Global Information Assurance Certification Paper

*** Northcutt, I like the work that has been put into determining history of some of these traces. Also this student is quite comfortable with a variety of traces. You know you need a real life when you can look at a stream and know instantly, that must be a "Binette @home" (trace 7). There is an analysis process, accuracy if fine, lists his sources. 80 *

Harald Skotnes
10 detects for the IDIC in Orlando.

Detect 1.

Background:
Cisco router ACL wich dēnies access to tcp port 12346 on 129.242.*.* from the address 207.53.185.*. This goes on for several days, but not every day in the period.

Description: Port scanning for port 12346.

Technique:
The technique for this scan seems to be low and slow. The intruder is using several days and is only probing a couple of machines a day.

Intent:
This is a scan to find out if trojans like Gabanbus and Netbus is listening on port 12346 on a windows machine. If so an intruder can deploy other trojans or use it to find passwords on the machine. The intruder can then take full control over the machine.
This is an information phase and if successful the attacker will be back.

Severity: None, the ACL stops it.

```
Apr  7 10:31:11 hyper-gw 56484: Apr  7 10:31:10.255: %SEC-6-IPACCESSLOGP: list
111 denied tcp 207.53.185.169(1164) () -> 129.242.152.145(12346), 1 packet
Apr  7 11:04:09 hyper-gw 56534: Apr  7 11:04:08.110: %SEC-6-IPACCESSLOGP: list
111 denied tcp 207.53.185.169(3998) (6) -> 129.242.13.151(12346), 1 packet

Apr 10 11:08:23 hyper-gw 60548: Apr 10 11:08:22.792: %SEC-6-IPACCESSLOGP: list
111 denied tcp 207.53.185.171(1820) () -> 129.242.13.151(12346), 1 packet
Apr 10 12:41:42 hyper-gw 60681: Apr 10 12:41:41.440: %SEC-6-IPACCESSLOGP: list
111 denied tcp 207.53.185.171(4187) () -> 129.242.13.189(12346), 1 packet
Apr 10 12:46:43 hyper-gw 60690: Apr 10 12:46:42.085: %SEC-6-IPACCESSLOGP: list
111 denied tcp 207.53.185.171(4187) () -> 129.242.13.189(12346), 2 packets
Apr 10 13:22:41 hyper-gw 60755: Apr 10 13:22:40.541: %SEC-6-IPACCESSLOGP: list
111 denied tcp 207.53.185.171(4350) () -> 129.242.219.27(12346), 1 packet
Apr 10 13:27:43 hyper-gw 60765: Apr 10 13:27:42.746: %SEC-6-IPACCESSLOGP: list
111 denied tcp 207.53.185.171(4350) () -> 129.242.219.27(12346), 2 packets

Apr 11 12:35:51 hyper-gw 62403: Apr 11 12:35:50.640: %SEC-6-IPACCESSLOGP: list
111 denied tcp 207.53.185.196(3470) () -> 129.242.31.47(12346), 1 packet
Apr 11 12:36:17 hyper-gw 62405: Apr 11 12:36:16.380: %SEC-6-IPACCESSLOGP: list
111 denied tcp 207.53.185.196(3684) () -> 129.242.219.27(12346), 1 packet
Apr 11 12:41:54 hyper-gw 62421: Apr 11 12:41:53.325: %SEC-6-IPACCESSLOGP: list
111 denied tcp 207.53.185.196(3684) () -> 129.242.219.27(12346), 3 packets
Apr 11 14:03:10 hyper-gw 62544: Apr 11 14:03:09.965: %SEC-6-IPACCESSLOGP: list
111 denied tcp 207.53.185.196(1987) () -> 129.242.200.110(12346), 1 packet
Apr 11 14:09:07 hyper-gw 62552: Apr 11 14:09:06.666: %SEC-6-IPACCESSLOGP: list
111 denied tcp 207.53.185.196(1987) () -> 129.242.200.110(12346), 3 packets
```

```
Apr 11 15:20:02 hyper-gw 62664: Apr 11 15:20:01.771: %SEC-6-IPACCESSLOGP: list
111 denied tcp 207.53.185.176(2219) () -> 129.242.152.183(12346), 1 packet
Apr 11 15:25:08 hyper-gw 62673: Apr 11 15:25:07.888: %SEC-6-IPACCESSLOGP: list
111 denied tcp 207.53.185.176(2219) () -> 129.242.152.183(12346), 2 packets
Apr 11 16:42:59 hyper-gw 62786: Apr 11 16:42:58.188: %SEC-6-IPACCESSLOGP: list
111 denied tcp 207.53.185.176(2973) () -> 129.242.26.219(12346), 1 packet
Apr 11 16:48:10 hyper-gw 62795: Apr 11 16:48:09.224: %SEC-6-IPACCESSLOGP: list
111 denied tcp 207.53.185.176(2973) () -> 129.242.26.219(12346), 3 packets

Apr 12 15:04:19 hyper-gw 65356: Apr 12 15:04:18.572: %SEC-6-IPACCESSLOGP: list
111 denied tcp 207.53.185.199(2953) () -> 129.242.209.106(12346), 1 packet
Apr 12 15:09:07 hyper-gw 65368: Apr 12 15:09:06.737: %SEC-6-IPACCESSLOGP: list
111 denied tcp 207.53.185.199(2954) () -> 129.242.209.106(20034), 3 packets
Apr 12 15:09:32 hyper-gw 65369: Apr 12 15:09:31.289: %SEC-6-IPACCESSLOGP: list
111 denied tcp 207.53.185.199(2953) () -> 129.242.209.106(12346), 3 packets
```

Detect 2.

Background:
Detected by Shadow.

Description: Tcp port scan.

Technique:
What we see here is a automated fast scan of all our machines for tcp port 109. This is easy to detect because there is only one source address. Source ports are incrementing.

Intent:
The intent is to find pop2 servers. If pop is not patched it may be possible to cause a buffer overflow and execute arbitrary instructions with root privileges.

Severity: A simple program running on one machine scanning for pop2 servers. Low severity.

```
17:35:33.251486 207.182.98.2.16413 > 129.242.4.207.109: S
2082977510:2082977510(0) win 512 <mss 1460>
17:35:33.251697 129.242.4.207.109 > 207.182.98.2.16413: R 0:0(0) ack 2082977511
win 0
17:35:33.315403 207.182.98.2.16414 > 129.242.4.246.109: S
3846620956:3846620956(0) win 512 <mss 1460>
17:35:33.317046 129.242.4.246.109 > 207.182.98.2.16414: R 0:11(11) ack
3846620957 win 0
17:35:33.419308 207.182.98.2.16416 > 129.242.4.34.109: S
4108381592:4108381592(0) win 512 <mss 1460>
17:35:33.419517 129.242.4.34.109 > 207.182.98.2.16416: R 0:0(0) ack 4108381593
win 0
17:35:44.067062 207.182.98.2.16443 > 129.242.6.106.109: S
2897104215:2897104215(0) win 512 <mss 1460>
17:35:44.067155 129.242.6.106.109 > 207.182.98.2.16443: R 0:0(0) ack 2897104216
win 0
17:35:44.113615 207.182.98.2.16444 > 129.242.6.107.109: S 234548107:234548107(0)
win 512 <mss 1460>
17:35:44.113725 129.242.6.107.109 > 207.182.98.2.16444: R 0:0(0) ack 234548108
win 0
```

```
17:35:44.168085 207.182.98.2.16446 > 129.242.6.108.109: S
2260514480:2260514480(0) win 512 <mss 1460>
17:35:44.168238 129.242.6.108.109 > 207.182.98.2.16446: R 0:0(0) ack 2260514481
win 0
```

Detect 3:

Background:
Here we have a Cisco router wich denies access to all incoming SNMP traffic.

Description: Udp port scan.

Technique:
This seems to be a automated not very fast scan for udp port 161. It goes on for several days and starts at the same time each day.

Intent:
The source machine Sys. Admin. actually replied to an enquiry and reported that it was a misconfigured HP Web JetAdmin searching for printers.

Severity: None.

```
Apr  6 01:05:39 hyper-gw.Uit.No 54047: Apr  6 01:05:38.863: %SEC-6-IPACCESSLOGP:
list 111 denied udp 129.a.b.29(4354) (ATM0/1/0.1 VC 26) -> 129.242.240.142(161),
1 packet
Apr  6 01:06:10 hyper-gw.Uit.No 54049: Apr  6 01:06:09.939: %SEC-6-IPACCESSLOGP:
list 111 denied udp 129.a.b.29(3438) (ATM0/1/0.1 VC 26) -> 129.242.210.25(161),
1 packet
Apr  6 01:06:28 hyper-gw.Uit.No 54050: Apr  6 01:06:27.987: %SEC-6-IPACCESSLOGP:
list 111 denied udp 129.a.b.29(4531) (ATM0/1/0.1 VC 26) -> 129.242.240.141(161),
1 packet
Apr  6 01:06:38 hyper-gw.Uit.No 54051: Apr  6 01:06:37.543: %SEC-6-IPACCESSLOGP:
list 111 denied udp 129.a.b.29(1367) (ATM0/1/0.1 VC 26) -> 129.242.217.92(161),
1 packet
Apr  7 01:05:40 hyper-gw.Uit.No 55941: Apr  7 01:05:39.085: %SEC-6-IPACCESSLOGP:
list 111 denied udp 129.a.b.29(2306) (ATM0/1/0.1 VC 26) -> 129.242.240.142(161),
1 packet
Apr  7 01:06:18 hyper-gw.Uit.No 55943: Apr  7 01:06:17.105: %SEC-6-IPACCESSLOGP:
list 111 denied udp 129.a.b.29(3880) (ATM0/1/0.1 VC 26) -> 129.242.210.25(161),
1 packet
Apr  7 01:06:36 hyper-gw.Uit.No 55945: Apr  7 01:06:35.193: %SEC-6-IPACCESSLOGP:
list 111 denied udp 129.a.b.29(2361) (ATM0/1/0.1 VC 26) -> 129.242.240.141(161),
1 packet
Apr  7 01:06:45 hyper-gw.Uit.No 55946: Apr  7 01:06:44.729: %SEC-6-IPACCESSLOGP:
list 111 denied udp 129.a.b.29(2216) (ATM0/1/0.1 VC 26) -> 129.242.217.92(161),
1 packet
```

Detect 4.
Background:
Detect taken from GIAC web 13. April. http://www.sans.org/y2k/041300.htm

Description:
Snort has detected a scan for UDB port 53 on several machines in the subnet a.b.e.*.

Technique:
This seems to be a automated scan for udp port 53. The attacker uses random source port from the same source machine. Scan is done in the middle of the night. The attacker uses 13 second to scan 18 machines.

Intent:
This is not a random portscan the attacker knows what he is looking for. There are multiple vulnerabilities in BIND and the attacker is looking for one or more machines he can compromise.

It is well known that unpatched versions of BIND is vulnerable for attacks but if BIND is up to date the severity of this is low.

```
Apr 12 03:44:34 hosth snort[87556]: spp_portscan:
PORTSCAN DETECTED from 159.148.165.250
Apr 12 03:44:41 hosth snort[87556]: spp_portscan:
portscan status from
159.148.165.250: 11 connections across 11 hosts:
TCP(0), UDP(11)
Apr 12 03:44:48 hosth snort[87556]: spp_portscan:
portscan status from
159.148.165.250: 7 connections across 7 hosts:
TCP(0), UDP(7)
Apr 12 03:44:54 hosth snort[87556]: spp_portscan:
End of portscan from
159.148.165.250
--------
Apr 12 03:44:29 159.148.165.250:4369 -> a.b.e.79:53 UDP
Apr 12 03:44:34 159.148.165.250:1151 -> a.b.e.101:53 UDP
Apr 12 03:44:34 159.148.165.250:3035 -> a.b.e.52:53 UDP
Apr 12 03:44:34 159.148.165.250:3504 -> a.b.e.63:53 UDP
Apr 12 03:44:34 159.148.165.250:3526 -> a.b.e.68:53 UDP
Apr 12 03:44:37 159.148.165.250:4950 -> a.b.e.91:53 UDP
Apr 12 03:44:37 159.148.165.250:4962 -> a.b.e.88:53 UDP
Apr 12 03:44:37 159.148.165.250:3279 -> a.b.e.118:53 UDP
Apr 12 03:44:38 159.148.165.250:4180 -> a.b.e.128:53 UDP
Apr 12 03:44:39 159.148.165.250:4626 -> a.b.e.135:53 UDP
```

Detect 5.
Background:
Taken from giac web 12 April. http://www.sans.org/y2k/041200.htm

Description:
A trinoo master server trying to install bcast deamons.

Technique:
One ip-address is performing a fast and automated scan to several ip-adresses on their tcp port 1524.

Intent:
To check if the systems are compromised by trinoo and if so install bcast deamons which can be used for
later DDOS attacks.

Severity: High.

```
Apr 9 00:54:05 hostp portsentry[522]: attackalert: Connect
from host: 195.145.171.21/195.145.171.21 to TCP port: 1524
Apr 9 00:54:05 hostp portsentry[522]: attackalert: Connect
from host: 195.145.171.21/195.145.171.21 to TCP port: 1524
Apr 9 00:54:05 hostr portsentry[418]: attackalert: Connect
from host: 195.145.171.21/195.145.171.21 to TCP port: 1524
Apr 9 00:54:05 hostb portsentry[334]: attackalert: Connect
from host: 195.145.171.21/195.145.171.21 to TCP port: 1524
Apr 9 00:54:49 hostc portsentry[15996]: attackalert: Connect
from host: 195.145.171.21/195.145.171.21 to TCP port: 1524
Apr 9 00:57:59 hostd portsentry[416]: attackalert: Connect
from host: 195.145.171.21/195.145.171.21 to TCP port: 1524
Apr 9 01:19:11 dns1 portsentry[438328]: attackalert: Connect
from host: 195.145.171.21/195.145.171.21 to TCP port: 1524
```

Detect 6.
Background:
Detect from giac web 12 April. http://www.sans.org/y2k/041200.htm

Description:
Snort has detected a scan for TCP port 80 on several machines in the subnet a.b.e.*, a.b.f and a.b.c.

Technique:
One source IP is sending a SYN flag to port 80 on 5 different machines in 2 seconds.

Intent:
Because the fast rate of connections this is active targeting and not regular http requests.
It could be the sscan tool in its second fase. In fase one it found the webservers in a.b.*.* and now in the second fase it tries identify potentioal vulnerabilities.

Severity:
This is a medium severity attack and the probed systems needs to be inspected for exploits.

```
Apr 9 00:42:47 hosth snort[87556]: spp_portscan:
PORTSCAN DETECTED from 205.215.135.210
Apr 9 00:42:54 hosth snort[87556]: spp_portscan: portscan status
from 205.215.135.210: 5 connections across 5 hosts: TCP(5), UDP(0)
Apr 9 00:43:00 hosth snort[87556]: spp_portscan: End of portscan
from 205.215.135.210
--------
Apr 9 00:42:46 205.215.135.210:62908 -> a.b.e.79:80 SYN **S*****
Apr 9 00:42:46 205.215.135.210:62909 -> a.b.e.91:80 SYN **S*****
Apr 9 00:42:46 205.215.135.210:62910 -> a.b.e.101:80 SYN **S*****
Apr 9 00:42:47 205.215.135.210:62941 -> a.b.f.79:80 SYN **S*****
Apr 9 00:42:47 205.215.135.210:62882 -> a.b.c.32:80 SYN **S****
```

Detect 7.
Background:
Detect from giac web 11 April. http://www.sans.org/y2k/041100.htm

Description:
A firewall detecting probes for udp ports 137, 22, 5632 and tcp ports 20043, 1243, 111, 8080, 27374 on target 24.3.21.199

Technique:
We have two attempts of windows name resolution from different ip adresses. We have two attempts of RPC to port 111
from two differnt ip-adresses. We have two attempts (different ip) to connect to port 1243 a known trojan (SubSeven).
We have three attempts (two src ip-adresses) to connect to port 8080 where Wingate may be running.
We have one attempt to connect to port 20034 which is used by the trojan Netbus.
Last we have several attempts to connect to udp port 5632 and 22 used by PCanywhere.

Intent:
The intent is to exploit trojans.

Severity: Low.

Apr 9 03:02:51 cc1014244-a kernel: securityalert: udp if=ef0 from 24.4.225.13:137 to 24.3.21.199 on unserved port 137
Apr 9 03:52:47 cc1014244-a kernel: securityalert: tcp if=ef0 from 129.71.227.212:1392 to 24.3.21.199 on unserved port 20034
Apr 9 03:53:05 cc1014244-a kernel: securityalert: tcp if=ef0 from 129.71.227.212:1646 to 24.3.21.199 on unserved port 1243
Apr 9 05:43:03 cc1014244-a kernel: securityalert: tcp if=ef0 from 208.232.120.196:623 to 24.3.21.199 on unserved port 111
Apr 9 05:48:02 cc1014244-a kernel: securityalert: udp if=ef0 from 24.3.21.225:3592 to 24.3.21.199 on unserved port 5632
Apr 9 05:48:02 cc1014244-a kernel: securityalert: udp if=ef0 from 24.3.21.225:3592 to 24.3.21.199 on unserved port 22
Apr 9 05:49:33 cc1014244-a kernel: securityalert: udp if=ef0 from 24.3.21.225:3595 to 24.3.21.199 on unserved port 5632
Apr 9 05:49:33 cc1014244-a kernel: securityalert: udp if=ef0 from 24.3.21.225:3595 to 24.3.21.199 on unserved port 22
Apr 9 06:06:43 cc1014244-a kernel: securityalert: udp if=ef0 from 24.3.21.225:1081 to 24.3.21.199 on unserved port 5632
Apr 9 06:06:43 cc1014244-a kernel: securityalert: udp if=ef0 from 24.3.21.225:1081 to 24.3.21.199 on unserved port 22
Apr 9 06:24:31 cc1014244-a kernel: securityalert: tcp if=ef0 from 209.44.129.184:56136 to 24.3.21.199 on unserved port 8080
Apr 9 06:24:31 cc1014244-a kernel: securityalert: tcp if=ef0 from 209.44.129.184:56148 to 24.3.21.199 on unserved port 8080
Apr 9 11:55:31 cc1014244-a kernel: securityalert: udp if=ef0 from 24.3.21.225:1848 to 24.3.21.199 on unserved port 22
Apr 9 12:48:16 cc1014244-a kernel: securityalert: udp if=ef0 from 24.3.21.225:1949 to 24.3.21.199 on unserved port 22
Apr 9 14:30:39 cc1014244-a kernel: securityalert: tcp if=ef0 from 209.67.70.56:828 to 24.3.21.199 on unserved port 111
Apr 9 17:54:03 cc1014244-a kernel: securityalert: udp if=ef0 from 24.3.21.225:1140 to 24.3.21.199 on unserved port 22
Apr 9 18:16:52 cc1014244-a kernel: securityalert: udp if=ef0 from 24.16.166.96:137 to 24.3.21.199 on unserved port 137
Apr 9 18:38:29 cc1014244-a kernel: securityalert: tcp if=ef0 from 24.17.11.6:4481 to 24.3.21.199 on unserved port 27374
Apr 9 18:51:46 cc1014244-a kernel: securityalert: udp if=ef0 from

```
24.3.21.225:2068 to 24.3.21.199 on unserved port 22
Apr 9 18:55:42 cc1014244-a kernel: securityalert: udp if=ef0 from
24.3.21.225:2235 to 24.3.21.199 on unserved port 22
Apr 9 19:06:08 cc1014244-a kernel: securityalert: udp if=ef0 from
24.3.21.225:2662 to 24.3.21.199 on unserved port 22
Apr 9 19:08:15 cc1014244-a kernel: securityalert: udp if=ef0 from
24.3.21.225:2757 to 24.3.21.199 on unserved port 22
Apr 9 19:10:11 cc1014244-a kernel: securityalert: udp if=ef0 from
24.3.21.225:2842 to 24.3.21.199 on unserved port 5632
Apr 9 19:10:11 cc1014244-a kernel: securityalert: udp if=ef0 from
24.3.21.225:2842 to 24.3.21.199 on unserved port 22
Apr 9 19:10:42 cc1014244-a kernel: securityalert: udp if=ef0 from
24.3.21.225:2864 to 24.3.21.199 on unserved port 5632
Apr 9 19:10:42 cc1014244-a kernel: securityalert: udp if=ef0 from
24.3.21.225:2864 to 24.3.21.199 on unserved port 22
Apr 9 19:12:49 cc1014244-a kernel: securityalert: udp if=ef0 from
24.3.21.225:2953 to 24.3.21.199 on unserved port 22
Apr 9 19:13:37 cc1014244-a kernel: securityalert: udp if=ef0 from
24.3.21.225:2989 to 24.3.21.199 on unserved port 5632
Apr 9 19:13:37 cc1014244-a kernel: securityalert: udp if=ef0 from
24.3.21.225:2989 to 24.3.21.199 on unserved port 22
Apr 9 19:15:41 cc1014244-a kernel: securityalert: udp if=ef0 from
24.3.21.225:3077 to 24.3.21.199 on unserved port 22
Apr 9 19:16:13 cc1014244-a kernel: securityalert: udp if=ef0 from
24.3.21.225:3101 to 24.3.21.199 on unserved port 5632
Apr 9 19:16:13 cc1014244-a kernel: securityalert: udp if=ef0 from
24.3.21.225:3101 to 24.3.21.199 on unserved port 22
Apr 9 19:34:20 cc1014244-a kernel: securityalert: tcp if=ef0 from
24.14.174.4:4409 to 24.3.21.199 on unserved port 8080
Apr 9 21:00:43 cc1014244-a kernel: securityalert: udp if=ef0 from
24.3.21.225:3713 to 24.3.21.199 on unserved port 22
Apr 9 21:41:33 cc1014244-a kernel: securityalert: tcp if=ef0 from
24.200.22.104:1936 to 24.3.21.199 on unserved port 1243
Apr 9 22:26:26 cc1014244-a kernel: securityalert: udp if=ef0 from
24.64.52.247:137 to 24.3.21.199 on unserved port 137
```

Detect 8.

Background:
Shadow reported 48 connections from 207.244.122.87 in hour 16.

Description: SYN scan for port 53.

Technique:
This is a fast and automated SYN scan for port 53. As soon as the three-way handshake completes the scanner initiates a normal FIN termination of the connection.
Source IP is always the same.

Intent:
The scanner is looking for primary nameservers from wich he can do a zone transfer.

Severity: When using the latest version of bind and it is properly configured the severity is low.

```
16:39:35.690592 207.244.122.87.2467 > 129.242.4.200.53: S
1207776198:1207776198(0) win 32120 <mss 1460,sackOK,timestamp 47716193[|tcp]>
(DF) (ttl 36, id 338)
16:39:35.692030 129.242.4.200.53 > 207.244.122.87.2467: S
1386766848:1386766848(0) ack 1207776199 win 60816 <mss 1460,nop,wscale
0,nop,nop,timestamp[|tcp]> (DF) (ttl 60, id 53534)
16:39:35.854763 207.244.122.87.2467 > 129.242.4.200.53: . ack 1 win 32120
<nop,nop,timestamp 47716211 27250100> (DF) (ttl 36, id 894)
16:39:35.866546 207.244.122.87.2467 > 129.242.4.200.53: F 1:1(0) ack 1 win 32120
<nop,nop,timestamp 47716211 27250100> (DF) (ttl 36, id 897)
16:39:35.866558 129.242.4.200.53 > 207.244.122.87.2467: . ack 2 win 60816
<nop,nop,timestamp 27250101 47716211> (DF) (ttl 60, id 53647)
16:39:35.869981 129.242.4.200.53 > 207.244.122.87.2467: F 1:1(0) ack 2 win 60816
<nop,nop,timestamp 27250101 47716211> (DF) (ttl 60, id 53655)
16:39:36.034695 207.244.122.87.2467 > 129.242.4.200.53: . ack 2 win 32120
<nop,nop,timestamp 47716228 27250101> (DF) (ttl 36, id 921)

16:39:35.690586 207.244.122.87.2480 > 129.242.4.206.53: S
1211445565:1211445565(0) win 32120 <mss 1460,sackOK,timestamp 47716193[|tcp]>
(DF) (ttl 36, id 351)
16:39:35.691766 129.242.4.206.53 > 207.244.122.87.2480: S
1386691072:1386691072(0) ack 1211445566 win 60816 <mss 1460,nop,wscale
0,nop,nop,timestamp[|tcp]> (DF) (ttl 60, id 53533)
16:39:35.853973 207.244.122.87.2480 > 129.242.4.206.53: . ack 1 win 32120
<nop,nop,timestamp 47716211 27250100> (DF) (ttl 36, id 893)
16:39:35.874563 207.244.122.87.2480 > 129.242.4.206.53: F 1:1(0) ack 1 win 32120
<nop,nop,timestamp 47716212 27250100> (DF) (ttl 36, id 902)
16:39:35.874569 129.242.4.206.53 > 207.244.122.87.2480: . ack 2 win 60816
<nop,nop,timestamp 27250101 47716212> (DF) (ttl 60, id 53662)
16:39:35.875823 129.242.4.206.53 > 207.244.122.87.2480: F 1:1(0) ack 2 win 60816
<nop,nop,timestamp 27250101 47716212> (DF) (ttl 60, id 53665)
16:39:36.038952 207.244.122.87.2480 > 129.242.4.206.53: . ack 2 win 32120
<nop,nop,timestamp 47716229 27250101> (DF) (ttl 36, id 922)

16:39:36.704900 207.244.122.87.3332 > 129.242.6.119.53: S
1217240269:1217240269(0) win 32120 <mss 1460,sackOK,timestamp 47716294[|tcp]>
(DF) (ttl 37, id 1740)
16:39:36.706427 207.244.122.87.3330 > 129.242.6.117.53: S
1208559027:1208559027(0) win 32120 <mss 1460,sackOK,timestamp 47716294[|tcp]>
(DF) (ttl 37, id 1738)
16:39:36.710521 207.244.122.87.3331 > 129.242.6.118.53: S
1202879989:1202879989(0) win 32120 <mss 1460,sackOK,timestamp 47716294[|tcp]>
(DF) (ttl 37, id 1739)

16:39:39.693428 207.244.122.87.3116 > 129.242.6.11.53: S
1204703141:1204703141(0) win 32120 <mss 1460,sackOK,timestamp 47716593[|tcp]>
(DF) (ttl 37, id 4219)
16:39:39.695332 129.242.6.11.53 > 207.244.122.87.3116: R 0:11(11) ack 1204703142
win 0 (DF) (ttl 64, id 41985)

16:39:40.656480 207.244.122.87.3401 > 129.242.6.185.53: S
1216906238:1216906238(0) win 32120 <mss 1460,sackOK,timestamp 47716690[|tcp]>
(DF) (ttl 36, id 5204)
16:39:40.656692 129.242.6.185.53 > 207.244.122.87.3401: R 0:0(0) ack 1216906239
win 0 (ttl 64, id 263)
```

Detect 9.
Background:
Shadow reported 7 connections from 207.244.122.87 in hour 22.

Description:
The attacker from detect 8 is coming back 6 hours later probing for the version of BIND for the machines that answered on the port scan.
Source IP is always the same and is also the same as for previous port scan.

Technique:
Fast automated probe for the version of BIND.

Intent: Find the version of BIND to exploit security holes. There may be no known security holes in the current version of BIND, but what about next month?

Severity: Low.

```
22:43:27.593170 207.244.122.87.1197 > 129.242.4.200.53: 6+ (30) (ttl 36, id
1875)
22:43:27.594122 129.242.4.200.53 > 207.244.122.87.1197: 6* q: version.bind.
1/0/0 . (63) (ttl 60, id 49142)
22:43:27.602780 207.244.122.87.1198 > 129.242.4.206.53: 6+ (30) (ttl 36, id
1876)
22:43:27.604129 129.242.4.206.53 > 207.244.122.87.1198: 6* q: version.bind.
1/0/0 . (63) (ttl 60, id 49145)
```

Detect 10:
Background:
Shadow reported 69 connections from 208.3.221.245 in hour 17.

Description:
Fast and automated scan for tcp port 1080.

Technique:
The source ip is sending four SYN packets at each target. For each target the sequence number is the same.

Intent:
It could be someone looking for wingate http://www.cert.org/vul_notes/VN-98.03.WinGate.html or
MERCUR WebView WebMail-Client 1.0 http://www.ussrback.com/labs36.html.

Severity: Low.

```
17:09:41.149177 208.3.221.245.4492 > 129.242.4.44.1080: S 28008656:28008656(0)
win 8192 <mss 536,nop,wscale 0,nop,nop,timestamp[|tcp]> (DF) [tos 0xc8] (ttl
104, id 1436)
17:09:41.149604 129.242.4.44.1080 > 208.3.221.245.4492: R 0:11(11) ack 28008657
win 0 (DF) [tos 0xc8] (ttl 64, id 7154)
17:09:42.076712 208.3.221.245.4492 > 129.242.4.44.1080: S 28008656:28008656(0)
win 8192 <mss 536,nop,wscale 0,nop,nop,timestamp[|tcp]> (DF) [tos 0xc8] (ttl
104, id 13212)
17:09:42.077095 129.242.4.44.1080 > 208.3.221.245.4492: R 0:11(11) ack 1 win 0
(DF) [tos 0xc8] (ttl 64, id 15589)
```

```
17:09:42.892611 208.3.221.245.4492 > 129.242.4.44.1080: S 28008656:28008656(0)
win 8192 <mss 536,nop,wscale 0,nop,nop,timestamp[|tcp]> (DF) [tos 0xc8] (ttl
104, id 17820)
17:09:42.892998 129.242.4.44.1080 > 208.3.221.245.4492: R 0:11(11) ack 1 win 0
(DF) [tos 0xc8] (ttl 64, id 18961)
17:09:43.753243 208.3.221.245.4492 > 129.242.4.44.1080: S 28008656:28008656(0)
win 8192 <mss 536,nop,wscale 0,nop,nop,timestamp[|tcp]> (DF) [tos 0xc8] (ttl
104, id 19100)
17:09:43.753566 129.242.4.44.1080 > 208.3.221.245.4492: R 0:11(11) ack 1 win 0
(DF) [tos 0xc8] (ttl 64, id 44229)

17:12:57.208467 208.3.221.245.1350 > 129.242.6.68.1080: S 28204733:28204733(0)
win 8192 <mss 536,nop,wscale 0,nop,nop,timestamp[|tcp]> (DF) [tos 0x48] (ttl
104, id 35754)
17:12:57.208955 129.242.6.68.1080 > 208.3.221.245.1350: R 0:11(11) ack 28204734
win 0 (DF) [tos 0x48] (ttl 64, id 9682)
17:12:58.063982 208.3.221.245.1350 > 129.242.6.68.1080: S 28204733:28204733(0)
win 8192 <mss 536,nop,wscale 0,nop,nop,timestamp[|tcp]> (DF) [tos 0x48] (ttl
104, id 40618)
17:12:58.063993 129.242.6.68.1080 > 208.3.221.245.1350: R 0:11(11) ack 1 win 0
(DF) [tos 0x48] (ttl 64, id 1025)
17:12:58.965777 208.3.221.245.1350 > 129.242.6.68.1080: S 28204733:28204733(0)
win 8192 <mss 536,nop,wscale 0,nop,nop,timestamp[|tcp]> (DF) [tos 0x48] (ttl
104, id 44202)
17:12:58.966229 129.242.6.68.1080 > 208.3.221.245.1350: R 0:11(11) ack 1 win 0
(DF) [tos 0x48] (ttl 64, id 57989)
17:12:59.799732 208.3.221.245.1350 > 129.242.6.68.1080: S 28204733:28204733(0)
win 8192 <mss 536,nop,wscale 0,nop,nop,timestamp[|tcp]> (DF) [tos 0x48] (ttl
104, id 45738)
17:12:59.800186 129.242.6.68.1080 > 208.3.221.245.1350: R 0:11(11) ack 1 win 0
(DF) [tos 0x48] (ttl 64, id 61221)
```

Last Updated: July 25th, 2017

Upcoming Training

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