Global Information Assurance Certification Paper

GIAC Intrusion Detection Analysis
Practical Assignment for SNAP San Jose '2000
GCIA Certification
Submitted by Chris Grout

Detect #1

```
[**] IDS216/ICMP subnet mask request [**]
05/31-13:40:20.628464 208.45.147.85 -> 256.46.217.256
ICMP TTL:253 TOS:0x0 ID:1028  DF
ADDRESS REQUEST

[**] SNMP public access [**]
05/31-13:40:20.631789 208.45.147.85:61454 -> 256.46.217.256:161
UDP TTL:62 TOS:0x0 ID:1029  DF
Len: 44

 [**] IDS171/Ping zeros [**]
05/31-13:41:23.956294 208.45.147.85 -> 256.46.217.256
ICMP TTL:253 TOS:0x0 ID:1052  DF
ID:5138   Seq:35009  ECHO

[**] IDS171/Ping zeros [**]
05/31-13:46:23.319508 208.45.147.85 -> 256.46.217.256
ICMP TTL:253 TOS:0x0 ID:1053  DF
ID:5138   Seq:35422  ECHO

[**] IDS171/Ping zeros [**]
05/31-13:51:23.326596 208.45.147.85 -> 256.46.217.256
ICMP TTL:253 TOS:0x0 ID:1054  DF
ID:5138   Seq:35744  ECHO

[**] IDS246/large-icmp [**]
05/31-13:56:24.195556 208.45.147.85 -> 256.46.217.256
ICMP TTL:253 TOS:0x0 ID:1025  DF
ID:39612   Seq:57072  ECHO

[**] IDS171/Ping zeros [**]
05/31-13:56:24.195566 208.45.147.85 -> 256.46.217.256
ICMP TTL:253 TOS:0x0 ID:1026
ID:5138   Seq:36067  ECHO

[**] IDS171/Ping zeros [**]
05/31-14:01:24.129256 208.45.147.85 -> 256.46.217.256
ICMP TTL:253 TOS:0x0 ID:1027  DF
ID:5138   Seq:36391  ECHO
```

1. Source of trace: These detections were observed while monitoring a group of servers at a co-location facility.

2. System(s) generating detects: These alerts were recorded using Snort 1.6 in conjunction with what was a current copy of the arachNIDS signature file.

3. Spoofed Probability: Very unlikely. The attacker makes no real attempt to mask this scan. The intention here appears to be a combination of network reconnaissance and some sort of monitoring/testing. Also a trace back to the source address shows the same TTL (253), and even though this certainly cannot be called conclusive, it does add a bit of strength to my belief.

4. Description of attack: The above alerts are a condensed representation of what amounted to (18) consecutive SNMP queries using "public" as the community string and (6) ICMP subnet mask requests. Immediately afterwards the source began sending a pattern of (4) "Ping zeros", one every five minutes and a single "large-icmp" packet every twenty minutes.

5. Attack mechanism: After contacting the owner of the source IP, it was discovered that our "friendly" ISP that runs the co-location facility decided to add one of our servers to their HP Openview system. Apparently by default, when adding a system to HP Openview, it attempts to auto-discover further information through the SNMP and ICMP subnet mask requests.

6. Correlations: I was not able to find any other references to this specific pattern. I was able find information specific to the different alerts that does further support the ISP's explanation.

- IDS216/ICMP subnet mask request
- IDS171/Ping zeros
- IDS246/large-icmp

7. Evidence of active targeting: Yes. Only one system was targeted, and it just so happened to also be the only system that responds to ICMP echo requests. This system is not configured to use SNMP. Since the activity started, there was no other traffic from this system to any other system on our network. It is assumed that prior to these alerts, the source scanned for systems that would respond to ICMP echo requests and therefore eventually targeted only this one system.

8. Severity: -3

- **Criticality 4** – This type of scan is normally targeted towards routers, switches and other hardware devices.
- **Lethality 2** – Information gathering/confidentiality attack
- **System Countermeasures 4** – The server has a current operating system, all current patches installed, and does not run any SNMP services.
- **Network Countermeasures 3** – Firewalls and routers block SNMP traffic. This ICMP traffic is allowed to pass.

9. Defense recommendation: Defenses were sufficient. Routers and Firewalls should be configured to block all unnecessary ICMP traffic.

10. What does the above trace probably show?

a) Denial of Service
b) ICMP buffer overflow
c) Misconfigured management device
d) "Slow and low" network reconnaissance

Answer: c

Detect #2

```
[**] SMB Name Wildcard [**]
05/26-17:02:40.985172 208.223.125.119:1045 -> 256.46.217.xx1:137
UDP TTL:112 TOS:0x0 ID:5511
Len: 58

[**] SMB Name Wildcard [**]
05/26-17:02:48.855817 208.223.125.119:1045 -> 256.46.217.xx2:137
UDP TTL:112 TOS:0x0 ID:37767
Len: 58

[**] SMB Name Wildcard [**]
05/26-17:02:59.405655 208.223.125.119:1045 -> 256.46.217.xx3:137
UDP TTL:112 TOS:0x0 ID:15752
Len: 58

 [**] SMB Name Wildcard [**]
05/26-17:03:09.911184 208.223.125.119:1045 -> 256.46.217.xx4:137
UDP TTL:112 TOS:0x0 ID:59272
Len: 58

 [**] SMB Name Wildcard [**]
05/26-17:03:20.460761 208.223.125.119:1045 -> 256.46.217.xx5:137
UDP TTL:112 TOS:0x0 ID:37257
Len: 58

 [**] SMB Name Wildcard [**]
05/26-17:03:30.995862 208.223.125.119:1024 -> 256.46.217.xx6:137
UDP TTL:112 TOS:0x0 ID:15242
Len: 58
```

1. Source of trace: These detections were observed while monitoring a group of our servers at a co-location facility.

2. System(s) generating detects: These alerts were recorded using Snort 1.6 in conjunction with what was a current copy of the arachNIDS signature file.

3. Spoofed Probability: Unlikely. This appears to be a fast and very unsophisticated scan. The attacker makes no real attempt to mask this scan.

4. Description of attack: The attack was a scan for systems that respond to a standard Netbios name table retrieval query. Netbios, which is used by Microsoft operating systems as well as Samba servers, has a vast array of known vulnerabilities and should be protected at all costs. This service is the "Achilles' Heel" of Microsoft systems. These queries are identified quite easily because they always contain the same payload pattern as in the below example:

```
06/08-00:23:44.807585 4.17.177.34:137 -> 216.103.247.256:137
UDP TTL:114 TOS:0x0 ID:16209
Len: 58
CC D6 00 10 00 01 00 00 00 00 00 00 20 43 4B 41  ............ CKA
41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41  AAAAAAAAAAAAAAAA
41 41 41 41 41 41 41 41 41 41 41 41 41 00 00 21  AAAAAAAAAAAAA..!
```

00 01

5. Attack mechanism: This attack was obviously done by some sort of script or application due to the speed at which the scan was done. The attacking system was scanning at the rate of approximately one class "C" subnet every 45 seconds. Since this scan appears to have been done so carelessly I would guess that this was a "script-kiddie" running Legion, NAT, or some other similar application. These types of queries can also be launched manually by running the "nbtstat –a *xxx.xxx.xxx.xxx*" on Microsoft based systems or by "nmblookup –A xxx.xxx.xxx.xxx" on *nix systems running Samba services.

An easy way to automate a scan such as this one would be to use a simple script such as the one below.

```
For /L %x in (1,1,254) do nbtstat –A 123.123.123.%x
```

This would cause the entire class "C" subnet to be scanned sequentially. The output could also be piped to a text file.

6. Correlations: This pre-attack scan is extremely common and usually the sign of an unsophisticated attacker searching for easy targets. Traffic to UDP 137 in general is fairly common due to the noisy natural of most Microsoft systems. For example, many MS Exchange servers will attempt Netbios lookups on each other when delivering mail between the two for some reason. But in this specific case, the speed at which the queries came in and the fact that the destination addresses were successively incremented, proves this scan was definitely a deliberate network reconnaissance attempt.

7. Evidence of active targeting: No. The attacker scanned my entire range, including non-existent systems. Probably scanning large random blocks of addresses.

8. Severity: 1

- **Criticality 3** – Apparent blind sweep which includes routers, servers and other critical systems.
- **Lethality 4** – Unsecured systems allowing Netbios access are vulnerable to everything from numerous denial of service attacks, information gathering, and access elevation exploits.
- **System Countermeasures 2** – All routers, switches and servers are running current operating systems and patches. Microsoft and Samba systems do allow access to Netbios services due to interoperability needs.
- **Network Countermeasures 4** – Firewalls and routers block all Netbios traffic. Any attempts are logged even though this traffic is quite common.

9. Defense recommendation: Defenses are currently sufficient. However, could further increase security by implementing "IP filtering" on all necessary systems to allow Netbios traffic only between trusted hosts.

10. What does the above trace probably show?

a) Denial of Service
b) Trojan horse traffic
c) Half-open NMAP scan
d) Legion/NAT scan

Answer: d

Detect #3

```
[**] Source Port traffic [**]
05/26-18:19:41.672916 134.76.247.8:53 -> 256.46.217.xx1:53
TCP TTL:21 TOS:0x0 ID:39426
**SF**** Seq: 0x3989DD36   Ack: 0x57121E84   Win: 0x404

[**] Source Port traffic [**]
05/26-18:19:41.694800 134.76.247.8:53 -> 256.46.217.xx2:53
TCP TTL:21 TOS:0x0 ID:39426
**SF**** Seq: 0x3989DD36   Ack: 0x57121E84   Win: 0x404

[**] Source Port traffic [**]
05/26-18:19:41.714156 134.76.247.8:53 -> 256.46.217.xx3:53
TCP TTL:21 TOS:0x0 ID:39426
**SF**** Seq: 0x3989DD36   Ack: 0x57121E84   Win: 0x404


[**] Source Port traffic [**]
05/28-10:56:33.124236 203.149.232.20:53 -> 256.46.217.xx1:53
TCP TTL:24 TOS:0x0 ID:39426
**SF**** Seq: 0x71AB232D   Ack: 0x3D631EFB   Win: 0x404

[**] Source Port traffic [**]
05/28-10:56:33.127743 203.149.232.20:53 -> 256.46.217.xx2:53
TCP TTL:24 TOS:0x0 ID:39426
**SF**** Seq: 0x71AB232D   Ack: 0x3D631EFB   Win: 0x404

[**] Source Port traffic [**]
05/28-10:56:33.141373 203.149.232.20:53 -> 256.46.217.xx3:53
TCP TTL:24 TOS:0x0 ID:39426
**SF**** Seq: 0x71AB232D   Ack: 0x3D631EFB   Win: 0x404


[**] Source Port traffic [**]
05/30-02:43:34.360304 203.149.232.20:53 -> 256.46.217.xx1:53
TCP TTL:24 TOS:0x0 ID:39426
**SF**** Seq: 0x465437C5   Ack: 0x6D3FF95A   Win: 0x404

[**] Source Port traffic [**]
05/30-02:43:34.380384 203.149.232.20:53 -> 256.46.217.xx2:53
TCP TTL:24 TOS:0x0 ID:39426
**SF**** Seq: 0x465437C5   Ack: 0x6D3FF95A   Win: 0x404

[**] Source Port traffic [**]
05/30-02:43:34.412772 203.149.232.20:53 -> 256.46.217.xx3:53
TCP TTL:24 TOS:0x0 ID:39426
**SF**** Seq: 0x465437C5   Ack: 0x6D3FF95A   Win: 0x404
```

1. Source of trace: These detections were observed while monitoring a group of our servers at a co-location facility.

2. System(s) generating detects: These alerts were recorded using Snort 1.6 in conjunction with what was a current copy of the arachNIDS signature file.

3. Spoofed Probability: Unlikely. These scans appear to be very fast, loud and very unsophisticated. On the other hand, if these scans were seen concurrently, I would tend to believe that two of the three might have been spoofed.

4. Description of attack: Even though Snort alerted on this traffic as being "Source Port Traffic" a better diagnosis would have been a SYN/FIN port scan. The "Source Port Traffic" detect is logged when a packet is received that appears to be trying to exploit an vulnerability which is common with old or misconfigured firewalls that allows inbound traffic from "DNS servers" to ports otherwise normally denied. A recent example of this type of vulnerability was recently discovered with ZoneLab's ZoneAlarm personal firewall product.

There are a few reasons why this traffic is considered "interesting". First of all is the SYN/FIN scan. This alone is a major red flag as to someone should look into what's going on here. Secondly is that this packet is trying to access TCP port 53, which is only used for zone transfers and on very rare occasions, very large DNS query responses. And lastly is the fact that the sequence, acknowledgement, and packet ID's are identical across multiple connection attempts.

Linux systems do in fact respond to SYN/FIN packets with SYN/FIN/ACK's. Therefore the script may be written to specifically find systems running this operating system. Also, even though the SYN/FIN flag combination is completely bogus and is normally dropped by most firewalls, some router based ACL's allow this type of traffic to pass, such as Fore's PowerHub 7000 series routers.

More than likely this scan was used to search for DNS servers vulnerable to the many buffer overflow attacks against pre v.8.2.2 patch level 5 BIND implementations.

CVE-1999-0833 Buffer overflow in BIND 8.2 via NXT records
CVE-1999-0009 Inverse query buffer overflow in BIND 4.9 and BIND 8 Releases
CVE-1999-0835 Denial of service in BIND named via malformed SIG records
CVE-1999-0848 Denial of service in BIND named via consuming more than "fdmax" file descriptors.
CVE-1999-0849 Denial of service in BIND named via maxdname
CVE-1999-0851 Denial of service in BIND named via naptr

5. Attack mechanism: Since approximately May 30, 2000, an enormous number of these types of scans originating in China, Taiwan, and Malaysia have been reported. I tend to believe that a script might have been developed to automate these types of scans and that it was initially released somewhere in the Asian region.

6. Correlations: This type of scanning has become extremely popular over the last month. These scans, including some with the same source IP's, have been confirmed by Dforster, Dwhite, and Rhys of the Snort Forums, Exploit Discussion group.

7. Evidence of active targeting: No. The attackers scanned my entire range, including non-existent systems. Probably scanning large random blocks of addresses.

8. Severity: 1

- **Criticality 3** – Confirmed blind sweep, which includes both critical and non-critical systems.
- **Lethality 5** – DNS servers running outdated versions of BIND are vulnerable to root level compromise.
- **System Countermeasures 4** – All servers are current with both application and OS patches.
- **Network Countermeasures 4** – Firewalls and routers block TCP port 53 from all systems not authorized for zone transfers. Firewalls and routers have also been tested for "source port traffic" vulnerabilities and to ensure SYN/FIN packets are dropped.

9. Defense recommendation: Defenses currently are sufficient. All future traffic from the source IP's should be logged.

10. What does the above trace probably show?

a) A misconfigured DNS server
b) DNS buffer overflow
c) Denial of Service
d) Scripted DNS scan

Answer: d

Detect #4

```
[**] SNMP public access [**]
05/29-03:23:48.583664 208.29.1.49:1039 -> 256.46.217.xx9:161
UDP TTL:16 TOS:0x0 ID:37478
Len: 52
30 2A 02 01 00 04 06 70 75 62 6C 69 63 A0 1D 02  0*.....public...
01 02 02 01 00 02 01 00 30 12 30 10 06 0C 2B 06  ........0.0...+.
01 04 01 0B 02 04 03 0A 06 00 05 00              ............

[**] SNMP public access [**]
05/29-03:23:48.584094 208.29.1.49:1039 -> 256.46.217.xx8:161
UDP TTL:16 TOS:0x0 ID:37734
Len: 52
30 2A 02 01 00 04 06 70 75 62 6C 69 63 A0 1D 02  0*.....public...
01 02 02 01 00 02 01 00 30 12 30 10 06 0C 2B 06  ........0.0...+.
01 04 01 0B 02 04 03 0A 06 00 05 00              ............

[**] SNMP public access [**]
05/29-03:23:48.584540 208.29.1.49:1039 -> 256.46.217.xx7:161
UDP TTL:16 TOS:0x0 ID:37990
Len: 52
30 2A 02 01 00 04 06 70 75 62 6C 69 63 A0 1D 02  0*.....public...
01 02 02 01 00 02 01 00 30 12 30 10 06 0C 2B 06  ........0.0...+.
01 04 01 0B 02 04 03 0A 06 00 05 00              ............

[**] SNMP public access [**]
05/29-03:23:48.585015 208.29.1.49:1039 -> 256.46.217.xx6:161
UDP TTL:16 TOS:0x0 ID:38246
Len: 52
30 2A 02 01 00 04 06 70 75 62 6C 69 63 A0 1D 02  0*.....public...
01 02 02 01 00 02 01 00 30 12 30 10 06 0C 2B 06  ........0.0...+.
01 04 01 0B 02 04 03 0A 06 00 05 00              ............
```

1. Source of trace: These detections were observed while monitoring a group of servers in a co-location facility.

2. System(s) generating detects: These alerts were recorded using Snort 1.6 in conjunction with what was a current copy of the arachNIDS signature file.

3. Spoofed Probability: Unlikely. This scan was executed very quickly and there has been no other attempts at this type of scan.

4. Description of attack: This scan was searching for devices that would respond to queries using "public" as the community string. "public" is usually the default for most devices and many people forget or never get around to changing this password. Devices responding to these queries can give out valuable information to the attacker, such as the type of device, its internal and external interface addresses, and OS level or firmware revision. This information can then be used to exploit known vulnerabilities with the specific devices. Another key element is that if the owner/admin left the read-only password as "public", there's a good chance that they also left the read-write password as the default.

5. Attack mechanism: Due to the speed at which the scan was done, I believe that this was a scripted scan. However, I cannot be sure that this scan was done with malicious intent, since it could also have been launched by a misconfigured network management software that was attempting to "auto-discover" valid devices.

6. Correlations: This type of scanning might have been revitalized due to the recent DOS attack on AboveNet. It is believed that the "hacker" may have compromised AboveNet's switches due to open SNMP configurations. The source address of this probe is no longer accessible and no prior history regarding this source IP was found. I've added this address to my "interesting" visitors in case they return.

7. Evidence of active targeting: Possible. Due to the fact that this was seen on a subnet used by a co-location facility, searching for SNMP devices might prove to return a fairly high success rate due to the high density of routers and switches.

8. Severity: -1

- **Criticality 5** – This type of scan is focused towards devices usually associated with connectivity services such as routers, switches and servers.
- **Lethality 3** – If an open system was found confidential information would be available.
- **System Countermeasures 5** – All unnecessary systems have SNMP services disabled. All community strings comply with accepted "strong password" standards.
- **Network Countermeasures 4** – Firewalls and routers block all SNMP traffic except to/from approved management stations. Firewalls and routers also log all unauthorized attempts.

9. Defense recommendation: Defenses are sufficient. Suggest monitoring the source IP for further suspicious active.

10. What does the above trace probably show?

a) Misdirected SNMP traps
b) SNMP buffer overflow
c) SNMP queries
d) Smurf attack

Answer: c

Detect #5

```
[Date]     [Time]  [Action]   [If]      [Prot]       [SourceIP]         [Dest IP]        [SrcPort] [DstPort] [Flags]

06/07/00 02:14 deny in eth0 60 tcp 20 48 204.203.63.117 204.210.6.256 1041 23 syn
06/07/00 02:14 deny in eth0 60 tcp 20 48 204.203.63.117 204.210.6.256 1041 23 syn

06/07/00 03:15 deny in eth0 60 tcp 20 48 204.203.63.117 204.210.6.256 1042 12345 syn
06/07/00 03:15 deny in eth0 60 tcp 20 48 204.203.63.117 204.210.6.256 1042 12345 syn
06/07/00 03:16 deny in eth0 60 tcp 20 48 204.203.63.117 204.210.6.256 1042 12345 syn

06/07/00 04:17 deny in eth0 60 tcp 20 48 204.203.63.117 204.210.6.256 1043 21 syn (FTP)
06/07/00 04:17 deny in eth0 60 tcp 20 48 204.203.63.117 204.210.6.256 1043 21 syn (FTP)
06/07/00 04:17 deny in eth0 60 tcp 20 48 204.203.63.117 204.210.6.256 1043 21 syn (FTP)
06/07/00 04:17 deny in eth0 60 tcp 20 48 204.203.63.117 204.210.6.256 1043 21 syn (FTP)

06/07/00 05:19 deny in eth0 60 tcp 20 48 204.203.63.117 204.210.6.256 1044 25 syn (SMTP)
06/07/00 05:19 deny in eth0 60 tcp 20 48 204.203.63.117 204.210.6.256 1044 25 syn (SMTP)
06/07/00 05:19 deny in eth0 60 tcp 20 48 204.203.63.117 204.210.6.256 1044 25 syn (SMTP)
06/07/00 05:19 deny in eth0 60 tcp 20 48 204.203.63.117 204.210.6.256 1044 25 syn (SMTP)

06/07/00 06:20 deny in eth0 60 tcp 20 48 204.203.63.117 204.210.6.256 1045 80 syn (HTTP)
06/07/00 06:20 deny in eth0 60 tcp 20 48 204.203.63.117 204.210.6.256 1045 80 syn (HTTP)
06/07/00 06:20 deny in eth0 60 tcp 20 48 204.203.63.117 204.210.6.256 1045 80 syn (HTTP)
06/07/00 06:20 deny in eth0 60 tcp 20 48 204.203.63.117 204.210.6.256 1045 80 syn (HTTP)
06/07/00 06:20 deny in eth0 60 tcp 20 48 204.203.63.117 204.210.6.256 1045 80 syn (HTTP)

06/08/00 21:34 deny in eth0 78 udp 20 108 204.203.63.117 204.210.6.256 137 137 (default)
06/08/00 21:34 deny in eth0 78 udp 20 108 204.203.63.117 204.210.6.256 137 137 (default)
06/08/00 21:34 deny in eth0 78 udp 20 108 204.203.63.117 204.210.6.256 137 137 (default)

06/08/00 21:35 deny in eth0 48 tcp 20 108 204.203.63.117 204.210.6.256 1491 445 syn
06/08/00 21:35 deny in eth0 48 tcp 20 108 204.203.63.117 204.210.6.256 1493 139 syn
06/08/00 21:35 deny in eth0 48 tcp 20 108 204.203.63.117 204.210.6.256 1491 445 syn
06/08/00 21:35 deny in eth0 48 tcp 20 108 204.203.63.117 204.210.6.256 1493 139 syn
06/08/00 21:35 deny in eth0 48 tcp 20 108 204.203.63.117 204.210.6.256 1491 445 syn
06/08/00 21:35 deny in eth0 48 tcp 20 108 204.203.63.117 204.210.6.256 1493 139 syn

06/08/00 21:35 deny in eth0 78 udp 20 108 204.203.63.117 204.210.6.256 137 137 (default)
06/08/00 21:35 deny in eth0 78 udp 20 108 204.203.63.117 204.210.6.256 137 137 (default)
06/08/00 21:35 deny in eth0 78 udp 20 108 204.203.63.117 204.210.6.256 137 137 (default)
```

1. **Source of trace:** A home system on a cable modem.

2. **System(s) generating detects:** Logs were generated from a WatchGuard FireBox II.

3. **Spoofed Probability:** Unlikely. More than likely this attacker thought he was evading the firewall's security by going "slow and low". Therefore it appears the attacker did not take any further efforts to hide their identity.

4. **Description of attack:** These probes were just basic SYN packets, which attempt to initiate a standard TCP 3-way handshake. If the service were available, a SYN/ACK packet would be sent back to the scanning system.

The attacker initially scanned some common ports with known vulnerabilities and Trojans. Then the following day came back and attempted a Netbios name query for some reason.

5. **Attack mechanism:** Due to the fact that the initial probes were almost exactly one hour apart my guess is that this was a scripted attack. Possibly done by a simple batch file or custom script. I'd also take a guess that the attacker's OS was Windows 98. Some versions of Windows 98 were known to have a bug where the clock would loose approximately one to two minutes per hour, which is consistent with this attacker's frequency.

Some attackers will attempt to spread out their attacks over a large period of time to avoid setting off intrusion detection systems which attempt to detect port scanning and also to avoid setting off routers and firewalls which are configured to automatically block traffic from port scanning systems. (Note: This is sometimes a bad thing to enable on a firewall since an attacker could spoof the address of a valid site, thus causing a denial of service attack against a valid network.)

The second set of probes could very easily have been an "nbtstat –a 204.210.6.256" followed by a "net view \\204.210.6.256".

6. Correlations: I do not know of a tool that has an option to scan once an hour. Therefore I tend to believe that this could have been a simple batch file or Perl script. This attacker is probably more than likely a "script kiddie" since an experienced attacker would of at least varied the interval.

7. Evidence of active targeting: Yes. The attacker may have previously scanned this system and noticed that the firewall does in fact block systems attempting to port scan and therefore attempted a very slow scan to avoid this.

8. Severity: -5

- **Criticality 2** – This system is a simple isolated workstation.
- **Lethality 2** – This system does have some of the scanned services running.
- **System Countermeasures 4** – Firewall and internal system's are running current operating systems with all current patches.
- **Network Countermeasures 5** – Firewall blocks all inbound traffic.

9. Defense recommendation: Defenses are sufficient. Continue logging all future traffic from the source IP.

10. What does the above detect probably show?

a) NMAP scan
b) Netbios "chatter"
c) Denial of service
d) Slow and low scan

Answer: d

Detect #6

```
FWIN,2000/05/22,13:13:30 -8:00 GMT,256.11.63.107:0,24.30.142.256:0,ICMP
FWIN,2000/05/22,13:13:30 -8:00 GMT,256.11.63.120:0,24.30.142.256:0,ICMP
FWIN,2000/05/22,13:13:44 -8:00 GMT,256.11.63.107:40286,24.30.142.256:23,TCP
FWIN,2000/05/22,13:13:44 -8:00 GMT,256.11.63.120:1058,24.30.142.256:23,TCP
FWIN,2000/05/22,13:13:44 -8:00 GMT,256.11.63.120:1059,24.30.142.256:23,TCP
FWIN,2000/05/22,13:14:00 -8:00 GMT,256.11.63.107:40287,24.30.142.256:23,TCP
FWIN,2000/05/22,13:14:00 -8:00 GMT,256.11.63.120:1060,24.30.142.256:23,TCP
FWIN,2000/05/22,13:14:00 -8:00 GMT,256.11.63.120:1061,24.30.142.256:23,TCP
FWIN,2000/05/22,13:14:16 -8:00 GMT,256.11.63.107:40288,24.30.142.256:23,TCP
FWIN,2000/05/22,13:14:16 -8:00 GMT,256.11.63.120:1062,24.30.142.256:23,TCP
FWIN,2000/05/22,13:14:16 -8:00 GMT,256.11.63.120:1063,24.30.142.256:23,TCP
FWIN,2000/05/22,13:14:30 -8:00 GMT,256.11.63.107:40286,24.30.142.256:80,TCP
FWIN,2000/05/22,13:14:30 -8:00 GMT,256.11.63.120:1058,24.30.142.256:80,TCP
FWIN,2000/05/22,13:14:30 -8:00 GMT,256.11.63.120:1059,24.30.142.256:80,TCP
FWIN,2000/05/22,13:15:00 -8:00 GMT,256.11.63.120:1060,24.30.142.256:80,TCP
FWIN,2000/05/22,13:15:00 -8:00 GMT,256.11.63.120:1061,24.30.142.256:80,TCP
FWIN,2000/05/22,13:15:00 -8:00 GMT,256.11.63.107:40287,24.30.142.256:80,TCP
FWIN,2000/05/22,13:15:16 -8:00 GMT,256.11.63.107:40288,24.30.142.256:80,TCP
FWIN,2000/05/22,13:15:16 -8:00 GMT,256.11.63.120:1062,24.30.142.256:80,TCP
FWIN,2000/05/22,13:15:16 -8:00 GMT,256.11.63.120:1063,24.30.142.256:80,TCP
FWIN,2000/05/22,13:15:30 -8:00 GMT,256.11.63.107:40286,24.30.142.256:111,TCP
FWIN,2000/05/22,13:15:30 -8:00 GMT,256.11.63.120:1058,24.30.142.256:111,TCP
FWIN,2000/05/22,13:15:30 -8:00 GMT,256.11.63.120:1059,24.30.142.256:111,TCP
FWIN,2000/05/22,13:15:46 -8:00 GMT,256.11.63.120:1060,24.30.142.256:111,TCP
FWIN,2000/05/22,13:15:46 -8:00 GMT,256.11.63.120:1061,24.30.142.256:111,TCP
FWIN,2000/05/22,13:15:46 -8:00 GMT,256.11.63.107:40287,24.30.142.256:111,TCP
FWIN,2000/05/22,13:16:02 -8:00 GMT,256.11.63.107:40288,24.30.142.256:111,TCP
FWIN,2000/05/22,13:16:02 -8:00 GMT,256.11.63.120:1062,24.30.142.256:111,TCP
FWIN,2000/05/22,13:16:02 -8:00 GMT,256.11.63.120:1063,24.30.142.256:111,TCP
FWIN,2000/05/22,13:16:30 -8:00 GMT,256.11.63.107:40286,24.30.142.256:161,TCP
FWIN,2000/05/22,13:16:30 -8:00 GMT,256.11.63.120:1058,24.30.142.256:161,TCP
FWIN,2000/05/22,13:16:30 -8:00 GMT,256.11.63.120:1059,24.30.142.256:161,TCP
FWIN,2000/05/22,13:16:46 -8:00 GMT,256.11.63.107:40287,24.30.142.256:161,TCP
FWIN,2000/05/22,13:16:46 -8:00 GMT,256.11.63.120:1060,24.30.142.256:161,TCP
FWIN,2000/05/22,13:16:46 -8:00 GMT,256.11.63.120:1061,24.30.142.256:161,TCP
FWIN,2000/05/22,13:17:00 -8:00 GMT,256.11.63.107:40288,24.30.142.256:161,TCP
FWIN,2000/05/22,13:17:00 -8:00 GMT,256.11.63.120:1062,24.30.142.256:161,TCP
FWIN,2000/05/22,13:17:00 -8:00 GMT,256.11.63.120:1063,24.30.142.256:161,TCP
FWIN,2000/05/22,13:17:30 -8:00 GMT,256.11.63.107:40286,24.30.142.256:21,TCP
FWIN,2000/05/22,13:17:30 -8:00 GMT,256.11.63.120:1058,24.30.142.256:21,TCP
FWIN,2000/05/22,13:17:30 -8:00 GMT,256.11.63.120:1059,24.30.142.256:21,TCP
FWIN,2000/05/22,13:17:46 -8:00 GMT,256.11.63.107:40287,24.30.142.256:21,TCP
FWIN,2000/05/22,13:17:46 -8:00 GMT,256.11.63.120:1060,24.30.142.256:21,TCP
FWIN,2000/05/22,13:17:46 -8:00 GMT,256.11.63.120:1061,24.30.142.256:21,TCP
FWIN,2000/05/22,13:18:02 -8:00 GMT,256.11.63.107:40288,24.30.142.256:21,TCP
FWIN,2000/05/22,13:18:02 -8:00 GMT,256.11.63.120:1062,24.30.142.256:21,TCP
FWIN,2000/05/22,13:18:02 -8:00 GMT,256.11.63.120:1063,24.30.142.256:21,TCP
FWIN,2000/05/22,13:18:30 -8:00 GMT,256.11.63.107:40286,24.30.142.256:53,TCP
FWIN,2000/05/22,13:18:30 -8:00 GMT,256.11.63.120:1058,24.30.142.256:53,TCP
FWIN,2000/05/22,13:18:30 -8:00 GMT,256.11.63.120:1059,24.30.142.256:53,TCP
FWIN,2000/05/22,13:18:46 -8:00 GMT,256.11.63.107:40287,24.30.142.256:53,TCP
FWIN,2000/05/22,13:18:46 -8:00 GMT,256.11.63.120:1060,24.30.142.256:53,TCP
FWIN,2000/05/22,13:18:46 -8:00 GMT,256.11.63.120:1061,24.30.142.256:53,TCP
FWIN,2000/05/22,13:19:00 -8:00 GMT,256.11.63.107:40288,24.30.142.256:53,TCP
FWIN,2000/05/22,13:19:00 -8:00 GMT,256.11.63.120:1062,24.30.142.256:53,TCP
FWIN,2000/05/22,13:19:00 -8:00 GMT,256.11.63.120:1063,24.30.142.256:53,TCP
FWIN,2000/05/22,13:19:30 -8:00 GMT,256.11.63.107:40289,24.30.142.256:53,TCP
FWIN,2000/05/22,13:19:30 -8:00 GMT,256.11.63.120:1064,24.30.142.256:53,TCP
FWIN,2000/05/22,13:19:30 -8:00 GMT,256.11.63.120:1065,24.30.142.256:53,TCP
FWIN,2000/05/22,13:19:46 -8:00 GMT,256.11.63.107:40290,24.30.142.256:53,TCP
FWIN,2000/05/22,13:19:46 -8:00 GMT,256.11.63.120:1066,24.30.142.256:53,TCP
FWIN,2000/05/22,13:19:46 -8:00 GMT,256.11.63.120:1067,24.30.142.256:53,TCP
FWIN,2000/05/22,13:20:02 -8:00 GMT,256.11.63.107:40291,24.30.142.256:53,TCP
FWIN,2000/05/22,13:20:02 -8:00 GMT,256.11.63.120:1068,24.30.142.256:53,TCP
FWIN,2000/05/22,13:20:02 -8:00 GMT,256.11.63.120:1069,24.30.142.256:53,TCP
FWIN,2000/05/22,13:20:16 -8:00 GMT,256.11.63.107:40289,24.30.142.256:21,TCP
FWIN,2000/05/22,13:20:16 -8:00 GMT,256.11.63.120:1064,24.30.142.256:21,TCP
```

1. **Source of trace:** A home system on a cable modem.

2. **System(s) generating detects:** Logs were generated from a ZoneAlarm personal firewall.

3. **Spoofed Probability:** Yes. There are two IP's recorded yet one appeared to be doing everything twice. After emailing the administrator assigned to the block of addresses, I found that the source IP's were assigned to a high school computer lab. The administrator eventually found that one of the systems in the lab had a script running that was port scanning a large number of known cable modem addresses and was configured to use two other "decoy" addresses. The person setting this up either forgot or did not realize that the lab's router was doing NAT for any IP address that did not have a static mapping. Therefore the "decoys" were sent out using the address assigned for NAT services (in this case the 256.11.63.120 address) and not the intended decoy addresses.

```
FWIN,5/22/2000,13:15:30 -8:00 GMT,256.11.63.107:40286,24.30.142.256:111,TCP
FWIN,5/22/2000,13:15:46 -8:00 GMT,256.11.63.107:40287,24.30.142.256:111,TCP
FWIN,5/22/2000,13:16:02 -8:00 GMT,256.11.63.107:40288,24.30.142.256:111,TCP
FWIN,5/22/2000,13:22:02 -8:00 GMT,256.11.63.107:40289,24.30.142.256:111,TCP
FWIN,5/22/2000,13:22:18 -8:00 GMT,256.11.63.107:40290,24.30.142.256:111,TCP
FWIN,5/22/2000,13:22:32 -8:00 GMT,256.11.63.107:40291,24.30.142.256:111,TCP
FWIN,5/22/2000,13:15:30 -8:00 GMT,256.11.63.120:1058,24.30.142.256:111,TCP
FWIN,5/22/2000,13:15:30 -8:00 GMT,256.11.63.120:1059,24.30.142.256:111,TCP
FWIN,5/22/2000,13:15:46 -8:00 GMT,256.11.63.120:1060,24.30.142.256:111,TCP
FWIN,5/22/2000,13:15:46 -8:00 GMT,256.11.63.120:1061,24.30.142.256:111,TCP
FWIN,5/22/2000,13:16:02 -8:00 GMT,256.11.63.120:1062,24.30.142.256:111,TCP
FWIN,5/22/2000,13:16:02 -8:00 GMT,256.11.63.120:1063,24.30.142.256:111,TCP
FWIN,5/22/2000,13:22:02 -8:00 GMT,256.11.63.120:1064,24.30.142.256:111,TCP
FWIN,5/22/2000,13:22:02 -8:00 GMT,256.11.63.120:1065,24.30.142.256:111,TCP
FWIN,5/22/2000,13:22:18 -8:00 GMT,256.11.63.120:1066,24.30.142.256:111,TCP
FWIN,5/22/2000,13:22:18 -8:00 GMT,256.11.63.120:1067,24.30.142.256:111,TCP
FWIN,5/22/2000,13:22:32 -8:00 GMT,256.11.63.120:1068,24.30.142.256:111,TCP
FWIN,5/22/2000,13:22:32 -8:00 GMT,256.11.63.120:1069,24.30.142.256:111,TCP
FWIN,5/22/2000,13:16:30 -8:00 GMT,256.11.63.107:40286,24.30.142.256:161,TCP
FWIN,5/22/2000,13:16:46 -8:00 GMT,256.11.63.107:40287,24.30.142.256:161,TCP
FWIN,5/22/2000,13:17:00 -8:00 GMT,256.11.63.107:40288,24.30.142.256:161,TCP
FWIN,5/22/2000,13:21:02 -8:00 GMT,256.11.63.107:40289,24.30.142.256:161,TCP
FWIN,5/22/2000,13:21:18 -8:00 GMT,256.11.63.107:40290,24.30.142.256:161,TCP
FWIN,5/22/2000,13:21:32 -8:00 GMT,256.11.63.107:40291,24.30.142.256:161,TCP
FWIN,5/22/2000,13:16:30 -8:00 GMT,256.11.63.120:1058,24.30.142.256:161,TCP
FWIN,5/22/2000,13:16:30 -8:00 GMT,256.11.63.120:1059,24.30.142.256:161,TCP
FWIN,5/22/2000,13:16:46 -8:00 GMT,256.11.63.120:1060,24.30.142.256:161,TCP
FWIN,5/22/2000,13:16:46 -8:00 GMT,256.11.63.120:1061,24.30.142.256:161,TCP
FWIN,5/22/2000,13:17:00 -8:00 GMT,256.11.63.120:1062,24.30.142.256:161,TCP
FWIN,5/22/2000,13:17:00 -8:00 GMT,256.11.63.120:1063,24.30.142.256:161,TCP
FWIN,5/22/2000,13:21:02 -8:00 GMT,256.11.63.120:1064,24.30.142.256:161,TCP
FWIN,5/22/2000,13:21:02 -8:00 GMT,256.11.63.120:1065,24.30.142.256:161,TCP
FWIN,5/22/2000,13:21:18 -8:00 GMT,256.11.63.120:1066,24.30.142.256:161,TCP
FWIN,5/22/2000,13:21:18 -8:00 GMT,256.11.63.120:1067,24.30.142.256:161,TCP
FWIN,5/22/2000,13:21:32 -8:00 GMT,256.11.63.120:1068,24.30.142.256:161,TCP
FWIN,5/22/2000,13:21:32 -8:00 GMT,256.11.63.120:1069,24.30.142.256:161,TCP
```

4. **Description of attack:** This was just a regular TCP connect port scan searching for some known exploitable services and for general reconnaissance purposes.

5. **Attack mechanism:** The script showed that the attack mechanism was NMAP with the decoy switch (-D…) set to use two other addresses.

6. **Correlations:** Ends up the system had not been scanning for very long since the speed was set to NMAP's slowest setting. And since these scans targeted cable modem address spaces, there were probably very few systems running anything to log this type of activity. Therefore no collaboration was available.

7. Evidence of active targeting: Slightly. These scans were specifically targeting cable modem address spaces.

8. Severity: -4

- **Criticality 2** – This system is a simple isolated workstation.
- **Lethality 2** – This system does have some of the scanned services running.
- **System Countermeasures 4** – Operating system is current with patches.
- **Network Countermeasures 4** – Firewall was current and blocks all inbound traffic except for ICMP.

9. Defense recommendation: Defenses were sufficient.

10. How many source addresses are actually at work?

a) 1
b) 2
c) 3
d) 3+

Answer: c

Detect #7

```
May 30 03:16:52 : SYN FIN Scan: 182.33.2.11:109 -> 192.168.1.98:109
May 30 03:16:52 : SYN FIN Scan: 182.33.2.11:109 -> 192.168.1.99:109
May 30 03:16:52 : SYN FIN Scan: 182.33.2.11:109 -> 192.168.1.100:109
May 30 03:16:52 : SYN FIN Scan: 182.33.2.11:109 -> 192.168.1.102:109
May 30 03:16:52 : SYN FIN Scan: 182.33.2.11:109 -> 192.168.1.103:109
May 30 03:16:52 : SYN FIN Scan: 182.33.2.11:109 -> 192.168.1.105:109


May 30 03:16:52 : SYN FIN Scan: 182.33.2.11:109 -> 192.168.1.109:109
05/30-03:16:51.598346 182.33.2.11:109 -> 192.168.1.98:109
  TCP TTL:34 TOS:0x0 ID:39426 SF**** Seq: 0x56F53897
  Ack: 0x297E3ED9 Win: 0x40400 00 00 00 00 00
05/30-03:16:51.622800 182.33.2.11:109 -> 192.168.1.99:109
  TCP TTL:34 TOS:0x0 ID:39426 SF**** Seq: 0x56F53897
  Ack: 0x297E3ED9 Win: 0x404 00 00 00 00 00
```

1. Source of trace: Arrigo Triulzi posted these scans to the GIAC we site on 5/31/00 (http://www.sans.org/y2k/053100-1200.htm). I believe the destination addresses were sanitized for posting but in actuality the real destinations were valid, routable addresses.

2. System(s) generating detects: Not sure about the top one. The lower detect appears to be a Snort output.

3. Spoofed Probability: Yes. The source IP is an un-routed address that does not appear to be assigned to anyone.

4. Description of attack: A TCP SYN/FIN packet sent to POP2. A first glance this appears to be an attempt to find POP2 servers. Which there is a known buffer overflow vulnerability (CVE-1999-0920). But what good is it if the responses cannot be received? Three (of many) possible explanations are as follows:

1. The attacker has compromised a system upstream (towards default gateway) from the probed systems and is able to sniff for responses before core routers return undeliverables.
2. If enough sources send this or similar traffic, a denial of service condition could possibly be created at major peering points (i.e. MaeEast, MaeWest).
3. Someone just screwed up entering the source address into the script.

5. Attack mechanism: I believe this scan was created using the same script as detect #3. May just be a new varient or someone modified the source to specifically target POP2. These SYN/FIN scans against known *nix vulnerabilities have been extremely popular lately, especially from the Asian regions. Some of the similarities are:

- SYN/FIN flags
- Similar speeds
- Source and destination ports are always the same
- Packet, sequence and acknowledgement ID's are identical across multiple hosts
- No payload

6. Correlations: Appears to be similar to the very popular SYN/FIN scans against ports 53 (DNS) and 111 (SunRPC).

7. Evidence of active targeting: Probably not since similar types of scans have been seen across entire class "B" subnets.

8. Severity: -1 (if this was targeted against my network)

- **Criticality 4** – This scan targeted mail servers running older services.
- **Lethality 3** – If the scanner were to find vulnerable POP2 servers, certain privileges could be gained.
- **System Countermeasures 4** – Operating systems are current with no POP2 services running.
- **Network Countermeasures 4** – Routers and firewalls block all traffic to TCP 109.

9. Defense recommendation: Defenses are sufficient. Log all future traffic from this un-routed network.

10. What does the above detect probably show?

a) POP2 scan
b) Network reconnaissance attempt
c) DDOS against 182.33.2.1
d) POP2 buffer overflow attempt

Answer: a

Detect #8

```
[**] IDS152/Ping BSDtype [**]
05/30-20:52:52.664284 209.19.115.147 -> 256.46.217.0
ICMP TTL:54 TOS:0x0 ID:43636
ID:33148   Seq:0  ECHO
2F 84 34 39 E3 FC 07 00 08 09 0A 0B 0C 0D 0E 0F  /.49............
10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F  ................
20 21 22 23 24 25 26 27 28 29 2A 2B 2C 2D 2E 2F   !"#$%&'()*+,-./
30 31 32 33 34 35 36 37                          01234567

[**] IDS152/Ping BSDtype [**]
05/30-20:52:53.683005 209.19.115.147 -> 256.46.217.0
ICMP TTL:54 TOS:0x0 ID:43698
ID:33148   Seq:256  ECHO
30 84 34 39 5D EF 07 00 08 09 0A 0B 0C 0D 0E 0F  0.49]...........
10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F  ................
20 21 22 23 24 25 26 27 28 29 2A 2B 2C 2D 2E 2F   !"#$%&'()*+,-./
30 31 32 33 34 35 36 37                          01234567

[**] IDS163/Ping OpenBSD-Linux [**]
06/07-15:13:35.413670 151.27.128.203 -> 256.46.217.0
ICMP TTL:104 TOS:0xE0 ID:43051
ID:16643   Seq:256  ECHO
4C C8 3E 39 F3 0B 08 00 26 D0 2E D9 00 0D 0E 0F  L.>9....&.......
10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F  ................
20 21 22 23 24 25 26 27 28 29 2A 2B 2C 2D 2E 2F   !"#$%&'()*+,-./
30 31 32 33 34 35 36 37                          01234567

[**] IDS171/Ping zeros [**]
06/09-11:15:53.332810 62.82.211.237 -> 256.46.217.0
ICMP TTL:230 TOS:0x0 ID:450
ID:11776   Seq:55552  ECHO
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00                             .......
```

1. Source of trace: These detections were observed while monitoring a group of our servers at a co-location facility.

2. System(s) generating detects: These alerts were recorded using Snort 1.6 in conjunction with what was a current copy of the arachNIDS signature file.

3. Spoofed Probability: Probably not. The three scanning systems shown above seem to be working independently.

4. Description of attack: All of the above detects show ICMP echo requests being sent to a standard broadcast address. Any systems that respond to broadcasts echo requests would then reply back with ICMP echo replies. Also due to the standard payloads some operating systems use, it appears the scanning systems appear to be running different variants of *nix operating systems. Solaris has been observed sending “Ping zero” type echo requests as shown in detect #1.

5. Attack mechanism: Could have been just about anything. Probably just a script written to ping common class “C” broadcast addresses and record results from responding systems.

6. Correlations: Most routers, switches, printers and Unix systems respond to these requests. Windows and other Microsoft systems do not. This is addition to the fact that it appears the systems sending the pings appear to be *nix based systems, the scanners are probably probing for other *nix based systems.

7. Evidence of active targeting: Not directly. We do not have a standard class "C" subnet and no other non-standard broadcast addresses appear to have been probed.

8. Severity: -3

- **Criticality 2** – This was a non-discriminate scan.
- **Lethality 2** – This scan was used for recon purposes.
- **System Countermeasures 3** – Most systems do not respond to this type of echo request. Routers, switches and some other hardware would.
- **Network Countermeasures 4** – Firewalls and routers block all inbound traffic to network and broadcast addresses. Some limited ICMP traffic is allowed to pass.

9. Defense recommendation: Defenses were sufficient.

10. What does the above detect probably show?

a) ICMP Subnet mask request
b) ICMP reconnaissance attempt
c) LOKI traffic
d) Internal system is compromised

Answer: b

Detect #9

```
[**] FrontPage Service PWD Scan [**]
06/05-04:09:05.995695 204.210.6.163:22182 -> 256.103.247.256:80
TCP TTL:48 TOS:0x0 ID:27045  DF
*****PA* Seq: 0xCE40528A   Ack: 0xB6B94EC5   Win: 0x1C84
47 45 54 20 2F 5F 76 74 69 5F 70 76 74 2F 73 65  GET /_vti_pvt/se
72 76 69 63 65 2E 70 77 64 20 48 54 54 50 2F 31  rvice.pwd HTTP/1
2E 30 0D 0A 48 6F 73 74 3A 20 32 31 36 2E 31 30  .0..Host: 216.10
33 2E 32 34 37 2E 31 39 36 0D 0A 0D 0A           3.247.196....


[**] FrontPage User PWD Scan [**]
06/05-04:09:07.564710 204.210.6.163:22184 -> 256.103.247.256:80
TCP TTL:48 TOS:0x0 ID:27092  DF
*****PA* Seq: 0x168DEFBF   Ack: 0x10EEAD23   Win: 0x1C84
47 45 54 20 2F 5F 76 74 69 5F 70 76 74 2F 75 73  GET /_vti_pvt/us
65 72 73 2E 70 77 64 20 48 54 54 50 2F 31 2E 30  ers.pwd HTTP/1.0
0D 0A 48 6F 73 74 3A 20 32 31 36 2E 31 30 33 2E  ..Host: 216.103.
32 34 37 2E 31 39 36 0D 0A 0D 0A                 247.196....

[**] FrontPage Admin PWD Scan [**]
06/05-04:09:08.429066 204.210.6.163:22185 -> 256.103.247.256:80
TCP TTL:48 TOS:0x0 ID:27108  DF
*****PA* Seq: 0xA8D45946   Ack: 0xE1752ACF   Win: 0x1C84
47 45 54 20 2F 5F 76 74 69 5F 70 76 74 2F 61 64  GET /_vti_pvt/ad
6D 69 6E 69 73 74 72 61 74 6F 72 73 2E 70 77 64  ministrators.pwd
20 48 54 54 50 2F 31 2E 30 0D 0A 48 6F 73 74 3A   HTTP/1.0..Host:
20 32 31 36 2E 31 30 33 2E 32 34 37 2E 31 39 36   216.103.247.196
0D 0A 0D 0A                                      ....

[**] FrontPage Author PWD Scan [**]
06/05-04:09:09.038080 204.210.6.163:22186 -> 256.103.247.256:80
TCP TTL:48 TOS:0x0 ID:27125  DF
*****PA* Seq: 0x6B3221F7   Ack: 0x9C225B0B   Win: 0x1C84
47 45 54 20 2F 5F 76 74 69 5F 70 76 74 2F 61 75  GET /_vti_pvt/au
74 68 6F 72 73 2E 70 77 64 20 48 54 54 50 2F 31  thors.pwd HTTP/1
2E 30 0D 0A 48 6F 73 74 3A 20 32 31 36 2E 31 30  .0..Host: 216.10
33 2E 32 34 37 2E 31 39 36 0D 0A 0D 0A           3.247.196....
```

1. Source of trace: These detections were observed while monitoring a small web hosting environment.

2. System(s) generating detects: These alerts were recorded using Snort 1.6 in conjunction with what was a current copy of the arachNIDS signature file.

3. Spoofed Probability: Probably not. The source IP address was also found performing some normal web browsing.

4. Description of attack: The above detects show attempts to retrieve some default password files used with FrontPage enabled web servers. FrontPage has an enormous amount of security flaws. This attack is just an attempt to exploit one of the most basic. FrontPage stores the passwords used for HTTP uploads in the "/vti_pvt/" directory. If the passwords are stored with the default names (as they usually are) and the GET request is successful, the password can then be decrypted offline. FrontPage stores these password in a Unix format so even though they have the standard *.pwd extension, a normal "pwd" decoder will not work. The attacker would then just need to use a Unix password cracker similar to Cracker Jack or Jack the Ripper.

5. Attack mechanism: These attacks appear to be a launched by a script due to the speed at which the GET requests were sent. There are a number of automated scanners out there that test for these vulnerabilities so pin pointing the exact attack mechanism would be very difficult.

6. Correlations: Other than some normal web browsing, no other traffic was seen from this source.

7. Evidence of active targeting: Yes. Our web-hosting environment houses both IIS and Apache servers. This attack was only seen on our most popular IIS server.

8. Severity: 0

- **Criticality 4** – This attack was focused against a corporate web server.
- **Lethality 2** – No FrontPage is used. However an illegal server could have exposed a serious security flaw.
- **System Countermeasures 5** – FrontPage extensions are not used on any servers and directories are completely removed. (Maybe I should put one there and let them bang on it a while! ☺)
- **Network Countermeasures 1** – Firewalls proxy all web traffic but would have allowed this traffic to pass.

9. Defense recommendation: Defenses were sufficient. Should enable URL logging on firewalls and log all future traffic from this source.

10. What does the above detect probably show?

a) HTTP Buffer overflows
b) Normal HTTP traffic
c) Netcat traffic
d) Password retrieval attempts

Answer: d

Detect #10

```
[**] Ripper Pro [**]
05/28-16:25:05.790030 129.119.63.12:123 -> 256.46.217.256:2023
UDP TTL:237 TOS:0x0 ID:59441  DF
Len: 56
14 03 0B F0 00 00 09 1C 00 00 08 08 81 77 03 02  .............w..
BC DC 29 1A 8F 4B D0 00 BC DC 29 2B 42 0C 49 BA  ..)..K....)+B.I.
BC DC 29 3B E5 06 F0 00 BC DC 29 3B E5 0C E0 00  ..);......);....

[**] Ripper Pro [**]
05/28-16:25:10.781353 129.119.63.12:123 -> 256.46.217.256:2023
UDP TTL:237 TOS:0x0 ID:59442  DF
Len: 56
14 03 0B F0 00 00 09 1C 00 00 08 0C 81 77 03 02  .............w..
BC DC 29 1A 8F 4B D0 00 BC DC 29 30 42 0C 49 BA  ..)..K....)0B.I.
BC DC 29 40 E4 F9 E0 00 BC DC 29 40 E5 02 F0 00  ..)@......)@....

[**] Ripper Pro [**]
05/28-16:25:15.779743 129.119.63.12:123 -> 256.46.217.256:2023
UDP TTL:237 TOS:0x0 ID:59443  DF
Len: 56
14 03 0B F0 00 00 09 1C 00 00 08 10 81 77 03 02  .............w..
BC DC 29 1A 8F 4B D0 00 BC DC 29 35 42 0C 49 BA  ..)..K....)5B.I.
BC DC 29 45 E4 92 90 00 BC DC 29 45 E4 98 80 00  ..)E......)E....

[**] Striker [**]
06/04-15:51:13.618152 129.119.63.12:123 -> 256.46.217.256:2565
UDP TTL:236 TOS:0x0 ID:32667  DF
Len: 56
14 03 0B F0 00 00 03 78 00 00 0F 2D 81 77 03 02  .......x...-.w..
BC E5 59 B4 8F 10 B0 00 BC E5 5B 9F A1 06 24 DD  ..Y.......[...$.
BC E5 5B AF EA F9 F0 00 BC E5 5B AF EA FF E0 00  ..[.......[.....

[**] Striker [**]
06/04-15:51:18.607863 129.119.63.12:123 -> 256.46.217.256:2565
UDP TTL:236 TOS:0x0 ID:32668  DF
Len: 56
14 03 0B F0 00 00 03 78 00 00 0F 31 81 77 03 02  .......x...1.w..
BC E5 59 B4 8F 10 B0 00 BC E5 5B A4 A1 06 24 DD  ..Y.......[...$.
BC E5 5B B4 EA 59 B0 00 BC E5 5B B4 EA 5F 10 00  ..[..Y....[.._..

[**] Striker [**]
06/04-15:51:23.609124 129.119.63.12:123 -> 256.46.217.256:2565
UDP TTL:236 TOS:0x0 ID:32669  DF
Len: 56
14 03 0B F0 00 00 03 78 00 00 0F 34 81 77 03 02  .......x...4.w..
BC E5 59 B4 8F 10 B0 00 BC E5 5B A9 A1 06 24 DD  ..Y.......[...$.
BC E5 5B B9 EA 7C 30 00 BC E5 5B B9 EA 81 90 00  ..[..|0...[.....

[**] Portal Of Doom [**]
05/30-01:35:45.665582 129.119.63.12:123 -> 256.46.217.256:3700
UDP TTL:237 TOS:0x0 ID:3117  DF
Len: 56
14 03 0B F0 00 00 09 3D 00 00 11 83 81 77 03 02  .......=.....w..
BC DD F7 E5 8F 8E B0 00 BC DD FB D0 E1 89 37 4B  ..............7K
BC DD FB E1 76 9A 90 00 BC DD FB E1 76 A4 D0 00  ....v.......v...

[**] Portal Of Doom [**]
05/30-01:35:50.655339 129.119.63.12:123 -> 256.46.217.256:3700
UDP TTL:237 TOS:0x0 ID:3118  DF
Len: 56
14 03 0B F0 00 00 09 1B 00 00 0E 42 81 77 03 02  ...........B.w..
BC DD FB E5 8E D6 30 00 BC DD FB D5 E1 89 37 4B  ......0.......7K
BC DD FB E6 76 2F F0 00 BC DD FB E6 76 35 50 00  ....v/......v5P.
```

```
[**] Portal Of Doom [**]
05/30-01:35:55.655883 129.119.63.12:123 -> 256.46.217.256:3700
UDP TTL:237 TOS:0x0 ID:3119  DF
Len: 56
14 03 0B F0 00 00 09 1B 00 00 0E 45 81 77 03 02  ...........E.w..
BC DD FB E5 8E D6 30 00 BC DD FB DA E1 89 37 4B  ......0.......7K
BC DD FB EB 76 30 E0 00 BC DD FB EB 76 37 B0 00  ....v0......v7..

[**] Sockets De Troie [**]
06/06-13:06:31.617074 129.119.63.12:123 -> 256.46.217.256:5000
UDP TTL:236 TOS:0x0 ID:38231  DF
Len: 56
14 03 0B F0 00 00 09 2F 00 00 4B 30 81 77 03 02  ......./..K0.w..
BC E7 D5 5C 8E 8F 00 00 BC E7 D7 F5 02 D0 E5 60  ...\...........`
BC E7 D8 05 33 41 70 00 BC E7 D8 05 33 4D A0 00  ....3Ap.....3M..

[**] Sockets De Troie [**]
06/06-13:06:36.612217 129.119.63.12:123 -> 256.46.217.256:5000
UDP TTL:236 TOS:0x0 ID:38232  DF
Len: 56
14 03 0B F0 00 00 09 2F 00 00 4B 33 81 77 03 02  ......./..K3.w..
BC E7 D5 5C 8E 8F 00 00 BC E7 D7 FA 02 D0 E5 60  ...\...........`
BC E7 D8 0A 32 9A F0 00 BC E7 D8 0A 32 A3 C0 00  ....2.......2...

[**] Sockets De Troie [**]
06/06-13:06:41.632875 129.119.63.12:123 -> 256.46.217.256:5000
UDP TTL:236 TOS:0x0 ID:38233  DF
Len: 56
14 03 0B F0 00 00 09 2F 00 00 4B 37 81 77 03 02  ......./..K7.w..
BC E7 D5 5C 8E 8F 00 00 BC E7 D7 FF 02 D0 E5 60  ...\...........`
BC E7 D8 0F 33 26 C0 00 BC E7 D8 0F 33 2C D0 00  ....3&......3,..
```

1. Source of trace: These detections were observed while monitoring a group of our servers at a co-location facility.

2. System(s) generating detects: These alerts were recorded using Snort 1.6 in conjunction with what was a current copy of the arachNIDS signature file.

3. Spoofed Probability: Possibly but not very likely. These alerts were seen over a period of about a week. The source IP resolved to **prod02.fits.smu.edu**, which did not ring any bells as to a system we normally "talk" to.

4. Description of attack: The attacks appeared to be a "slow and low" UDP scan of some known trojan ports against a system which was believed to be completely unreachable from the internet. Our firewalls also did not log these packets as being dropped either, which meant the traffic probably originated internally. It could have also been that someone might be attempting to bypass our firewalls using the NTP port as the source. The only NTP traffic should have been to/from our core router which acted as our timeserver.

After picking apart the destination system, which ended up being a Windows 2000 server, it was found that Microsoft's Active Directory service requires a connection to an NTP server and during the configuration, the engineer building the server, entered **time.smu.edu** as the time server. **time.smu.edu** ends up being an alias for **prod02.fits.smu.edu**. Hence this was illegal, but normal NTP traffic.

5. Attack mechanism: Eventually found to be regular NTP traffic. When the NTP server responded to the internal system's randomly chosen port, Snort alerted when this port corresponded to a known Trojan port.

6. Correlations: Initially appeared to be NTP traffic but all internal systems were thought to have been configured to speak only with our core router.

7. Evidence of active targeting: Yes. This traffic was not seen to any other system and was fairly constant.

8. Severity: -1 (What it initially appeared)

- **Criticality 2** – This attack was focused against a test server with no trusts or rights.
- **Lethality 3** – If a Trojan was found, administrative access to this system would be compromised.
- **System Countermeasures 4** – Operating system was current with patches.
- **Network Countermeasures 0** – Appeared to penetrate our firewalls and routers.

9. Defense recommendation: Implement egress filters to block illegal NTP as well as other illegal outbound traffic.

10. What does the above detect probably show?

a) UDP Trojan scan
b) Normal NTP traffic
c) NTP buffer overflow
d) Denial of service

Answer: b