Global Information Assurance Certification Paper

Detect #1

```
04/05/2000 19:50:09.278559 ep0 @0:1 b 203.236.214.72,3172 -> 192.168.4.2,23 P
R tcp len 20 60 -S IN
04/05/2000 19:50:09.295784 ep0 @0:1 b 203.236.214.72,3171 -> 192.168.4.1,23 P
R tcp len 20 60 -S IN
04/05/2000 19:50:09.317147 ep0 @0:1 b 203.236.214.72,3173 -> 192.168.4.3,23 P
R tcp len 20 60 -S IN
04/05/2000 19:50:09.332854 ep0 @0:1 b 203.236.214.72,3174 -> 192.168.4.4,23 P
R tcp len 20 60 -S IN
04/05/2000 19:50:12.212825 ep0 @0:1 b 203.236.214.72,3172 -> 192.168.4.2,23 P
R tcp len 20 60 -S IN
04/05/2000 19:50:12.218241 ep0 @0:1 b 203.236.214.72,3173 -> 192.168.4.3,23 P
R tcp len 20 60 -S IN
04/05/2000 19:50:12.231769 ep0 @0:1 b 203.236.214.72,3171 -> 192.168.4.1,23 P
R tcp len 20 60 -S IN
04/05/2000 19:50:12.241254 ep0 @0:1 b 203.236.214.72,3174 -> 192.168.4.4,23 P
R tcp len 20 60 -S IN
```

This trace comes from the home network.

The detect was generated by ipmon, part of ip-filter. (see appendix below about ip-filter format)

In this scan the attacker would need the tcp handshake to occur; therefore I propose that the source IP address is not spoofed. Unfortunately, there is not enough information to tell definitively if the packet was crafted, IP header id's and sequence numbers would be useful information.

This individual is attempting to connect to tcp port 23 either to check for trojans like Tiny Telnet server, Truva Atl, or to see if there are any telnet daemons running that will give the attacker an Operating System banner so that they could launch attacks directed at that Operating System. There are also buffer overrun problems in some versions of the telnet daemon. The defenses are sufficient in this case as the firewall blocked this attack.

some relevant URL's:

http://www.cert.org/advisories/CA-89.03.telnet.breakin.warning.html
http://www.cert.org/advisories/CA-95.14.Telnetd_Environment_Vulnerability.html

Severity:

criticality:	directed at desktops and servers:	3
lethality:	trojan / root access:	5
system countermeasures:	Modern OS:	4
network countermeasures:	Restrictive Firewall:	4
		-
		0

Multiple choice question:

Which of the following is true about the above scan?

A) This packet was crafted as the 203.236.214.0 network is specified as a private internet network in RFC 1918

B) These packets were blocked by ip-filter
C) These connection attempts were run by a user as opposed to being run from a script or program

ANSWER: B

Detect #2

```
14/05/2000 15:25:18.528182 ep0 @0:1 b 208.36.197.100,1359 -> 192.168.4.1,22 P
R udp len 20 30  IN
14/05/2000 15:26:08.587135 ep0 @0:1 b 208.36.197.100,1359 -> 192.168.4.1,5632
 PR udp len 20 30  IN
14/05/2000 15:26:08.591544 ep0 @0:1 b 208.36.197.100,1359 -> 192.168.4.1,22 P
R udp len 20 30  IN
14/05/2000 15:26:08.597087 ep0 @0:1 b 208.36.197.100,1359 -> 192.168.4.2,5632
 PR udp len 20 30  IN
14/05/2000 15:26:08.602890 ep0 @0:1 b 208.36.197.100,1359 -> 192.168.4.2,22 P
R udp len 20 30  IN
14/05/2000 15:26:58.667722 ep0 @0:1 b 208.36.197.100,1359 -> 192.168.4.1,5632
 PR udp len 20 30  IN
14/05/2000 15:26:58.671927 ep0 @0:1 b 208.36.197.100,1359 -> 192.168.4.1,22 P
R udp len 20 30  IN
14/05/2000 15:26:58.674097 ep0 @0:1 b 208.36.197.100,1359 -> 192.168.4.2,5632
 PR udp len 20 30  IN
14/05/2000 15:26:58.677604 ep0 @0:1 b 208.36.197.100,1359 -> 192.168.4.2,22 P
R udp len 20 30  IN

12:40:23.512143 208.36.197.100.1359 > 192.168.4.1.5632:  udp 2
12:40:23.516664 208.36.197.100.1359 > 192.168.4.1.22:  udp 2
12:40:23.518935 208.36.197.100.1359 > 192.168.4.2.5632:  udp 2
12:40:23.523447 208.36.197.100.1359 > 192.168.4.2.22:  udp 2
12:40:23.527914 208.36.197.100.1359 > 192.168.4.3.5632:  udp 2
12:40:23.532475 208.36.197.100.1359 > 192.168.4.3.22:  udp 2
12:40:23.552737 208.36.197.100.1359 > 192.168.4.4.22:  udp 2
12:41:13.573092 208.36.197.100.1359 > 192.168.4.1.5632:  udp 2
12:41:13.575438 208.36.197.100.1359 > 192.168.4.1.22:  udp 2
12:41:13.578709 208.36.197.100.1359 > 192.168.4.2.5632:  udp 2
12:41:13.583813 208.36.197.100.1359 > 192.168.4.2.22:  udp 2
12:41:13.585516 208.36.197.100.1359 > 192.168.4.3.5632:  udp 2
12:41:13.587706 208.36.197.100.1359 > 192.168.4.3.22:  udp 2
12:41:13.592329 208.36.197.100.1359 > 192.168.4.4.5632:  udp 2
12:41:13.597846 208.36.197.100.1359 > 192.168.4.4.22:  udp 2
12:42:03.568869 208.36.197.100.1359 > 192.168.4.1.5632:  udp 2
12:42:03.573340 208.36.197.100.1359 > 192.168.4.1.22:  udp 2
12:42:03.577835 208.36.197.100.1359 > 192.168.4.2.5632:  udp 2
12:42:03.583449 208.36.197.100.1359 > 192.168.4.2.22:  udp 2
12:42:03.588268 208.36.197.100.1359 > 192.168.4.3.5632:  udp 2
12:42:03.593574 208.36.197.100.1359 > 192.168.4.3.22:  udp 2
12:42:03.597972 208.36.197.100.1359 > 192.168.4.4.5632:  udp 2
12:42:03.602628 208.36.197.100.1359 > 192.168.4.4.22:  udp 2
12:42:53.657914 208.36.197.100.1359 > 192.168.4.1.5632:  udp 2
12:42:53.663543 208.36.197.100.1359 > 192.168.4.1.22:  udp 2
12:42:53.669257 208.36.197.100.1359 > 192.168.4.2.5632:  udp 2
12:42:53.673607 208.36.197.100.1359 > 192.168.4.2.22:  udp 2
12:42:53.678266 208.36.197.100.1359 > 192.168.4.3.5632:  udp 2
12:42:53.683718 208.36.197.100.1359 > 192.168.4.3.22:  udp 2
```

```
12:42:53.688355 208.36.197.100.1359 > 192.168.4.4.5632:  udp 2
12:42:53.693818 208.36.197.100.1359 > 192.168.4.4.22:  udp 2
... continued in 50 second intervals ...
13:12:56.479514 208.36.197.100.1359 > 192.168.4.1.5632:  udp 2
13:12:56.482030 208.36.197.100.1359 > 192.168.4.1.22:  udp 2
13:12:56.487673 208.36.197.100.1359 > 192.168.4.2.5632:  udp 2
13:12:56.493408 208.36.197.100.1359 > 192.168.4.2.22:  udp 2
13:12:56.498923 208.36.197.100.1359 > 192.168.4.3.5632:  udp 2
13:12:56.501129 208.36.197.100.1359 > 192.168.4.3.22:  udp 2
13:12:56.503476 208.36.197.100.1359 > 192.168.4.4.5632:  udp 2
13:12:56.507912 208.36.197.100.1359 > 192.168.4.4.22:  udp 2


23:07:04.602858 208.36.197.100.1359 > 192.168.4.1.5632:  udp 2 (ttl 127, id
5947)
23:07:04.606963 208.36.197.100.1359 > 192.168.4.1.22:  udp 2 (ttl 127, id 6203)
23:07:04.608524 208.36.197.100.1359 > 192.168.4.2.5632:  udp 2 (ttl 127, id
6459)
23:07:04.610667 208.36.197.100.1359 > 192.168.4.2.22:  udp 2 (ttl 127, id 6715)
23:07:04.612873 208.36.197.100.1359 > 192.168.4.3.5632:  udp 2 (ttl 127, id
6971)
23:07:04.615088 208.36.197.100.1359 > 192.168.4.3.22:  udp 2 (ttl 127, id 7227)
23:07:04.617512 208.36.197.100.1359 > 192.168.4.4.5632:  udp 2 (ttl 127, id
7483)
23:07:04.619699 208.36.197.100.1359 > 192.168.4.4.22:  udp 2 (ttl 127, id 7739)
23:07:54.670662 208.36.197.100.1359 > 192.168.4.1.5632:  udp 2 (ttl 127, id
8509)
23:07:54.672877 208.36.197.100.1359 > 192.168.4.1.22:  udp 2 (ttl 127, id 8765)
23:07:54.675080 208.36.197.100.1359 > 192.168.4.2.5632:  udp 2 (ttl 127, id
9021)
23:07:54.677207 208.36.197.100.1359 > 192.168.4.2.22:  udp 2 (ttl 127, id 9277)
23:07:54.679591 208.36.197.100.1359 > 192.168.4.3.5632:  udp 2 (ttl 127, id
9533)
23:07:54.681841 208.36.197.100.1359 > 192.168.4.3.22:  udp 2 (ttl 127, id 9789)
23:07:54.684062 208.36.197.100.1359 > 192.168.4.4.5632:  udp 2 (ttl 127, id
10045)
23:07:54.686237 208.36.197.100.1359 > 192.168.4.4.22:  udp 2 (ttl 127, id 10301)
... continued in 50 second intervals ...
23:21:15.686481 208.36.197.100.1359 > 192.168.4.1.5632:  udp 2 (ttl 127, id
43869)
23:21:15.688571 208.36.197.100.1359 > 192.168.4.1.22:  udp 2 (ttl 127, id 44125)
23:21:15.692032 208.36.197.100.1359 > 192.168.4.2.5632:  udp 2 (ttl 127, id
44381)
23:21:15.694322 208.36.197.100.1359 > 192.168.4.2.22:  udp 2 (ttl 127, id 44637)
23:21:15.696470 208.36.197.100.1359 > 192.168.4.3.5632:  udp 2 (ttl 127, id
44893)
23:21:15.698657 208.36.197.100.1359 > 192.168.4.3.22:  udp 2 (ttl 127, id 45149)
23:21:15.701036 208.36.197.100.1359 > 192.168.4.4.5632:  udp 2 (ttl 127, id
45405)
23:21:15.703308 208.36.197.100.1359 > 192.168.4.4.22:  udp 2 (ttl 127, id 45661)
```

This is another trace from the home network

The first section is generated from ipmon (ip-filter), the
second and third are tcpdump output.

There is no conclusive evidence that the source IP is spoofed. There

appears to be no evidence of packet crafting either (IP header id's look resonable)

This is the signature of a scan for PC Anywhere. PC Anywhere is the Symantec product that allows remote administration of machines running a Microsoft Windows Operating System which can be a problem if it is running and the wrong people gain control.

Our network had been seeing these for a while; in the beginning it would be a quick scan on udp 22 and udp 5632 on each host and then it would end. After some time however this became 20-40 minutes of 50 second interval packet groups. Needless to say this was rather annoying so we contacted their ISP and got them to stop. The defenses are sufficient as the firewall blocked this attack.

This attack has been seen several times on this network; I found other mention of it on the sans website:
http://www.sans.org/y2k/123199-1220.htm

Severity:

criticality:	directed at windows desktops:	2
lethality:	root access:	5
system countermeasures:	Modern OS:	4
network countermeasures:	Restrictive Firewall:	4
		-
		-1

Multiple choice question:

The signature of the above detect is:

A) a subseven trojan scan
B) an ssh scan
C) a udp high port scan
D) a combination of B and C
E) None of the above

ANSWER: E (PC Anywhere scan)

Detect #3

```
04:06:50.168248 206.176.81.2.4585 > 192.168.4.2.110: S 4276539445:4276539445(0)
win 32120 <mss 1460,sackOK,timestamp 38944489 0,nop,wscale 0> (DF) (ttl 52, id
15997)
04:06:50.170375 206.176.81.2.4582 > 192.168.4.1.110: S 921492165:921492165(0)
win 32120 <mss 1460,sackOK,timestamp 38944489 0,nop,wscale 0> (DF) (ttl 52, id
15996)
04:06:53.164412 206.176.81.2.4585 > 192.168.4.2.110: S 4276539445:4276539445(0)
win 32120 <mss 1460,sackOK,timestamp 38944789 0,nop,wscale 0> (DF) (ttl 52, id
20604)
04:06:53.166247 206.176.81.2.4582 > 192.168.4.1.110: S 921492165:921492165(0)
win 32120 <mss 1460,sackOK,timestamp 38944789 0,nop,wscale 0> (DF) (ttl 52, id
20603)
```

This trace comes from the home network. It was generated using tcpdump.

In this probe the attacker would want the tcp handshake to occur so I propose that the source IP is not spoofed. I do not see any evidence of packet crafting in this detect.

This individual is attempting to connect to tcp port 110. They are probably attempting to connect to a pop3 daemon and attempt a buffer overrun. Another possibility is that they are searching for the promail trojan; however, since they only targeted well known mail servers running unix, I would make an educated guess that they are targeting a pop3 daemon. This attack is a targeted attack directed at our mail servers. Fortunately the firewall blocked this probe.

URL's related to imap/pop buffer overruns:
http://www.cert.org/advisories/CA-98.09.imapd.html
http://www.cert.org/advisories/CA-97.09.imap_pop.html

Severity:

criticality:	directed at mail relay and mail hub:	4
lethality:	root access:	5
system countermeasures:	Modern OS:	4
network countermeasures:	Restrictive Firewall:	4
		-
		1

Multiple choice question:

The tcp flags set in the above detect:

A) the Fin flag
B) the Syn flag and the Ack flag
C) the Syn flag
D) None of the above

ANSWER: C

Detect #4

```
14:42:26.291888 192.168.1.1.137 > 192.168.4.1.137:
>>> NBT UDP PACKET(137): QUERY; REQUEST; BROADCAST
TrnID=0x2F2E
OpCode=0
NmFlags=0x1
Rcode=0
QueryCount=1
AnswerCount=0
AuthorityCount=0
AddressRecCount=0
QuestionRecords:
Name=*               NameType=0x00 (Workstation)
QuestionType=0x21
QuestionClass=0x1
```

```
 (ttl 115, id 60220)
14:42:26.298132 169.254.186.43.137 > 192.168.4.1.137:
>>> NBT UDP PACKET(137): QUERY; REQUEST; BROADCAST
TrnID=0x2F30
OpCode=0
NmFlags=0x1
Rcode=0
QueryCount=1
AnswerCount=0
AuthorityCount=0
AddressRecCount=0
QuestionRecords:
Name=*               NameType=0x00 (Workstation)
QuestionType=0x21
QuestionClass=0x1

 (ttl 115, id 60476)
14:42:26.304927 64.228.81.68.137 > 192.168.4.1.137:
>>> NBT UDP PACKET(137): QUERY; REQUEST; BROADCAST
TrnID=0x2F32
OpCode=0
NmFlags=0x1
Rcode=0
QueryCount=1
AnswerCount=0
AuthorityCount=0
AddressRecCount=0
QuestionRecords:
Name=*               NameType=0x00 (Workstation)
QuestionType=0x21
QuestionClass=0x1

 (ttl 115, id 60732)
14:42:27.784316 192.168.1.1.137 > 192.168.4.1.137:
>>> NBT UDP PACKET(137): QUERY; REQUEST; BROADCAST
TrnID=0x2F34
OpCode=0
NmFlags=0x1
Rcode=0
QueryCount=1
AnswerCount=0
AuthorityCount=0
AddressRecCount=0
QuestionRecords:
Name=*               NameType=0x00 (Workstation)
QuestionType=0x21
QuestionClass=0x1

 (ttl 115, id 60988)
14:42:27.794369 64.228.81.68.137 > 192.168.4.1.137:
>>> NBT UDP PACKET(137): QUERY; REQUEST; BROADCAST
TrnID=0x2F36
OpCode=0
NmFlags=0x1
Rcode=0
QueryCount=1
```

```
AnswerCount=0
AuthorityCount=0
AddressRecCount=0
QuestionRecords:
Name=*               NameType=0x00 (Workstation)
QuestionType=0x21
QuestionClass=0x1

 (ttl 115, id 61244)
14:42:27.801119 169.254.186.43.137 > 192.168.4.1.137:
>>> NBT UDP PACKET(137): QUERY; REQUEST; BROADCAST
TrnID=0x2F38
OpCode=0
NmFlags=0x1
Rcode=0
QueryCount=1
AnswerCount=0
AuthorityCount=0
AddressRecCount=0
QuestionRecords:
Name=*               NameType=0x00 (Workstation)
QuestionType=0x21
QuestionClass=0x1

 (ttl 115, id 61500)
14:42:29.287296 192.168.1.1.137 > 192.168.4.1.137:
>>> NBT UDP PACKET(137): QUERY; REQUEST; BROADCAST
TrnID=0x2F3A
OpCode=0
NmFlags=0x1
Rcode=0
QueryCount=1
AnswerCount=0
AuthorityCount=0
AddressRecCount=0
QuestionRecords:
Name=*               NameType=0x00 (Workstation)
QuestionType=0x21
QuestionClass=0x1

 (ttl 115, id 61756)
14:42:29.293961 169.254.186.43.137 > 192.168.4.1.137:
>>> NBT UDP PACKET(137): QUERY; REQUEST; BROADCAST
TrnID=0x2F3C
OpCode=0
NmFlags=0x1
Rcode=0
QueryCount=1
AnswerCount=0
AuthorityCount=0
AddressRecCount=0
QuestionRecords:
Name=*               NameType=0x00 (Workstation)
QuestionType=0x21
QuestionClass=0x1

 (ttl 115, id 62012)
```

```
14:42:29.301881 64.228.81.68.137 > 192.168.4.1.137:
>>> NBT UDP PACKET(137): QUERY; REQUEST; BROADCAST
TrnID=0x2F3E
OpCode=0
NmFlags=0x1
Rcode=0
QueryCount=1
AnswerCount=0
AuthorityCount=0
AddressRecCount=0
QuestionRecords:
Name=*               NameType=0x00 (Workstation)
QuestionType=0x21
QuestionClass=0x1

 (ttl 115, id 62268)
14:42:36.817906 192.168.1.1.137 > 192.168.4.2.137:
>>> NBT UDP PACKET(137): QUERY; REQUEST; BROADCAST
TrnID=0x2F48
OpCode=0
NmFlags=0x1
Rcode=0
QueryCount=1
AnswerCount=0
AuthorityCount=0
AddressRecCount=0
QuestionRecords:
Name=*               NameType=0x00 (Workstation)
QuestionType=0x21
QuestionClass=0x1

 (ttl 115, id 63804)
14:42:36.821214 169.254.186.43.137 > 192.168.4.2.137:
>>> NBT UDP PACKET(137): QUERY; REQUEST; BROADCAST
TrnID=0x2F4A
OpCode=0
NmFlags=0x1
Rcode=0
QueryCount=1
AnswerCount=0
AuthorityCount=0
AddressRecCount=0
QuestionRecords:
Name=*               NameType=0x00 (Workstation)
QuestionType=0x21
QuestionClass=0x1

 (ttl 115, id 64060)
14:42:36.826828 64.228.81.68.137 > 192.168.4.2.137:
>>> NBT UDP PACKET(137): QUERY; REQUEST; BROADCAST
TrnID=0x2F4C
OpCode=0
NmFlags=0x1
Rcode=0
QueryCount=1
AnswerCount=0
AuthorityCount=0
```

```
AddressRecCount=0
QuestionRecords:
Name=*               NameType=0x00 (Workstation)
QuestionType=0x21
QuestionClass=0x1

 (ttl 115, id 64316)
14:42:38.310790 192.168.1.1.137 > 192.168.4.2.137:
>>> NBT UDP PACKET(137): QUERY; REQUEST; BROADCAST
TrnID=0x2F4E
OpCode=0
NmFlags=0x1
Rcode=0
QueryCount=1
AnswerCount=0
AuthorityCount=0
AddressRecCount=0
QuestionRecords:
Name=*               NameType=0x00 (Workstation)
QuestionType=0x21
QuestionClass=0x1

 (ttl 115, id 64572)
14:42:38.319716 64.228.81.68.137 > 192.168.4.2.137:
>>> NBT UDP PACKET(137): QUERY; REQUEST; BROADCAST
TrnID=0x2F50
OpCode=0
NmFlags=0x1
Rcode=0
QueryCount=1
AnswerCount=0
AuthorityCount=0
AddressRecCount=0
QuestionRecords:
Name=*               NameType=0x00 (Workstation)
QuestionType=0x21
QuestionClass=0x1

 (ttl 115, id 64828)
14:42:38.326759 169.254.186.43.137 > 192.168.4.2.137:
>>> NBT UDP PACKET(137): QUERY; REQUEST; BROADCAST
TrnID=0x2F52
OpCode=0
NmFlags=0x1
Rcode=0
QueryCount=1
AnswerCount=0
AuthorityCount=0
AddressRecCount=0
QuestionRecords:
Name=*               NameType=0x00 (Workstation)
QuestionType=0x21
QuestionClass=0x1

 (ttl 115, id 65084)
14:42:39.811543 192.168.1.1.137 > 192.168.4.2.137:
>>> NBT UDP PACKET(137): QUERY; REQUEST; BROADCAST
```

```
TrnID=0x2F54
OpCode=0
NmFlags=0x1
Rcode=0
QueryCount=1
AnswerCount=0
AuthorityCount=0
AddressRecCount=0
QuestionRecords:
Name=*               NameType=0x00 (Workstation)
QuestionType=0x21
QuestionClass=0x1

 (ttl 115, id 65340)
14:42:39.819270 169.254.186.43.137 > 192.168.4.2.137:
>>> NBT UDP PACKET(137): QUERY; REQUEST; BROADCAST
TrnID=0x2F56
OpCode=0
NmFlags=0x1
Rcode=0
QueryCount=1
AnswerCount=0
AuthorityCount=0
AddressRecCount=0
QuestionRecords:
Name=*               NameType=0x00 (Workstation)
QuestionType=0x21
QuestionClass=0x1

 (ttl 115, id 61)
14:42:39.822676 64.228.81.68.137 > 192.168.4.2.137:
>>> NBT UDP PACKET(137): QUERY; REQUEST; BROADCAST
TrnID=0x2F58
OpCode=0
NmFlags=0x1
Rcode=0
QueryCount=1
AnswerCount=0
AuthorityCount=0
AddressRecCount=0
QuestionRecords:
Name=*               NameType=0x00 (Workstation)
QuestionType=0x21
QuestionClass=0x1

 (ttl 115, id 317)
14:42:47.373571 64.228.81.68.137 > 192.168.4.3.137:
>>> NBT UDP PACKET(137): QUERY; REQUEST; BROADCAST
TrnID=0x2F62
OpCode=0
NmFlags=0x1
Rcode=0
QueryCount=1
AnswerCount=0
AuthorityCount=0
AddressRecCount=0
QuestionRecords:
```

```
Name=*               NameType=0x00 (Workstation)
QuestionType=0x21
QuestionClass=0x1

 (ttl 115, id 1597)
14:42:48.827024 64.228.81.68.137 > 192.168.4.3.137:
>>> NBT UDP PACKET(137): QUERY; REQUEST; BROADCAST
TrnID=0x2F68
OpCode=0
NmFlags=0x1
Rcode=0
QueryCount=1
AnswerCount=0
AuthorityCount=0
AddressRecCount=0
QuestionRecords:
Name=*               NameType=0x00 (Workstation)
QuestionType=0x21
QuestionClass=0x1

 (ttl 115, id 2365)
14:42:48.838378 169.254.186.43.137 > 192.168.4.3.137:
>>> NBT UDP PACKET(137): QUERY; REQUEST; BROADCAST
TrnID=0x2F6A
OpCode=0
NmFlags=0x1
Rcode=0
QueryCount=1
AnswerCount=0
AuthorityCount=0
AddressRecCount=0
QuestionRecords:
Name=*               NameType=0x00 (Workstation)
QuestionType=0x21
QuestionClass=0x1

 (ttl 115, id 2621)
14:42:48.842736 192.168.1.1.137 > 192.168.4.3.137:
>>> NBT UDP PACKET(137): QUERY; REQUEST; BROADCAST
TrnID=0x2F6C
OpCode=0
NmFlags=0x1
Rcode=0
QueryCount=1
AnswerCount=0
AuthorityCount=0
AddressRecCount=0
QuestionRecords:
Name=*               NameType=0x00 (Workstation)
QuestionType=0x21
QuestionClass=0x1

 (ttl 115, id 2877)
14:42:50.325561 64.228.81.68.137 > 192.168.4.3.137:
>>> NBT UDP PACKET(137): QUERY; REQUEST; BROADCAST
TrnID=0x2F6E
OpCode=0
```

```
NmFlags=0x1
Rcode=0
QueryCount=1
AnswerCount=0
AuthorityCount=0
AddressRecCount=0
QuestionRecords:
Name=*               NameType=0x00 (Workstation)
QuestionType=0x21
QuestionClass=0x1

 (ttl 115, id 3133)
14:42:50.334527 192.168.1.1.137 > 192.168.4.3.137:
>>> NBT UDP PACKET(137): QUERY; REQUEST; BROADCAST
TrnID=0x2F70
OpCode=0
NmFlags=0x1
Rcode=0
QueryCount=1
AnswerCount=0
AuthorityCount=0
AddressRecCount=0
QuestionRecords:
Name=*               NameType=0x00 (Workstation)
QuestionType=0x21
QuestionClass=0x1

 (ttl 115, id 3389)
14:42:50.342429 169.254.186.43.137 > 192.168.4.3.137:
>>> NBT UDP PACKET(137): QUERY; REQUEST; BROADCAST
TrnID=0x2F72
OpCode=0
NmFlags=0x1
Rcode=0
QueryCount=1
AnswerCount=0
AuthorityCount=0
AddressRecCount=0
QuestionRecords:
Name=*               NameType=0x00 (Workstation)
QuestionType=0x21
QuestionClass=0x1

 (ttl 115, id 3645)
14:42:57.889814 192.168.1.1.137 > 192.168.4.4.137:
>>> NBT UDP PACKET(137): QUERY; REQUEST; BROADCAST
TrnID=0x2F7C
OpCode=0
NmFlags=0x1
Rcode=0
QueryCount=1
AnswerCount=0
AuthorityCount=0
AddressRecCount=0
QuestionRecords:
Name=*               NameType=0x00 (Workstation)
QuestionType=0x21
```

```
QuestionClass=0x1

 (ttl 115, id 5181)
14:42:59.350244 192.168.1.1.137 > 192.168.4.4.137:
>>> NBT UDP PACKET(137): QUERY; REQUEST; BROADCAST
TrnID=0x2F82
OpCode=0
NmFlags=0x1
Rcode=0
QueryCount=1
AnswerCount=0
AuthorityCount=0
AddressRecCount=0
QuestionRecords:
Name=*               NameType=0x00 (Workstation)
QuestionType=0x21
QuestionClass=0x1

 (ttl 115, id 5949)
14:42:59.356791 169.254.186.43.137 > 192.168.4.4.137:
>>> NBT UDP PACKET(137): QUERY; REQUEST; BROADCAST
TrnID=0x2F84
OpCode=0
NmFlags=0x1
Rcode=0
QueryCount=1
AnswerCount=0
AuthorityCount=0
AddressRecCount=0
QuestionRecords:
Name=*               NameType=0x00 (Workstation)
QuestionType=0x21
QuestionClass=0x1

 (ttl 115, id 6205)
14:42:59.362479 64.228.81.68.137 > 192.168.4.4.137:
>>> NBT UDP PACKET(137): QUERY; REQUEST; BROADCAST
TrnID=0x2F86
OpCode=0
NmFlags=0x1
Rcode=0
QueryCount=1
AnswerCount=0
AuthorityCount=0
AddressRecCount=0
QuestionRecords:
Name=*               NameType=0x00 (Workstation)
QuestionType=0x21
QuestionClass=0x1

 (ttl 115, id 6461)
14:43:00.854288 169.254.186.43.137 > 192.168.4.4.137:
>>> NBT UDP PACKET(137): QUERY; REQUEST; BROADCAST
TrnID=0x2F88
OpCode=0
NmFlags=0x1
Rcode=0
```

```
QueryCount=1
AnswerCount=0
AuthorityCount=0
AddressRecCount=0
QuestionRecords:
Name=*               NameType=0x00 (Workstation)
QuestionType=0x21
QuestionClass=0x1

 (ttl 115, id 6717)
14:43:00.857553 192.168.1.1.137 > 192.168.4.4.137:
>>> NBT UDP PACKET(137): QUERY; REQUEST; BROADCAST
TrnID=0x2F8A
OpCode=0
NmFlags=0x1
Rcode=0
QueryCount=1
AnswerCount=0
AuthorityCount=0
AddressRecCount=0
QuestionRecords:
Name=*               NameType=0x00 (Workstation)
QuestionType=0x21
QuestionClass=0x1

 (ttl 115, id 6973)
14:43:00.863249 64.228.81.68.137 > 192.168.4.4.137:
>>> NBT UDP PACKET(137): QUERY; REQUEST; BROADCAST
TrnID=0x2F8C
OpCode=0
NmFlags=0x1
Rcode=0
QueryCount=1
AnswerCount=0
AuthorityCount=0
AddressRecCount=0
QuestionRecords:
Name=*               NameType=0x00 (Workstation)
QuestionType=0x21
QuestionClass=0x1

 (ttl 115, id 7229)
```

This trace comes from the home network. This trace was generated by a netbios-aware version of tcpdump.

This attacker would need a response from a machine in order to obtain any useful information. The attacker in this case has disguised this attack with a few bogus IP addresses 169.254.186.43 and 192.168.1.1. The real IP address is 64.228.81.68. These are probably crafted packets that came from the same machine; notice that the time-to-live value is the same for all source IP's and that the IP header id increments by 256 between many of the packets even though the source address is supposedly different. Also notice that the netbios transaction id (TrnID) increases by 2 in many cases, even between different source IP addresses.

The attacker is targeting udp port 137. This is probably an attempt to divulge information such as the machine name, domain name, services running, username logged in, etc. This is a general scan of the entire network. This scan was blocked by the firewall.

Cert incident notes about unprotected windows machines:
http://www.cert.org/incident_notes/IN-2000-02.html

Severity:

criticality:	directed at windows desktops:	2
lethality:	informational:	2
system countermeasures:	Modern OS:	4
network countermeasures:	Restrictive Firewall:	4
		-
		-4

Multiple choice question:

In the above detect which source IP addresses are likely to be spoofed?

A) 192.168.1.1
B) 169.254.186.43
C) 64.228.81.68
D) Both A and B

ANSWER: D

Detect #5

```
15:38:29.163359 210.112.192.74.3699 > 192.168.4.1.98: S 2100220655:2100220655(0)
win 32120 <mss 1460,sackOK,timestamp 76173545 0,nop,wscale 0> (DF) (ttl 51, id
24326)
15:38:29.187050 210.112.192.74.3701 > 192.168.4.2.98: S 2097719354:2097719354(0)
win 32120 <mss 1460,sackOK,timestamp 76173545 0,nop,wscale 0> (DF) (ttl 51, id
24327)
15:38:32.161638 210.112.192.74.3699 > 192.168.4.1.98: S 2100220655:2100220655(0)
win 32120 <mss 1460,sackOK,timestamp 76173845 0,nop,wscale 0> (DF) (ttl 51, id
27277)
15:38:32.182925 210.112.192.74.3701 > 192.168.4.2.98: S 2097719354:2097719354(0)
win 32120 <mss 1460,sackOK,timestamp 76173845 0,nop,wscale 0> (DF) (ttl 51, id
27278)
```

This trace comes from the home network. It was generated by tcpdump.

The attacker would want the tcp handshake to occur so they would not spoof the source IP address. There is no evidence of packet crafting.

This individual is attempting to connect to tcp port 98. This is probably an attempt to scan for the linux-conf service to see if they can obtain remote root access. This was a scan of two hosts in our IP range.

The defenses are adequate, this scan was blocked by the firewall.

A brief note about linuxconf:
http://www.securityfocus.com/templates/archive.pike?list=1&date=1998-08-22&msg=19980826214112.A10022@boehm.org

Severity:

criticality:	directed at servers:	5
lethality:	root access:	5
system countermeasures:	Modern OS not prone to attack:	5
network countermeasures:	Restrictive Firewall:	4
		-
		1

Multiple choice question:

The source IP address in this detect is:

A) 10.62.13.5
B) 210.112.192.74
C) 63.57.255.3
D) 206.9.36.4

ANSWER: B

Detect #6

```
09:23:12.314107 208.36.197.44.1028 > 192.168.4.3.161:  |30|2a|02|01SNMPv1
|04|06|a0|1dGetRequest(29)|02|01|02|01|02|01|30|12
|30|10|06|0c.1.3.6.1.4.1.11.2.4.3.10.6.0|05|00 (ttl 127, id 41984)
0x0000       4500 0048 a400 0000 7f11 6c5c d024 c52c  E..H......l\.$.,
0x0010       d024 c5d2 0404 00a1 0034 8d34 302a 0201  .$.......4.40*..
0x0020       0004 0670 7562 6c69 63a0 1d02 0102 0201  ...public.......
0x0030       0002 0100 3012 3010 060c 2b06 0104 010b  ....0.0...+.....
0x0040       0204 030a 0600 0500                      ........
09:23:12.317638 208.36.197.44.1028 > 192.168.4.2.161:  |30|2a|02|01SNMPv1
|04|06|a0|1dGetRequest(29)|02|01|02|01|02|01|30|12
|30|10|06|0c.1.3.6.1.4.1.11.2.4.3.10.6.0|05|00 (ttl 127, id 42240)
0x0000       4500 0048 a500 0000 7f11 6b5d d024 c52c  E..H......k].$.,
0x0010       d024 c5d1 0404 00a1 0034 8d35 302a 0201  .$.......4.50*..
0x0020       0004 0670 7562 6c69 63a0 1d02 0102 0201  ...public.......
0x0030       0002 0100 3012 3010 060c 2b06 0104 010b  ....0.0...+.....
0x0040       0204 030a 0600 0500                      ........
09:23:12.320881 208.36.197.44.1028 > 192.168.4.1.161:  |30|2a|02|01SNMPv1
|04|06|a0|1dGetRequest(29)|02|01|02|01|02|01|30|12
|30|10|06|0c.1.3.6.1.4.1.11.2.4.3.10.6.0|05|00 (ttl 127, id 42496)
0x0000       4500 0048 a600 0000 7f11 6a5e d024 c52c  E..H......j^.$.,
0x0010       d024 c5d0 0404 00a1 0034 8d36 302a 0201  .$.......4.60*..
0x0020       0004 0670 7562 6c69 63a0 1d02 0102 0201  ...public.......
0x0030       0002 0100 3012 3010 060c 2b06 0104 010b  ....0.0...+.....
0x0040       0204 030a 0600 0500                      ........
```

```
09:23:12.344632 208.36.197.44.1028 > 192.168.4.4.161:  |30|2a|02|01SNMPv1
|04|06|a0|1dGetRequest(29)|02|01|02|01|02|01|30|12
|30|10|06|0c.1.3.6.1.4.1.11.2.4.3.10.6.0|05|00 (ttl 127, id 41728)
0x0000       4500 0048 a300 0000 7f11 6d5b d024 c52c  E..H......m[.$.,
0x0010       d024 c5d3 0404 00a1 0034 8d33 302a 0201  .$.......4.30*..
0x0020       0004 0670 7562 6c69 63a0 1d02 0102 0201  ...public.......
0x0030       0002 0100 3012 3010 060c 2b06 0104 010b  ....0.0...+.....
0x0040       0204 030a 0600 0500                      ........
09:23:17.408214 208.36.197.44.1028 > 192.168.4.4.161:  |30|3b|02|01SNMPv1
|04|06|a0|2eGetRequest(39)|02|01|02|01|02|01|30|23
|30|11|06|0d.1.3.6.1.4.1.11.2.4.3.8.3.2.0|05|00 |30|0e|06|0a.1.3.6.1.2.1[|snmp]
(ttl 127, id 46081)
0x0000       4500 0059 b401 0000 7f11 5c49 d024 c52c  E..Y......\I.$.,
0x0010       d024 c5d3 0404 00a1 0045 5f79 303b 0201  .$.......E_y0;..
0x0020       0004 0670 7562 6c69 63a0 2e02 0101 0201  ...public.......
0x0030       0002 0100 3023 3011 060d 2b06 0104 010b  ....0#0...+.....
0x0040       0204 0308 0302 0005 0030 0e06 0a2b 0601  .........0...+..
0x0050       0201                                     ..
09:23:17.412773 208.36.197.44.1028 > 192.168.4.3.161:  |30|3b|02|01SNMPv1
|04|06|a0|2eGetRequest(39)|02|01|02|01|02|01|30|23
|30|11|06|0d.1.3.6.1.4.1.11.2.4.3.8.3.2.0|05|00 |30|0e|06|0a.1.3.6.1.2.1[|snmp]
(ttl 127, id 46337)
0x0000       4500 0059 b501 0000 7f11 5b4a d024 c52c  E..Y......[J.$.,
0x0010       d024 c5d2 0404 00a1 0045 5f7a 303b 0201  .$.......E_z0;..
0x0020       0004 0670 7562 6c69 63a0 2e02 0101 0201  ...public.......
0x0030       0002 0100 3023 3011 060d 2b06 0104 010b  ....0#0...+.....
0x0040       0204 0308 0302 0005 0030 0e06 0a2b 0601  .........0...+..
0x0050       0201                                     ..
09:23:17.416036 208.36.197.44.1028 > 192.168.4.2.161:  |30|3b|02|01SNMPv1
|04|06|a0|2eGetRequest(39)|02|01|02|01|02|01|30|23
|30|11|06|0d.1.3.6.1.4.1.11.2.4.3.8.3.2.0|05|00 |30|0e|06|0a.1.3.6.1.2.1[|snmp]
(ttl 127, id 46593)
0x0000       4500 0059 b601 0000 7f11 5a4b d024 c52c  E..Y......ZK.$.,
0x0010       d024 c5d1 0404 00a1 0045 5f7b 303b 0201  .$.......E_{0;..
0x0020       0004 0670 7562 6c69 63a0 2e02 0101 0201  ...public.......
0x0030       0002 0100 3023 3011 060d 2b06 0104 010b  ....0#0...+.....
0x0040       0204 0308 0302 0005 0030 0e06 0a2b 0601  .........0...+..
0x0050       0201                                     ..
09:23:17.419564 208.36.197.44.1028 > 192.168.4.1.161:  |30|3b|02|01SNMPv1
|04|06|a0|2eGetRequest(39)|02|01|02|01|02|01|30|23
|30|11|06|0d.1.3.6.1.4.1.11.2.4.3.8.3.2.0|05|00 |30|0e|06|0a.1.3.6.1.2.1[|snmp]
(ttl 127, id 46849)
0x0000       4500 0059 b701 0000 7f11 594c d024 c52c  E..Y......YL.$.,
0x0010       d024 c5d0 0404 00a1 0045 5f7c 303b 0201  .$.......E_|0;..
0x0020       0004 0670 7562 6c69 63a0 2e02 0101 0201  ...public.......
0x0030       0002 0100 3023 3011 060d 2b06 0104 010b  ....0#0...+.....
0x0040       0204 0308 0302 0005 0030 0e06 0a2b 0601  .........0...+..
0x0050       0201                                     ..


01/05/2000 23:43:32.496919 ep0 @0:1 b 208.36.197.44,1028 -> 192.168.4.4,161 PR
udp len 20 72  IN
01/05/2000 23:43:32.501117 ep0 @0:1 b 208.36.197.44,1028 -> 192.168.4.3,161 PR
udp len 20 72  IN
01/05/2000 23:43:32.504407 ep0 @0:1 b 208.36.197.44,1028 -> 192.168.4.2,161 PR
udp len 20 72  IN
```

```
01/05/2000 23:43:32.507943 ep0 @0:1 b 208.36.197.44,1028 -> 192.168.4.1,161 PR
udp len 20 72  IN
01/05/2000 23:43:37.069056 ep0 @0:1 b 208.36.197.44,1028 -> 192.168.4.4,161 PR
udp len 20 89  IN
01/05/2000 23:43:37.073289 ep0 @0:1 b 208.36.197.44,1028 -> 192.168.4.3,161 PR
udp len 20 89  IN
01/05/2000 23:43:37.076553 ep0 @0:1 b 208.36.197.44,1028 -> 192.168.4.2,161 PR
udp len 20 89  IN
01/05/2000 23:43:37.080131 ep0 @0:1 b 208.36.197.44,1028 -> 192.168.4.1,161 PR
udp len 20 89  IN
02/05/2000 07:37:18.263416 ep0 @0:1 b 208.36.197.44,1028 -> 192.168.4.1,161 PR
udp len 20 72  IN
02/05/2000 07:37:18.276384 ep0 @0:1 b 208.36.197.44,1028 -> 192.168.4.4,161 PR
udp len 20 72  IN
02/05/2000 07:37:18.290033 ep0 @0:1 b 208.36.197.44,1028 -> 192.168.4.3,161 PR
udp len 20 72  IN
02/05/2000 07:37:18.304598 ep0 @0:1 b 208.36.197.44,1028 -> 192.168.4.2,161 PR
udp len 20 72  IN
02/05/2000 07:37:23.100737 ep0 @0:1 b 208.36.197.44,1028 -> 192.168.4.4,161 PR
udp len 20 89  IN
02/05/2000 07:37:23.104060 ep0 @0:1 b 208.36.197.44,1028 -> 192.168.4.3,161 PR
udp len 20 89  IN
02/05/2000 07:37:23.108527 ep0 @0:1 b 208.36.197.44,1028 -> 192.168.4.2,161 PR
udp len 20 89  IN
02/05/2000 07:37:23.111853 ep0 @0:1 b 208.36.197.44,1028 -> 192.168.4.1,161 PR
udp len 20 89  IN
02/05/2000 20:54:51.486086 ep0 @0:1 b 208.36.197.44,1028 -> 192.168.4.4,161 PR
udp len 20 72  IN
02/05/2000 20:54:51.490594 ep0 @0:1 b 208.36.197.44,1028 -> 192.168.4.3,161 PR
udp len 20 72  IN
02/05/2000 20:54:51.493637 ep0 @0:1 b 208.36.197.44,1028 -> 192.168.4.2,161 PR
udp len 20 72  IN
02/05/2000 20:54:51.497132 ep0 @0:1 b 208.36.197.44,1028 -> 192.168.4.1,161 PR
udp len 20 72  IN
02/05/2000 20:54:56.958993 ep0 @0:1 b 208.36.197.44,1028 -> 192.168.4.4,161 PR
udp len 20 89  IN
02/05/2000 20:54:56.961419 ep0 @0:1 b 208.36.197.44,1028 -> 192.168.4.3,161 PR
udp len 20 89  IN
02/05/2000 20:54:56.964703 ep0 @0:1 b 208.36.197.44,1028 -> 192.168.4.2,161 PR
udp len 20 89  IN
02/05/2000 20:54:56.969412 ep0 @0:1 b 208.36.197.44,1028 -> 192.168.4.1,161 PR
udp len 20 89  IN
```

This trace was from the home network. The first was generated using tcpdump and dumping out the hex and the ASCII. The second section is output from ipmon (ip-filter).

In this detect the attacker would need the tcp handshake to occur, so I propose that the source IP address is not spoofed. There is no evidence of crafting in these detects.

The individual is making connection attempts to tcp port 161. This is a pretty obvious attempt to send the defualt snmp community string "public" (see ASCII output in first section of detect). The attacker is attempting to gain information about the network using this mechanism. This was a general scan of the entire network. This attack was blocked at the firewall.

This detect has been seen a few times on this network, and is a fairly common detect: http://cve.mitre.org/ CAN-1999-0517

Severity of this attack:

criticality:	directed at desktops and servers:	3
lethality:	informational/limited access:	3
system countermeasures:	Modern OS:	4
network countermeasures:	Restrictive Firewall:	4
		-
		-2

Multiple choice question:

Snmp uses port 161, protocol?

A) udp
B) icmp
C) tcp
D) gre

ANSWER: A

Detect #7

```
21:50:09.036605 212.141.67.35.4449 > 192.168.4.1.1080: S 26795738:26795738(0)
win 8192 <mss 536,nop,nop,sackOK> (DF) (ttl 111, id 55495)
21:50:09.360567 212.141.67.35.4453 > 192.168.4.2.1080: S 26795994:26795994(0)
win 8192 <mss 536,nop,nop,sackOK> (DF) (ttl 111, id 60871)
21:50:09.402292 212.141.67.35.4454 > 192.168.4.3.1080: S 26795995:26795995(0)
win 8192 <mss 536,nop,nop,sackOK> (DF) (ttl 111, id 61127)
21:50:09.418964 212.141.67.35.4455 > 192.168.4.4.1080: S 26795997:26795997(0)
win 8192 <mss 536,nop,nop,sackOK> (DF) (ttl 111, id 61383)
21:50:11.992981 212.141.67.35.4449 > 192.168.4.1.1080: S 26795738:26795738(0)
win 8192 <mss 536,nop,nop,sackOK> (DF) (ttl 111, id 34504)
21:50:12.260793 212.141.67.35.4453 > 192.168.4.2.1080: S 26795994:26795994(0)
win 8192 <mss 536,nop,nop,sackOK> (DF) (ttl 111, id 39880)
21:50:12.418426 212.141.67.35.4455 > 192.168.4.4.1080: S 26795997:26795997(0)
win 8192 <mss 536,nop,nop,sackOK> (DF) (ttl 111, id 40136)
21:50:12.462317 212.141.67.35.4454 > 192.168.4.3.1080: S 26795995:26795995(0)
win 8192 <mss 536,nop,nop,sackOK> (DF) (ttl 111, id 41672)
```

This detect came from the home network. It was generated using tcpdump.

The attacker would want the tcp handshake to occur in this probe, so I propose that the source IP address is not spoofed. There is not any evidence of packet crafting in this detect.

The attacker is attempting to make a connection to tcp port 1080; they scanned across all of our internet-facing hosts. This is either a probe for unix machines that are running a socks proxy or the winhole trojan (a trojan version of wingate) that runs socks on windows. It is difficult to tell what the attacker was after here as the firewall blocked the packets before any of the targeted

machines (both windows and unix machines) could respond. In both cases they attempting to use a proxy; they could be attempting to find open socks daemons in use as a relay to attack someone else later. Fortunately the firewall blocked this attack.

****** related URL
Cert advisory on wingate:
http://www.cert.org/vul_notes/VN-98.03.WinGate.html

Cert incident note (mention of socks in scans):
http://www.cert.org/incident_notes/IN-98.02.html

Severity of this attack:

criticality:	directed at desktops and servers:	3
lethality:	masquerade as us:	3
system countermeasures:	Modern OS:	4
network countermeasures:	Restrictive Firewall:	4
		-
		-2

Multiple choice question:

Which IP responded to the probe?

A) 192.168.4.1
B) 192.168.4.2
C) 192.168.4.3
D) None of the above

ANSWER: D (None of them responded)

Detect #8

```
03:43:07.826435 207.78.247.50.65535 > 192.168.4.1.8080: S
1548877824:1548877824(0) win 512 (ttl 242, id 23634)
03:43:07.875777 207.78.247.50.65535 > 192.168.4.2.8080: S
1548877824:1548877824(0) win 512 (ttl 242, id 23634)
03:43:07.928655 207.78.247.50.65535 > 192.168.4.3.8080: S
1548877824:1548877824(0) win 512 (ttl 242, id 23634)
03:43:07.979379 207.78.247.50.65535 > 192.168.4.4.8080: S
1548877824:1548877824(0) win 512 (ttl 242, id 23634)
```

This detect was generated on the home network. It was generated using tcpdump.

The attacker wants the tcp handshake to occur, so I do not believe that the source IP address is spoofed. These are crafted packets. The IP header id is the same, the sequence number is the same and the source port is the same even though the connection attempts are to different hosts.

The intruder is attempting to make a connection to tcp port 8080. This could be a scan to check for the ringzero trojan. It could also be a check for web proxies as they sometimes run on this port.

In these cases the object is to find web proxies and probably use them as a relay to attack someone. Also sometimes tcp port 8080 is setup as the administration port for the web server which could also be subject to attack if not protected. This scan was across the entire network. The firewall blocked this scan so the defenses are sufficient.

Some detects on the sans site of 8080:
http://www.sans.org/y2k/031700.htm

Severity of this attack:

criticality:	directed at desktops and servers:	3
lethality:	masquerade as us:	3
system countermeasures:	Modern OS:	4
network countermeasures:	Restrictive Firewall:	4
		-
		-2

Multiple choice question:

The packets in this detect look crafted because:

A) The IP header id is the same for all connections
B) The tcp sequence numbers are the same for all connections
C) The source port is the same for all connections
D) All of the above

ANSWER: D

Detect #9

```
04:56:48.144695 160.12.167.102.111 > 192.168.4.1.111: SF 490453792:490453792(0)
win 1028 (ttl 26, id 39426)
04:56:48.163728 160.12.167.102.111 > 192.168.4.2.111: SF 490453792:490453792(0)
win 1028 (ttl 26, id 39426)
04:56:48.225630 160.12.167.102.111 > 192.168.4.3.111: SF 490453792:490453792(0)
win 1028 (ttl 26, id 39426)
04:56:48.227797 160.12.167.102.111 > 192.168.4.4.111: SF 490453792:490453792(0)
win 1028 (ttl 26, id 39426)
```

This detect comes from the home network, it was generated using tcpdump.

The attacker is not wanting the tcp handshake to occur; but they are expecting a response so I propose that the source IP address is not spoofed. We are dealing with crafted packets as they have the SYN and FIN flags set, which is something that never occurs in a normal connection. Also notice that the sequence numbers and the IP header id's are not changing even though we are connecting to different machines.

The attacker is attempting to scan with SYN-FIN across all the hosts on our network to detemine if there is an rpc daemon running. The purpose of this is probably attempt a buffer overrun and gain

control of one or more of the machines. A useful side-effect of using SYN-FIN to scan is that it may aid in fingerprinting the Operating System. Fortunately for us the firewall again comes to the rescue and blocks these packets.

Cert incident note about attacks involving RPC:
http://www.cert.org/incident_notes/IN-99-04.html

Severity of this attack:

criticality:	directed at desktops and servers:	3
lethality:	root access:	5
system countermeasures:	Modern OS:	4
network countermeasures:	Restrictive Firewall:	4
		-
		0

Multiple choice question:

Which of the follwing in the above detect are probably crafted?

A) tcp flags
B) sequence numbers
C) IP header id's
D) All of the above

ANSWER: D

Detect #10

```
01:37:55.773074 195.13.78.10.53 > 192.168.4.1.53: SF 1446533203:1446533203(0)
win 1028 (ttl 28, id 39426)
01:37:55.792472 195.13.78.10.53 > 192.168.4.2.53: SF 1446533203:1446533203(0)
win 1028 (ttl 28, id 39426)
01:37:55.870122 195.13.78.10.53 > 192.168.4.3.53: SF 1446533203:1446533203(0)
win 1028 (ttl 28, id 39426)
01:37:55.887014 195.13.78.10.53 > 192.168.4.4.53: SF 1446533203:1446533203(0)
win 1028 (ttl 28, id 39426)
06:40:48.600278 203.233.103.188.53 > 192.168.4.1.53: SF 155489164:155489164(0)
win 1028 (ttl 30, id 39426)
06:40:48.645230 203.233.103.188.53 > 192.168.4.2.53: SF 155489164:155489164(0)
win 1028 (ttl 30, id 39426)
06:40:48.686939 203.233.103.188.53 > 192.168.4.3.53: SF 155489164:155489164(0)
win 1028 (ttl 30, id 39426)
06:40:48.720754 203.233.103.188.53 > 192.168.4.4.53: SF 155489164:155489164(0)
win 1028 (ttl 30, id 39426)
```

This detect comes from the home network, it was created using tcpdump.

Like in the previous detect the attacker(s) have sent a packet with both the SYN and FIN tcp flags set, they again are looking for a response. We are again looking at crafted packets: the SYN-FIN is set, the sequences numbers are constant within the two groups, and the IP header id is the same in both groups.

The intruder is attempting to see if any of the hosts will respond to a probe to tcp port 53. The intruder would like to find a dns server running that they might later attempt a zone transfer or a buffer overrun. They also may be working to fingerprint the Operating System as the SYN and FIN flags are set. The firewall blocked this attack so the defenses are sufficient.

These two scans look to be related: the IP header id was the same for both scans. Could be the same person or the same tool that is attacking us from both sites.

URL's about security problems with Bind:
http://www.cert.org/advisories/CA-98.05.bind_problems.html
http://www.cert.org/advisories/CA-97.22.bind.html

Severity of this attack:

criticality:	directed at desktops and servers:	3
lethality:	root access:	5
system countermeasures:	Modern OS:	4
network countermeasures:	Restrictive Firewall:	4
		-
		0

Multiple choice question:

This was an attempt to?

A) do a zone transfer
B) scan for bind daemons
C) do a buffer overrun to obtain root access
D) None of the above

ANSWER: B

Bonus Detect

```
18:15:02.167522 192.168.4.1.32796 > 194.119.192.34.53:  43425 A? dns.wind.it.
(29) (DF) (ttl 254, id 57171)
18:15:02.418951 194.119.192.34.53 > 192.168.4.1.32796:  43425 q: dns.wind.it.
1/3/3 dns.wind.it. A 212.245.255.2 (156) (ttl 15, id 13128)
18:15:02.461910 194.119.192.34 > 192.168.4.1: icmp: echo request (DF) (ttl 240,
id 13129)
18:15:28.776437 194.119.192.34 > 192.168.4.1: icmp: echo request (DF) (ttl 240,
id 13970)
```

This detect comes from the home network and was generated using tcpdump.

Based on the fact that the first transaction is successful the second transaction's (icmp traffic) source IP address is not spoofed and there is no evidence of crafting either.

The dns server that I queried is attempting to icmp ping my host (targeted)

after a successful dns transaction. After talking with a couple individuals about this detect I am still as puzzled as when I started. This looks like an automatic ping by the server after the dns query. Notice the IP header id is incremented by only 1 and the time between the dns response and the ping is .05 seconds! This is much too fast for a human, it must be automatic! I have been informed of a program for evaluating dns root servers called skitter that contacts the machine asking the question; however, I am also told that it does not come back from the dns server itself and that it does not use icmp. Hmmmm...
The firewall did block the icmp echo request so it appears that defenses are sufficient. Not I or anyone I had conversed with had seen this before.

Severity of this attack:

criticality:	directed at server:	5
lethality:	informational?/unknown:	3
system countermeasures:	Modern OS:	4
network countermeasures:	Restrictive Firewall:	4
		-
		0

Multiple choice question:

Based on the timestamp and the IP header id the icmp packets were:

A) An automated response.
B) Erroniously generated traffic, a deficiency in the dns server's tcp/ip stack.
C) A totally weird computer thing.
D) All of the above

ANSWER: While there could be a good argument for C, I would go with A

Appendix

IP filter log format

This is output from the ipmon command (component of ip-filter)

03/05/2000 21:24:51.486477 ep0 @0:1 b 32.97.182.250,443 -> 172.31.98.3,1256 PR tcp len 20 80 -AFP IN

BREAKDOWN OF EACH FIELD:

03/05/2000 - is the date packet was received <day>/<month>/<year>

21:24:51.486477 - is the time packet was received <hour>:<minute>:<second>

ep0 - interface that packet was processed on

@0:1 - group and rule number that packet was processed on @<group>:<rule>

b - action that was taken b=blocked p=passed

32.97.182.250,443 - <source IP>,<source port>

172.31.98.3,1256 - <destination IP>,<destination port>

PR tcp - protocol name or number in this case tcp protocol

len 20 80 - len <header length> <total packet length>

-AFP - tcp flags that were set in this case ack-fin-push

IN - direction that the packet is going either IN or OUT

Last Updated: July 25th, 2017

Upcoming Training

SANS San Antonio 2017	San Antonio, TX	Aug 06, 2017 - Aug 11, 2017	Live Event
SANS Boston 2017	Boston, MA	Aug 07, 2017 - Aug 12, 2017	Live Event
SANS Virginia Beach 2017	Virginia Beach, VA	Aug 21, 2017 - Sep 01, 2017	Live Event
SANS Adelaide 2017	Adelaide, Australia	Aug 21, 2017 - Aug 26, 2017	Live Event
SANS Network Security 2017	Las Vegas, NV	Sep 10, 2017 - Sep 17, 2017	Live Event
SANS vLive - SEC503: Intrusion Detection In-Depth	SEC503 - 201709,	Sep 11, 2017 - Oct 18, 2017	vLive
SANS 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
Baltimore Fall 2017 - SEC503: Intrusion Detection In-Depth	Baltimore, MD	Sep 25, 2017 - Sep 30, 2017	vLive
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