks > 63.253.248.251.1208: R 0:0(0) ack 1 win 0 (DF)
```

3.8.1 Source of trace

A monitored network where the sensor is located between all external firewalls and the router to the ISP.

3.8.2 Detect was generated by

Initially detected by a filter searching for incoming pings and incoming SYN packets. A quick filter was devised to give the entire trace.

tcpdump format

```
hh:mm:ss.SSSSSS Source.SrcPort > Destination.DstPort: ProtocolSpecificData
```

3.8.3 Probability the source address was spoofed

The attacker was looking for a ping response and later a TCP connection, so there is little to no chance the source address was spoofed.

3.8.4 Description of attack

An echo request is sent to each host in ascending order. If an echo reply is received, then a TCP connection to port 80 and port 1080 is attempted.

3.8.5 Attack mechanism

Somewhat unclear, as the attacker never hit a machine that would respond to a TCP connection to the http (80) or socks (1080) ports. The script that was used required an initial ping response, and our web servers are configured to not respond to pings.

3.8.6 Correlations

These were normal pings accompanied by normal TCP connection attempts, so the attack would have to be something targeted towards a web or socks server. As those connections never fully were made, I cannot really say which attack was being initiated. I do know that it was extremely novice, as the initial ping actually thwarted them from even hitting a real web server.

3.8.7 Evidence of active targeting

The entire network of live hosts was scanned in an incremental fashion.

3.8.8 Severity

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

(4 + 2) - (5 + 3) = -2

System Criticality	4	Some of the machines targeted were very critical, but the scan was clearly not targeted, and none of the machines running web servers are critical.
Attack Lethality	2	Since only a scan actually occurred, the lethality is given a low rating. If actual content were to have actually been sent, then it would likely be rated much higher.
Network Countermeasures	5	All unsolicited packets to dynamic ports are blocked by a stateful firewall, as are icmp echo requests.
System Countermeasures	3	The machines that have web servers are hardened for web attacks, and the machines without them will not respond to a web or socks request.

3.8.9 Defensive recommendation

Only allow those machines to respond to pings that are really necessary, and do a pre-emptive scan of your web site to make sure that there aren't any cgi vulnerabilities, assuming this is what the person was really going to attack.

3.8.10 Multiple choice test question

```
03:28:27.679238 63.253.248.251 > web.server: icmp: echo request
03:28:27.850348 63.253.248.251 > proxy.firewall: icmp: echo request
03:28:27.852564 proxy.firewall > 63.253.248.251: icmp: echo reply
03:28:27.870354 63.253.248.251 > protected.machine: icmp: echo request
03:28:30.431777 63.253.248.251.1192 > proxy.firewall.www: S 435525705:435525705(0) win 16384 <mss 1460,nop,nop,sackOK> (DF)
03:28:30.431942 proxy.firewall.www > 63.253.248.251.1192: R 0:0(0) ack 1 win 0

What is strange about this scan?

   a) The web server never responds to the TCP connection.
   b) The sequence numbers in the SYN packet are identical.
   c) The scan requires an echo reply before initiating a TCP connection.
   d) The SYN packet has the do not fragment flag set.

answer: c
```

Start of Detect TOC

3.9 Detect 9

tcpdump output

```
02:32:01.465556 144.232.9.193 > 256.0.0.1: icmp: host 212.35.146.123 unreachable
```

3.9.1 Source of trace

A monitored network where the sensor is located between all external firewalls and the router to the ISP.

3.9.2 Detect was generated by

A tcpdump filter looking for any incoming icmp packets.

tcpdump format

```
hh:mm:ss.SSSSSS Source.SrcPort > Destination.DstPort: ProtocolSpecificData
```

3.9.3 Probability the source address was spoofed

There is a very low probability that the source address is spoofed as it corresponds to a real router that could very likely be notifying someone of not being able to reach the 212.35.146.0 network.

3.9.4 Description of attack

Most likely a reverse network mapping attempt, to determine what machines there are to strike at, and the local IP address was spoofed by the attacker to help cover their tracks.

Since the 212.35.146.123 address corresponds to an IP address owned by the Romanian National Computer Network for Research and Education and can see why the attacker would want to throw some misdirection at them.

3.9.5 Attack mechanism

Send multiple scan attempts at the target network with several spoofed addresses so the attackee cannot tell for sure where the actual scan is coming from.

3.9.6 Correlations

Let's see, nmap has several documents on the web describing how to use decoy scans effectively. There are even documents concerned with trying to analyze NMAP Decoy Storms to determine the real culprit of the scan.

3.9.7 Evidence of active targeting

None, as we are almost certainly being used as a decoy. The IP address has an arp entry, but there is no real machine, so it makes a great decoy if that is what someone wanted to do.

3.9.8 Severity

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

(1 + 1) - (5 + 5) = -8

System Criticality	1	How can you kill a machine that doesn't exist.
Attack Lethality	1	This doesn't even appear to be any kind of attack on this machine, rather a side-effect.
Network Countermeasures	5	All unsolicited icmp replies are filtered out by a stateful firewall.
System Countermeasures	5	The machine doesn't actually exist.

3.9.9 Defensive recommendation

I'm not sure there is all that much that the decoy site can do, short of limiting the number of machines that ever show any behavior of initiating a ping. Of course, with spoofing rampant, the most effective thing is to get ISPs to properly configure their routers to block spoofed packets from ever getting out of their network.

3.9.10 Multiple choice test question

```
02:32:01.465556 144.232.9.193 > 256.0.0.1: icmp: host 212.35.146.123 unreachable

What initially triggered this response from 144.232.9.193?

   a) an echo request.
   b) an echo reply.
   c) a UDP packet.
   d) a TCP packet.

answer: a
```

Start of Detect TOC

3.10 Detect 10

3.10.6.1 tcpdump output

```
14:29:02.926534 24.7.64.145 > 256.0.0.1: icmp: time exceeded in-transit for 256.0.0.1.55690 > 24.8.89.239.32443: [|tcp] [ttl 1] (id 43796) (ttl 246, id 0)
```

3.10.1 Source of trace

A monitored network where the sensor is located between all external firewalls and the router to the ISP.

3.10.2 Detect was generated by

A tcpdump filter looking for any incoming icmp packets.

tcpdump format

```
hh:mm:ss.SSSSSS Source.SrcPort > Destination.DstPort: ProtocolSpecificData
```

3.10.3 Probability the source address was spoofed

Almost none, as 24.7.64.145 corresponds to a live router.

3.10.4 Description of attack

This could be at least two things. The first is a simply decoy in a network scanning attempt. The second could be a blatant attempt to send back a malformed tcp packet to the spoofed host IP. Something is very bothersome about TCP port 32443 being used, since that is more commonly associated with traceroute and UDP packets, not TCP.

3.10.5 Attack mechanism

If it is a simple network scan, then it is a spoofed IP decoy mask, but if it is something more sinister, then I don't know, as I didn't receive the who TCP packet that was returned due to a too short snaplen in tcpdump.

3.10.6 Correlations

I found several time exceeded in-transit citings, even from SANS, but none of them showed the embedded contents, or they only had icmp or UDP data embedded.

The only use I saw for embedding TCP packets was to disrupt an existing TCP connection to attempt to break in, but that clearly wasn't the case here.

3.10.7 Evidence of active targeting

None, as we are almost certainly being used as a decoy. The IP address has an arp entry, but there is no real machine, so it makes a great decoy if that is what someone wanted to do.

3.10.8 Severity

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

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

System Criticality	1	How can you kill a machine that doesn't exist.
Attack Lethality	5	This doesn't even appear to be any kind of attack on this machine, but I'm not sure. This could be some kind of attack on a machine that can't handle icmp responses to machines it never sent data to.
Network Countermeasures	5	All unsolicited icmp replies are filtered out by a stateful firewall.
System Countermeasures	5	The machine doesn't actually exist.

3.10.9 Defensive recommendation

Again, there is little that can be done short of getting all ISPs to fix their routers so that only non-spoofed traffic leaves the sites they manage.

In this instance though, I would highly recommend that a stateful firewall be put into place to make sure that packets like these never get back to a real host, as the embedded contents could be of very malicious intent.

3.10.10 Multiple choice test question

```
14:29:02.926534 24.7.64.145 > 256.0.0.1: icmp: time exceeded in-transit for 256.0.0.1.55690 > 24.8.89.239.32443: [|tcp] [ttl 1] (id 43796) (ttl

What initially triggered this response from 144.232.9.193?

   a) an echo request.
   b) an echo reply.
   c) a UDP packet.
   d) a TCP packet.

answer: d
```

Start of Detect TOC

---

4 Evaluate an Attack

The attack to be examined is a scenario in operating system determination reconnaissance. This allows a more targeted attack in the future if an attack and compromise is indeed the desired end result.

Location of Reconnaissance Tool

NMAP can be found at http://www.nmap.org/nmap/index.html.

Command Used

```
nmap -vv -n -O -p 53,80 -P0 target.A
```

The **-vv** option is used to give the most verbose output.
The **-n** option tells nmap to not bother doing a DNS lookup of the IP.
The **-O** option is to do Operating System profiling.
The **-p** option is used to specify just the specific ports to probe.
The **-P0** option tells nmap not to do an initial ping.

Description of How Specific Reconnaissance Technique Works

The operating system profiling technique that NMAP uses is to send various TCP packets to ports on the target machine, and change the TCP flags and optional arguments so that it can get different responses and hopefully be able to determine the responding operating system.

NMAP specifically examines changes in the window size, options returned, and the sequence numbers that are generated. It also attempts to do interesting things with invalid flags, to see whether the machine responds or not.

Annotated Network Trace of Reconnaissance in Action

Although specified second, port 80 is the first port attempted to be scanned.

```
15:54:35.409684 nmap.host.17983 > target.A.www: S 3545234541:3545234541(0) win 512 <mss 1460>
15:54:38.407300 nmap.host.17983 > target.A.www: S 3545234541:3545234541(0) win 32120 <mss 1460>
```

Notice that even though the first two attempted connects are normal TCP SYN requests, but that the window size is set to wildly different sizes. Sometimes just the response to differing window sizes is enough to determine the OS.

It is really hoping that the target system will negotiate for a larger window size, as happens below.

```
15:55:06.102598 nmap.host.18013 > target.A.domain: S 1020791811:1020791811(0) win 512 <mss 1460>
15:55:06.103858 target.A.domain > nmap.host.18013: S 2555548784:2555548784(0) ack 1020791812 win 32120 <mss 1460> (DF)
15:55:06.104121 nmap.host.18013 > target.A.domain: . ack 1 win 32120 (DF)
15:55:06.104383 nmap.host.18013 > target.A.domain: F 1:1(0) ack 1 win 32120
15:55:06.104746 target.A.domain > nmap.host.18013: . ack 2 win 32120 (DF)
15:55:06.104860 target.A.domain > nmap.host.18013: F 1:1(0) ack 2 win 32120 (DF)
15:55:06.105118 nmap.host.18013 > target.A.domain: . ack 2 win 32120 (DF)
```

Well, this is nice. Not only does it respond nicely with a SYN-ACK, it bumps up the return window size, presumably with its default window size. At this point nmap closes down the connection gracefully and begins the process of sending invalid or at least unusual TCP packets.

Below I've broken things up slightly out of order so that each stimulus and response can be looked at independently.

```
15:55:06.109670 nmap.host.61833 > target.A.domain: S [ECN-Echo] 486802280:486802280(0) win 4096 <wscale 10,nop,mss 265,timestamp 1061109567 0,eol>
15:55:06.111120 target.A.domain > nmap.host.61833: S 2560529854:2560529854(0) ack 486802281 win 32595 <mss 265,nop,nop,timestamp 277897817 1061109567,nop,wscale 0> (
15:55:06.111353 nmap.host.61833 > target.A.domain: R 486802281:486802281(0) win 0
```

Apparently tcpdump knows about some of nmap's tricks, as it has labeled this connection an ECN-Echo. The important things to note here are what fields do and do not get returned, and what their values are. Again notice that the window size of 4096 elicits a larger return window size of 32595.

The wscale option is set to 10, but gets returned with a value of 0. That must be important. The timestamp seems to be honored, and a useful profiling value is probably being returned. Hmm, now why are there more nop fields returned than sent, I'm sure it makes a difference. And finally, notice how that trailing eol just seems to be ignored, or maybe it was turned into a nop in the return packet. nmap seems happy enough, so it sends a reset to close the connection quickly.

```
15:55:06.109821 nmap.host.61834 > target.A.domain: . win 4096 <wscale 10,nop,mss 265,timestamp 1061109567 0,eol>
15:55:06.739004 nmap.host.61834 > target.A.domain: . win 4096 <wscale 10,nop,mss 265,timestamp 1061109567 0,eol>
```

Huh, no flags set at all, but a bunch of options are set. Apparently, the operating system being tested doesn't like that at all and never replies to that connection attempt.

```
15:55:06.109967 nmap.host.61835 > target.A.domain: SFP 486802280:486802280(0) win 4096 urg 0 <wscale 10,nop,mss 265,timestamp 1061109567 0,eol>
15:55:06.111863 target.A.domain > nmap.host.61835: S 2556435319:2556435319(0) ack 486802281 win 32595 <mss 265,nop,nop,timestamp 277897817 1061109567,nop,wscale 0> (
15:55:06.112095 nmap.host.61835 > target.A.domain: R 486802281:486802281(0) win 0
```

Now this is interesting. A SYN, FIN, PSH, with the urgent bit set seems to be something the OS is willing to respond to. Apparently all it cares about is that the SYN flag is set and the ACK flag is not. It is even gracious enough to not send back any of the invalid flags that were set upon receipt.

```
15:55:06.110116 nmap.host.61836 > target.A.domain: . ack 0 win 4096 <wscale 10,nop,mss 265,timestamp 1061109567 0,eol>
15:55:06.739152 nmap.host.61836 > target.A.domain: . ack 1 win 4096 <wscale 10,nop,mss 265,timestamp 1061109567 0,eol>
```

Yep, it looks like this operating system really wants to see a SYN flag set before it responds, as it completely ignores the lone ACKs.

Ok, enough of the real funny stuff, now it's time to send a flurry of packets at it in a SYN, SYN-ACK, RST pattern as fast as we can. This lessens to probability that someone else will get a connection in between all of our requests. Why is this important? Well, nmap is now going to try to do some sequence number prediction.

```
15:55:07.369019 nmap.host.61827 > target.A.domain: S 486802281:486802281(0) win 4096
15:55:07.369833 target.A.domain > nmap.host.61827: S 2564245393:2564245393(0) ack 486802282 win 32696 <mss 536> (DF)
15:55:07.370072 nmap.host.61827 > target.A.domain: R 486802282:486802282(0) win 0
15:55:07.388998 nmap.host.61828 > target.A.domain: S 486802282:486802282(0) win 4096
15:55:07.389688 target.A.domain > nmap.host.61828: S 2556278846:2556278846(0) ack 486802283 win 32696 <mss 536> (DF)
15:55:07.389934 nmap.host.61828 > target.A.domain: R 486802283:486802283(0) win 0
15:55:07.408997 nmap.host.61829 > target.A.domain: S 486802283:486802283(0) win 4096
15:55:07.409698 target.A.domain > nmap.host.61829: S 2563185125:2563185125(0) ack 486802284 win 32696 <mss 536> (DF)
15:55:07.409935 nmap.host.61829 > target.A.domain: R 486802284:486802284(0) win 0
15:55:07.429003 nmap.host.61830 > target.A.domain: S 486802284:486802284(0) win 4096
15:55:07.429702 target.A.domain > nmap.host.61830: S 2561867378:2561867378(0) ack 486802285 win 32696 <mss 536> (DF)
15:55:07.429941 nmap.host.61830 > target.A.domain: R 486802285:486802285(0) win 0
15:55:07.449001 nmap.host.61831 > target.A.domain: S 486802285:486802285(0) win 4096
15:55:07.449709 target.A.domain > nmap.host.61831: S 2562391321:2562391321(0) ack 486802286 win 32696 <mss 536> (DF)
15:55:07.449953 nmap.host.61831 > target.A.domain: R 486802286:486802286(0) win 0
```

```
15:55:07.469013 nmap.host.61832 > target.A.domain: S 486802286:486802286(0) win 4096
15:55:07.469703 target.A.domain > nmap.host.61832: S 2551723519:2551723519(0) ack 486802287 win 32696 <mss 536> (DF)
15:55:07.469943 nmap.host.61832 > target.A.domain: R 486802287:486802287(0) win 0
```

The sequence goes, 2565245393, 2556278846, 2563185125, 2561867378, 2562391321, and 2551723519. Hmm, it doesn't mean a whole heck of a lot to me, but nmap states that a very large random integer is being added, and that determining the next sequence number is very hard at best. Of course, at this point nmap only cares about what machines use that type of sequence generation, not whether it will be easy to exploit or not.

Top of Evaluate TOC

---

5 "Analyze This" Scenario

Listed below are several machines that should be looked at for signs of existing Trojans, or they are leaking out potentially valuable information to a hacker trying to break into a system.

SNMP traffic using the public password is insecure no matter what and should be fixed. Internal machines doing massive scans of other internal machines or networks is generally not normal behavior. Having machines that respond to Netbios name service requests out in the open is just asking for trouble, as sooner or later they will get broken into.

MY.NET.253.12 Anomalous Behavior

MY.NET.253.12 is a machine doing some very unusual things. In May, it scanned various different internal networks. Unless this machine was purposely used for such things, I would say without a doubt that it has been compromised, as this is clearly not the type of behavior that a machine without hacker tools on it would exhibit.

SunRPC Scan of other internal networks

Below is shown an internal scan of port 32771 on the MY.NET.16.0 network and even though it isn't shown, the same thing happened on the MY.NET.19.0 network and the MY.NET.101.0 network. All of these connections came from the MY.NET.253.12 machine, which may be indicative of a compromise.

```
SnortA7.txt:05/28-14:30:50.876461  [**] SUNRPC highport access! [**] MY.NET.253.12:43746 -> MY.NET.16.0:32771
SnortA7.txt:05/28-14:30:51.185774  [**] SUNRPC highport access! [**] MY.NET.253.12:43747 -> MY.NET.16.0:32771
SnortA7.txt:05/28-14:31:04.905230  [**] SUNRPC highport access! [**] MY.NET.253.12:43749 -> MY.NET.16.0:32771
SnortA7.txt:05/28-14:31:05.245775  [**] SUNRPC highport access! [**] MY.NET.253.12:43750 -> MY.NET.16.0:32771
SnortA7.txt:05/28-14:34:34.562778  [**] SUNRPC highport access! [**] MY.NET.253.12:43746 -> MY.NET.16.3:32771
SnortA7.txt:05/28-14:34:34.860094  [**] SUNRPC highport access! [**] MY.NET.253.12:43747 -> MY.NET.16.3:32771
SnortA7.txt:05/28-14:34:48.011149  [**] SUNRPC highport access! [**] MY.NET.253.12:43749 -> MY.NET.16.3:32771
SnortA7.txt:05/28-14:34:48.331077  [**] SUNRPC highport access! [**] MY.NET.253.12:43750 -> MY.NET.16.3:32771
 ...
    (addresses between MY.NET.16.3 and MY.NET.16.162)
 ...
SnortA7.txt:05/28-23:56:06.805430  [**] SUNRPC highport access! [**] MY.NET.253.12:43747 -> MY.NET.16.162:32771
SnortA7.txt:05/28-23:56:20.647562  [**] SUNRPC highport access! [**] MY.NET.253.12:43749 -> MY.NET.16.162:32771
SnortA7.txt:05/28-23:56:20.965297  [**] SUNRPC highport access! [**] MY.NET.253.12:43750 -> MY.NET.16.162:32771

SnortA13.txt:05/29-00:06:16.141491  [**] SUNRPC highport access! [**] MY.NET.253.12:43746 -> MY.NET.16.165:32771
SnortA13.txt:05/29-00:06:16.496510  [**] SUNRPC highport access! [**] MY.NET.253.12:43747 -> MY.NET.16.165:32771
SnortA13.txt:05/29-00:06:31.060697  [**] SUNRPC highport access! [**] MY.NET.253.12:43749 -> MY.NET.16.165:32771
SnortA13.txt:05/29-00:06:31.390550  [**] SUNRPC highport access! [**] MY.NET.253.12:43750 -> MY.NET.16.165:32771
 ...
    (addresses between MY.NET.16.165 and MY.NET.16.255)
 ...
SnortA13.txt:05/29-06:18:13.492204  [**] SUNRPC highport access! [**] MY.NET.253.12:43749 -> MY.NET.16.255:32771
SnortA13.txt:05/29-06:18:13.815027  [**] SUNRPC highport access! [**] MY.NET.253.12:43750 -> MY.NET.16.255:32771
```

Random TCP port scan of host MY.NET.14.1

This is an extremely fast, and very random, scan of TCP ports on the machine MY.NET.14.1. This is a very good indicator that NMAP is running on this machine. A total of 1141 ports were scanned in about 5 seconds.

```
SnortS7.txt:May 27 23:44:42 MY.NET.253.12:43746 -> MY.NET.14.1:93 SYN **S*****
SnortS7.txt:May 27 23:44:42 MY.NET.253.12:43746 -> MY.NET.14.1:669 SYN **S*****
 ...
    (random set of destination ports between 1 and 43188)
 ...
SnortS7.txt:May 27 23:44:45 MY.NET.253.12:12045 -> MY.NET.14.1:79 SYN **S*****
SnortS7.txt:May 27 23:44:47 MY.NET.253.12:48810 -> MY.NET.14.1:2001 SYN **S*****
SnortS7.txt:May 27 23:44:47 MY.NET.253.12:47851 -> MY.NET.14.1:6001 SYN **S*****
```

And for good measure there were some invalid flags in packets sent to the echo and tcpmux ports at the end of the scan.

```
SnortS7.txt:May 27 23:44:47 MY.NET.253.12:43753 -> MY.NET.14.1:7 SYN 2*S***** RESERVEDBITS
SnortS7.txt:May 27 23:44:47 MY.NET.253.12:43754 -> MY.NET.14.1:7 NULL ********
SnortS7.txt:May 27 23:44:47 MY.NET.253.12:43755 -> MY.NET.14.1:7 NMAPID **SF*P*U
SnortS7.txt:May 27 23:44:47 MY.NET.253.12:43757 -> MY.NET.14.1:1 SYN **S*****
SnortS7.txt:May 27 23:44:47 MY.NET.253.12:43759 -> MY.NET.14.1:1 XMAS ***F*P*U
SnortS7.txt:May 27 23:44:47 MY.NET.253.12:43746 -> MY.NET.14.1:1 UDP
SnortS7.txt:May 27 23:44:47 MY.NET.253.12:43752 -> MY.NET.14.1:7 SYN **S*****
```

Probable machine profiling

Snort seems pretty sure that NMAP is likely being used to profile various internal systems. Now that's very important information for an attacker to know.

```
SnortA6.txt:05/27-23:44:47.357835  [**] Null scan! [**] MY.NET.253.12:43754 -> MY.NET.14.1:7
SnortA6.txt:05/27-23:44:47.358118  [**] Probable NMAP fingerprint attempt [**] MY.NET.253.12:43755 -> MY.NET.14.1:7
SnortA7.txt:05/28-14:32:56.087358  [**] Null scan! [**] MY.NET.253.12:43754 -> MY.NET.16.1:7
SnortA7.txt:05/28-14:32:56.087697  [**] Probable NMAP fingerprint attempt [**] MY.NET.253.12:43755 -> MY.NET.16.1:7
SnortA7.txt:05/28-14:33:06.464897  [**] Null scan! [**] MY.NET.253.12:43754 -> MY.NET.16.2:21
SnortA7.txt:05/28-14:33:06.465190  [**] Probable NMAP fingerprint attempt [**] MY.NET.253.12:43755 -> MY.NET.16.2:21
```

```
SnortA11.txt:05/28-14:32:56.087358  [**] Null scan! [**] MY.NET.253.12:43754 -> MY.NET.16.1:7
SnortA11.txt:05/28-14:32:56.087697  [**] Probable NMAP fingerprint attempt [**] MY.NET.253.12:43755 -> MY.NET.16.1:7
SnortA11.txt:05/28-14:33:06.464897  [**] Null scan! [**] MY.NET.253.12:43754 -> MY.NET.16.2:21
SnortA11.txt:05/28-14:33:06.465190  [**] Probable NMAP fingerprint attempt [**] MY.NET.253.12:43755 -> MY.NET.16.2:21
SnortA12.txt:05/28-14:32:56.087358  [**] Null scan! [**] MY.NET.253.12:43754 -> MY.NET.16.1:7
SnortA12.txt:05/28-14:32:56.087697  [**] Probable NMAP fingerprint attempt [**] MY.NET.253.12:43755 -> MY.NET.16.1:7
SnortA12.txt:05/28-14:33:06.464897  [**] Null scan! [**] MY.NET.253.12:43754 -> MY.NET.16.2:21
SnortA12.txt:05/28-14:33:06.465190  [**] Probable NMAP fingerprint attempt [**] MY.NET.253.12:43755 -> MY.NET.16.2:21
SnortA13.txt:05/29-07:32:41.151584  [**] Null scan! [**] MY.NET.253.12:43754 -> MY.NET.19.10:23
SnortA13.txt:05/29-07:32:41.151883  [**] Probable NMAP fingerprint attempt [**] MY.NET.253.12:43755 -> MY.NET.19.10:23
SnortA13.txt:05/29-07:32:44.619571  [**] Null scan! [**] MY.NET.253.12:43754 -> MY.NET.19.10:23
SnortA13.txt:05/29-07:32:51.529293  [**] Null scan! [**] MY.NET.253.12:43754 -> MY.NET.19.10:23
SnortA13.txt:05/29-07:32:51.529932  [**] Probable NMAP fingerprint attempt [**] MY.NET.253.12:43755 -> MY.NET.19.10:23
SnortA13.txt:05/29-07:32:54.961835  [**] Null scan! [**] MY.NET.253.12:43754 -> MY.NET.19.10:23
SnortA13.txt:05/29-07:33:05.034209  [**] Null scan! [**] MY.NET.253.12:43754 -> MY.NET.19.10:23
SnortA14.txt:05/29-07:32:41.151584  [**] Null scan! [**] MY.NET.253.12:43754 -> MY.NET.19.10:23
SnortA14.txt:05/29-07:32:41.151883  [**] Probable NMAP fingerprint attempt [**] MY.NET.253.12:43755 -> MY.NET.19.10:23
SnortA14.txt:05/29-07:32:44.619571  [**] Null scan! [**] MY.NET.253.12:43754 -> MY.NET.19.10:23
SnortA14.txt:05/29-07:32:51.529293  [**] Null scan! [**] MY.NET.253.12:43754 -> MY.NET.19.10:23
SnortA14.txt:05/29-07:32:51.529932  [**] Probable NMAP fingerprint attempt [**] MY.NET.253.12:43755 -> MY.NET.19.10:23
SnortA14.txt:05/29-07:32:54.961835  [**] Null scan! [**] MY.NET.253.12:43754 -> MY.NET.19.10:23
SnortA14.txt:05/29-07:33:05.034209  [**] Null scan! [**] MY.NET.253.12:43754 -> MY.NET.19.10:23
SnortA15.txt:05/31-14:49:23.623629  [**] Null scan! [**] MY.NET.253.12:43754 -> MY.NET.101.1:23
SnortA15.txt:05/31-14:49:23.623677  [**] Probable NMAP fingerprint attempt [**] MY.NET.253.12:43755 -> MY.NET.101.1:23
SnortA15.txt:05/31-14:49:28.733386  [**] Null scan! [**] MY.NET.253.12:43754 -> MY.NET.101.1:23
SnortA15.txt:05/31-14:49:42.555591  [**] Null scan! [**] MY.NET.253.12:43754 -> MY.NET.101.1:23
SnortA15.txt:05/31-14:49:51.419781  [**] Null scan! [**] MY.NET.253.12:43754 -> MY.NET.101.1:23
SnortA15.txt:05/31-14:49:51.421208  [**] Probable NMAP fingerprint attempt [**] MY.NET.253.12:43755 -> MY.NET.101.1:23
SnortA15.txt:05/31-14:49:56.535537  [**] Null scan! [**] MY.NET.253.12:43754 -> MY.NET.101.1:23
SnortA15.txt:05/31-22:11:09.250639  [**] Null scan! [**] MY.NET.253.12:43754 -> MY.NET.101.89:21
SnortA15.txt:05/31-22:11:09.250959  [**] Probable NMAP fingerprint attempt [**] MY.NET.253.12:43755 -> MY.NET.101.89:21
SnortA15.txt:05/31-22:11:11.071978  [**] Null scan! [**] MY.NET.253.12:43754 -> MY.NET.101.89:21
SnortA15.txt:05/31-22:11:11.072031  [**] Probable NMAP fingerprint attempt [**] MY.NET.253.12:43755 -> MY.NET.101.89:21
SnortA15.txt:05/31-22:11:39.719288  [**] Null scan! [**] MY.NET.253.12:43754 -> MY.NET.101.90:7
SnortA15.txt:05/31-22:11:39.719584  [**] Probable NMAP fingerprint attempt [**] MY.NET.253.12:43755 -> MY.NET.101.90:7
SnortA15.txt:05/31-22:11:46.628887  [**] Null scan! [**] MY.NET.253.12:43754 -> MY.NET.101.90:7
SnortA15.txt:05/31-22:11:46.628939  [**] Probable NMAP fingerprint attempt [**] MY.NET.253.12:43755 -> MY.NET.101.90:7
SnortA15.txt:05/31-22:11:50.338368  [**] Null scan! [**] MY.NET.253.12:43754 -> MY.NET.101.90:7
SnortA15.txt:05/31-22:11:50.338426  [**] Probable NMAP fingerprint attempt [**] MY.NET.253.12:43755 -> MY.NET.101.90:7
SnortA15.txt:05/31-23:30:24.956759  [**] Null scan! [**] MY.NET.253.12:43754 -> MY.NET.101.115:7
SnortA15.txt:05/31-23:30:24.956810  [**] Probable NMAP fingerprint attempt [**] MY.NET.253.12:43755 -> MY.NET.101.115:7
SnortA15.txt:05/31-23:30:26.738561  [**] Null scan! [**] MY.NET.253.12:43754 -> MY.NET.101.115:7
SnortA15.txt:05/31-23:30:26.740181  [**] Probable NMAP fingerprint attempt [**] MY.NET.253.12:43755 -> MY.NET.101.115:7
SnortA15.txt:05/31-23:30:44.135133  [**] Null scan! [**] MY.NET.253.12:43754 -> MY.NET.101.117:7
SnortA15.txt:05/31-23:30:44.135423  [**] Probable NMAP fingerprint attempt [**] MY.NET.253.12:43755 -> MY.NET.101.117:7
```

MY.NET.1.3 Anomalous Behavior

This machine has a scan originating on at least May 24th that continues up until the latest logs ending at June 23rd. This slow UDP scan covers the machine MY.NET.101.89 primarily, but it touches MY.NET.101.140, MY.NET.101.142, MY.NET.101.141, and MY.NET.101.158

Single MY.NET.101.158 Scan

What is odd is that the single scan to MY.NET.101.158 was to port 53, where as all the other scans were to high ports.

```
SnortS2.txt:May 25 15:08:21 MY.NET.1.3:53 -> MY.NET.101.158:53 UDP
```

Snippet of massive scan of MY.NET.101.89

While somewhat random, but usually in bunches of eight in a row, none of the destination port numbers are ever repeated in this slow scan. This means that either state is being kept between process shutdown, or that this process has been running non-stop continuously over this entire period of time which is very scary that something like this could go unnoticed this long.

```
SnortSca.txt:May 24 17:11:30 MY.NET.1.3:53 -> MY.NET.101.89:57009 UDP
SnortSca.txt:May 24 17:11:30 MY.NET.1.3:53 -> MY.NET.101.89:57010 UDP
SnortSca.txt:May 24 17:11:30 MY.NET.1.3:53 -> MY.NET.101.89:57011 UDP
 ...
    (many days of probing left out)
 ...
SnortS27.txt:Jun 23 15:10:31 MY.NET.1.3:53 -> MY.NET.101.89:47476 UDP
SnortS27.txt:Jun 23 15:10:31 MY.NET.1.3:53 -> MY.NET.101.89:47477 UDP
SnortS27.txt:Jun 23 15:10:31 MY.NET.1.3:53 -> MY.NET.101.89:47478 UDP
```

MY.NET.70.234 Interesting Behavior

While internal file sharing technically isn't supposed to be dangerous, if somebody is able to get a sniffer inside of your protected area, then they may find these non-advertised machines and be able to break into them regardless.

Netbios Name Query

This could be perfectly normal behavior if somebody on the machine MY.NET.70.234 was attempting to connect to the other machines simply by IP addresses if they are in separate Windows Domains. Still, the fact that all of the machines replied means that they might respond to an external request.

Network traffic seems to imply that MY.NET.101.0 is not advertised to the outside world, but both MY.NET.101.89 and MY.NET.101.55 have been scanned by the aforementioned potentially compromised machines, MY.NET.1.3 and MY.NET.253.12

```
SnortA7.txt:05/28-18:44:40.135907  [**] SMB Name Wildcard [**] MY.NET.70.234:137 -> MY.NET.101.55:137
SnortA7.txt:05/28-18:44:40.137976  [**] SMB Name Wildcard [**] MY.NET.101.55:137 -> MY.NET.70.234:137
SnortA7.txt:05/28-18:45:06.557464  [**] SMB Name Wildcard [**] MY.NET.70.234:137 -> MY.NET.101.89:137
SnortA7.txt:05/28-18:45:06.630606  [**] SMB Name Wildcard [**] MY.NET.101.89:137 -> MY.NET.70.234:137
```

```
SnortA7.txt:05/28-18:45:54.040922  [**] SMB Name Wildcard [**] MY.NET.70.234:137 -> MY.NET.101.145:137
SnortA7.txt:05/28-18:45:54.041819  [**] SMB Name Wildcard [**] MY.NET.101.145:137 -> MY.NET.70.234:137
SnortA7.txt:05/28-18:46:15.377066  [**] SMB Name Wildcard [**] MY.NET.70.234:137 -> MY.NET.101.147:137
SnortA7.txt:05/28-18:46:15.382603  [**] SMB Name Wildcard [**] MY.NET.101.147:137 -> MY.NET.70.234:137
SnortA7.txt:05/28-18:46:36.701913  [**] SMB Name Wildcard [**] MY.NET.70.234:137 -> MY.NET.101.151:137
SnortA7.txt:05/28-18:46:36.703580  [**] SMB Name Wildcard [**] MY.NET.101.151:137 -> MY.NET.70.234:137
```

Open Access to MY.NET.101.192

Any SNMP traffic is a potential security whole, but this is the most dangerous since access is restricted via a known exploit.

SNMP Public Password Used

Open SNMP access between machines in the MY.NET.97.0 network and host MY.NET.101.192. Anyone who knew that MY.NET.101.192 accepted SNMP packets would be able to do whatever the permission allow, but at least be able to read information if nothing else.

```
SnortA17.txt:05/16-09:24:56.936732  [**] SNMP public access [**] MY.NET.97.12:1055 -> MY.NET.101.192:161
SnortA2.txt:05/23-09:32:38.870319  [**] SNMP public access [**] MY.NET.97.129:1056 -> MY.NET.101.192:161
SnortA2.txt:05/23-10:44:29.333012  [**] SNMP public access [**] MY.NET.97.87:1252 -> MY.NET.101.192:161
SnortA2.txt:05/23-13:27:23.054450  [**] SNMP public access [**] MY.NET.97.133:1046 -> MY.NET.101.192:161
SnortA5.txt:05/25-09:49:50.437373  [**] SNMP public access [**] MY.NET.97.100:1053 -> MY.NET.101.192:161
SnortA5.txt:05/25-11:37:24.262886  [**] SNMP public access [**] MY.NET.97.76:1046 -> MY.NET.101.192:161
SnortA5.txt:05/25-13:32:24.475803  [**] SNMP public access [**] MY.NET.97.60:1060 -> MY.NET.101.192:161
SnortA5.txt:05/25-19:35:01.278977  [**] SNMP public access [**] MY.NET.97.190:1051 -> MY.NET.101.192:161
SnortA5.txt:05/25-20:04:49.337958  [**] SNMP public access [**] MY.NET.97.226:1122 -> MY.NET.101.192:161
SnortA10.txt:05/26-08:01:02.137479  [**] SNMP public access [**] MY.NET.97.59:1054 -> MY.NET.101.192:161
SnortA10.txt:05/26-20:54:34.196420  [**] SNMP public access [**] MY.NET.97.81:1050 -> MY.NET.101.192:161
SnortA11.txt:05/28-13:16:08.362466  [**] SNMP public access [**] MY.NET.97.52:1213 -> MY.NET.101.192:161
SnortA11.txt:05/28-16:05:38.579105  [**] SNMP public access [**] MY.NET.97.222:1053 -> MY.NET.101.192:161
SnortA11.txt:05/28-16:55:23.200035  [**] SNMP public access [**] MY.NET.97.74:1236 -> MY.NET.101.192:161
SnortA11.txt:05/28-18:52:40.278416  [**] SNMP public access [**] MY.NET.97.63:1053 -> MY.NET.101.192:161
SnortA14.txt:05/29-18:27:19.423820  [**] SNMP public access [**] MY.NET.97.183:1051 -> MY.NET.101.192:161
SnortA22.txt:06/18-16:01:50.420609  [**] SNMP public access [**] MY.NET.97.248:1037 -> MY.NET.101.192:161
SnortA22.txt:06/18-18:33:19.338888  [**] SNMP public access [**] MY.NET.97.215:1042 -> MY.NET.101.192:161
SnortA23.txt:06/19-10:29:11.511106  [**] SNMP public access [**] MY.NET.97.199:1071 -> MY.NET.101.192:161
SnortA25.txt:06/22-17:49:16.218149  [**] SNMP public access [**] MY.NET.97.37:1047 -> MY.NET.101.192:161
SnortA28.txt:05/22-17:01:15.642482  [**] SNMP public access [**] MY.NET.97.203:1365 -> MY.NET.101.192:161
```

Top of Analyze TOC

Last Updated: July 25th, 2017

Upcoming Training

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