Global Information Assurance Certification Paper

*** Northcutt, Oh my! Jonathan is clearly one smart cookie! I saw this and got up and poured a cup of coffee before starting. The deductions primarily are for clarity, it took a bit of work sometimes to read and comprehend the analysis, a few more words wouldn't hurt! Accuracy fine, analysis process is solid, the evidence of the research is there. It is our privilege to confer certification. 90 *

Ten Detects for SANS Intrusion Analysis Certification

Jonathan Good

These detects are from tcpdump captures (run through snort for analysis) and firewall logs (run through custom perl scripts for analysis), local information has been sanitized to x.y.z for an external class C subnet, or abc for an external domain.

Detect # 1

Initial Detect: This detect is from a snort analysis of a tcpdump capture

```
[**] Possible Zone Transfer [**]
03/30-15:09:37.445455 61.12.23.84:2884 -> x.y.z.22:53
TCP TTL:108 TOS:0x0 ID:44096  DF
**S***** Seq: 0xA5685AD2   Ack: 0x0   Win: 0x2238
TCP Options => MSS: 1460 NOP NOP SackOK

[**] Possible Zone Transfer [**]
03/30-15:09:38.694867 61.12.23.84:2885 -> x.y.z.22:53
TCP TTL:108 TOS:0x0 ID:44100  DF
**S***** Seq: 0xA56E1F9B   Ack: 0x0   Win: 0x2238
TCP Options => MSS: 1460 NOP NOP SackOK

[**] Possible Zone Transfer [**]
03/30-15:09:39.942596 61.12.23.84:2886 -> x.y.z.22:53
TCP TTL:108 TOS:0x0 ID:44105  DF
**S***** Seq: 0xA573873B   Ack: 0x0   Win: 0x2238
TCP Options => MSS: 1460 NOP NOP SackOK
```

Analysis: This was flagged due to TCP use for DNS, unusual at our site as we do not have long domain names.

There is active targeting because it targets a specific service on a machine providing that service.

I checked full logs to see if there was a UDP query which had been truncated, preceding this connection and found:

```
15:09:36.838956 async83-syd-isp-245.nas.one.net.au.2883 > x.y.z.domain: 119 op5 [2a] SOA? abc.com. (84) (ttl 108, id 44095)
15:09:36.840787 x.y.z.domain > async83-syd-isp-245.nas.one.net.au.2883: 119 op5 Refused q: abc.com. 2/0/0 WIN2000.abc.com.
(Class 254) CNAME WIN2000.abc.com., WIN2000.abc.com. A async83-syd-isp-245.nas.one.net.au (84) (DF) (ttl 255, id 40753)
```

This looks like an attempt to poison DNS with a CNAME pointing to the originator, followed by a zone transfer. I was not sure of the format of the Refused response, so I took a trace of a request for the same WIN2000.abc.com and the response was nothing like the above, so something is definitely fishy about that packet.

Source was an Australian ISP, OneNet:

Note: *Severity is based on initial detect only, as it would be used to determine how much analysis to devote to the detect.*

Severity		(Criticality		Lethality)		(System Countermeasures		Network Countermeasures)
Low	=	DNS Server	+	DNS Zone information	-	Split DNS, latest Rev. OS, Bind	+	Restrictive Firewall
0		5		3		4		4

Severity after analysis: 1 (Lethality increase due to poison DNS attempt)

Detect # 2

Initial Detect: This detect is from a snort analysis of a tcpdump capture

```
[**] Possible Zone Transfer [**]
03/30-13:08:06.894368 207.46.12.11:2200 -> x.y.z.22:53
TCP TTL:241 TOS:0x0 ID:16586
**S***** Seq: 0x34D26A51   Ack: 0x0   Win: 0x800

[**] Possible Zone Transfer [**]
03/30-13:08:06.894584 207.46.12.11:2201 -> x.y.z.22:53
TCP TTL:241 TOS:0x0 ID:31181
**S***** Seq: 0x45CEC9AF   Ack: 0x0   Win: 0x800

[**] Possible Zone Transfer [**]
03/30-13:08:06.894761 207.46.12.11:2202 -> x.y.z.22:53
TCP TTL:241 TOS:0x0 ID:48778
**S***** Seq: 0x2F469A16   Ack: 0x0   Win: 0x800

[**] Possible Zone Transfer [**]
03/30-13:08:06.895068 x.y.z.22:53 -> 207.46.12.11:2200
TCP TTL:255 TOS:0x0 ID:21059  DF
**S***A* Seq: 0x50CE85A0   Ack: 0x34D26A52   Win: 0x2398
TCP Options => MSS: 536

[**] Possible Zone Transfer [**]
03/30-13:08:06.895227 x.y.z.22:53 -> 207.46.12.11:2201
TCP TTL:255 TOS:0x0 ID:21060  DF
**S***A* Seq: 0x50D05778   Ack: 0x45CEC9B0   Win: 0x2398
TCP Options => MSS: 536

[**] Possible Zone Transfer [**]
03/30-13:08:06.895391 x.y.z.22:53 -> 207.46.12.11:2202
TCP TTL:255 TOS:0x0 ID:21061  DF
**S***A* Seq: 0x50D109F4   Ack: 0x2F469A17   Win: 0x2398
TCP Options => MSS: 536

[**] Possible Zone Transfer [**]
03/30-13:08:07.009305 207.46.12.11:2200 -> x.y.z.22:53
TCP TTL:50 TOS:0x0 ID:60505
****R*** Seq: 0x34D26A52   Ack: 0x0   Win: 0x0

[**] Possible Zone Transfer [**]
03/30-13:08:07.009364 207.46.12.11:2201 -> x.y.z.22:53
TCP TTL:50 TOS:0x0 ID:60507
****R*** Seq: 0x45CEC9B0   Ack: 0x0   Win: 0x0

[**] Possible Zone Transfer [**]
03/30-13:08:07.009414 207.46.12.11:2202 -> x.y.z.22:53
TCP TTL:50 TOS:0x0 ID:60509
****R*** Seq: 0x2F469A17   Ack: 0x0   Win: 0x0

[**] Possible Zone Transfer [**]
03/30-13:08:07.016093 207.46.12.11:2200 -> x.y.z.22:53
```

```
TCP TTL:241 TOS:0x0 ID:7975
****R*A* Seq: 0x34D26A52   Ack: 0x50CE85A1   Win: 0x800

[**] Possible Zone Transfer [**]
03/30-13:08:07.016449 207.46.12.11:2201 -> x.y.z.22:53
TCP TTL:241 TOS:0x0 ID:27722
****R*A* Seq: 0x45CEC9B0   Ack: 0x50D05779   Win: 0x800

[**] Possible Zone Transfer [**]
03/30-13:08:07.016624 207.46.12.11:2202 -> x.y.z.22:53
TCP TTL:241 TOS:0x0 ID:13768
****R*A* Seq: 0x2F469A17   Ack: 0x50D109F5   Win: 0x800
```

Analysis: This was flagged due to TCP use for DNS, unusual at our site as we do not have long domain names.
There is active targeting pointed at a name server.
These are spoofed SYN packets, due to the vast difference of the TTLs between the initial SYN and the RST packets. I received 234 of these spoofed packets in about a 12 hour period. 78 groups of 3, with the source ports of the form 2[01234]00, 2[01234]01, 2[01234]2
This could have been used to try and hide a real scan, but I did not see any other anomalous traffic that I could relate to this. It is more likely that we were being used as a screen to hide a scan of the spoofed machine.
Note: *Severity is based on initial detect only, as it would be used to determine how much analysis to devote to the detect.*

Severity		(Criticality		Lethality)		(System Countermeasures		Network Countermeasures)
	=	DNS Server	+	DNS Zone information	-	Split DNS, latest Rev. OS, Bind	+	Restrictive Firewall
0		5		3		4		4

Severity after analysis: -4 (Criticality and Lethality decrease due to non-targeting)

Detect # 3

Initial Detect: This detect is from a snort analysis of a tcpdump capture

```
[**] Unserved High Port [**]
03/30-15:33:54.243224 24.131.134.238:1025 -> x.y.z.22:44007
TCP TTL:116 TOS:0x0 ID:57955  DF
**S***** Seq: 0x5600A849   Ack: 0x0   Win: 0x4000
TCP Options => MSS: 1460 NOP NOP SackOK

[**] Unserved High Port [**]
03/30-15:34:13.192574 24.131.134.238:1025 -> x.y.z.22:44176
TCP TTL:116 TOS:0x0 ID:57965  DF
**S***** Seq: 0x5649A196   Ack: 0x0   Win: 0x4000
TCP Options => MSS: 1460 NOP NOP SackOK

[**] Unserved High Port [**]
03/30-15:34:38.453246 24.131.134.238:1025 -> x.y.z.22:44482
TCP TTL:116 TOS:0x0 ID:57996  DF
**S***** Seq: 0x56AA53A5   Ack: 0x0   Win: 0x4000
TCP Options => MSS: 1460 NOP NOP SackOK

[**] Unserved High Port [**]
03/30-15:34:50.195964 24.131.134.238:1025 -> x.y.z.22:44626
TCP TTL:116 TOS:0x0 ID:58006  DF
**S***** Seq: 0x56D84B9A   Ack: 0x0   Win: 0x4000
TCP Options => MSS: 1460 NOP NOP SackOK
```

```
[**] Unserved High Port [**]
03/30-15:34:56.004384 24.131.134.238:1025 -> x.y.z.22:44697
TCP TTL:116 TOS:0x0 ID:58013  DF
**S***** Seq: 0x56EF370C   Ack: 0x0   Win: 0x4000
TCP Options => MSS: 1460 NOP NOP SackOK

[**] Unserved High Port [**]
03/30-15:35:04.064752 24.131.134.238:1025 -> x.y.z.22:44815
TCP TTL:116 TOS:0x0 ID:58020  DF
**S***** Seq: 0x570E9905   Ack: 0x0   Win: 0x4000
TCP Options => MSS: 1460 NOP NOP SackOK

[**] Unserved High Port [**]
03/30-15:36:23.496008 24.131.134.238:1025 -> x.y.z.22:45595
TCP TTL:116 TOS:0x0 ID:58047  DF
**S***** Seq: 0x583DFF67   Ack: 0x0   Win: 0x4000
TCP Options => MSS: 1460 NOP NOP SackOK

[**] Unserved High Port [**]
03/30-15:36:29.672282 24.131.134.238:1025 -> x.y.z.22:45669
TCP TTL:116 TOS:0x0 ID:58056  DF
**S***** Seq: 0x585631A5   Ack: 0x0   Win: 0x4000
TCP Options => MSS: 1460 NOP NOP SackOK

[**] Unserved High Port [**]
03/30-17:55:51.204872 24.131.134.238:1025 -> x.y.z.22:52605
TCP TTL:116 TOS:0x0 ID:63821  DF
**S***** Seq: 0xD5CDD1ED   Ack: 0x0   Win: 0x4000
TCP Options => MSS: 1460 NOP NOP SackOK

[**] Unserved High Port [**]
03/30-17:57:28.023638 24.131.134.238:1025 -> x.y.z.22:53197
TCP TTL:116 TOS:0x0 ID:63926  DF
**S***** Seq: 0xD744A23F   Ack: 0x0   Win: 0x4000
TCP Options => MSS: 1460 NOP NOP SackOK

[**] Unserved High Port [**]
03/30-18:01:01.877738 24.131.134.238:1025 -> x.y.z.22:54127
TCP TTL:116 TOS:0x0 ID:65432  DF
**S***** Seq: 0xDA7AD961   Ack: 0x0   Win: 0x4000
TCP Options => MSS: 1460 NOP NOP SackOK
```

Analysis: These were flagged as TCP session initiations (SYN) on ports we don't offer services.
There is active targeting because several ports are being probed by one machine, on the same source port.
Technique here is fairly slow scanning, all from the same IP:port over a range of randomly increasing port numbers. Intent is reconnaissance, looking for open ports.
Source is Continental Cablevision, probably a cable modem.
Note: *Severity is based on initial detect only, as it would be used to determine how much analysis to devote to the detect.*

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

		Scan of Firewall		Recon		Latest patches		Restrictive Firewall
0		5		3		4		4

Severity after analysis: unchanged

Detect # 4

Initial Detect: This detect is from a snort analysis of a tcpdump capture

```
[**] MISC-Attempted Sun RPC high port access [**]
03/10-16:17:14.261308 204.71.200.245:80 -> x.y.z.22:32771
TCP TTL:52 TOS:0x0 ID:48234  DF
**S***A* Seq: 0x631C3104   Ack: 0x1B2E9FAE   Win: 0x4470
TCP Options => MSS: 1460

[**] MISC-Attempted Sun RPC high port access [**]
03/10-16:17:14.355288 204.71.200.245:80 -> x.y.z.22:32771
TCP TTL:52 TOS:0x0 ID:48398  DF
*****PA* Seq: 0x631C3105   Ack: 0x1B2EA0D9   Win: 0x4470

[**] MISC-Attempted Sun RPC high port access [**]
03/10-16:17:14.355667 204.71.200.245:80 -> x.y.z.22:32771
TCP TTL:52 TOS:0x0 ID:48399  DF
***F**A* Seq: 0x631C3118   Ack: 0x1B2EA0D9   Win: 0x4470

[**] MISC-Attempted Sun RPC high port access [**]
03/10-16:17:14.450766 204.71.200.245:80 -> x.y.z.22:32771
TCP TTL:52 TOS:0x0 ID:48574  DF
******A* Seq: 0x631C3119   Ack: 0x1B2EA0DA   Win: 0x4470
```

Analysis: This was flagged by the Snort rules as a port known to be used for Sun RPC.

This is a case of the firewall proxying on a port that might be used for a Sun RPC, not active targeting. The host is a real web server (www.yahoo.com), and further analysis of the traffic shows that the connections initiated from x.y.z.22:32771

Note: *Severity is based on initial detect only, as it would be used to determine how much analysis to devote to the detect.*

Severity		(Criticality		Lethality)		(System Countermeasures		Network Countermeasures)
	=	Firewall targeted	+	RPC access dangerous	-	Not running RPC, Up to date Firewall patches	+	Restrictive Firewall
0		5		4		5		4

Severity after analysis: NA, not a scan

Detect # 5 This detect is from a snort analysis of a tcpdump capture

Initial Detect:

```
[**] Syn-Fin flags [**]
04/05-10:48:37.680454 195.173.97.31:769 -> x.y.z.22:43849
TCP TTL:52 TOS:0x0 ID:46652  DF
**SFR*A* Seq: 0x0   Ack: 0x4500002C   Win: 0x4000

[**] Null Scan - no flags [**]
04/05-10:51:49.802735 195.173.97.31:1528 -> x.y.z.22:4091
TCP TTL:52 TOS:0x0 ID:47029
******** Seq: 0x20EEDE   Ack: 0x100018   Win: 0x5197

[**] Reserved flags [**]
```

```
04/05-10:54:37.443644 195.173.97.31:104 -> x.y.z.22:15650
TCP TTL:52 TOS:0x0 ID:47554
2***R*AU Seq: 0x38352220   Ack: 0x68656967   Win: 0x3D22
```

Analysis: This was flagged due to illegal combinations of TCP flag bits.

There isn't active targeting because even though it looks like a very interesting scan, it is more likely a 'Demon' internet packet. The source is a web server (shop.starshiptitanic.com), hosted by The Digital Village, on a class C subnet from Demon.net. Review of firewall logs show a user connected to the web site prior to the glitch. Demon.net claims to have faulty Ascend routers that mangle some traffic.

Note: *Severity is based on initial detect only, as it would be used to determine how much analysis to devote to the detect.*

Severity		(Criticality		Lethality)		(System Countermeasures		Network Countermeasures)
	=	Firewall Targeted	+	Recon	-	Up to date Firewall patches	+	Restrictive Firewall
0		5		3		4		4

Severity after analysis: NA - not an active target

Detect # 6

Initial Detect: This detect is from a snort analysis of a tcpdump capture

```
[**] MISC-WinGate-8080-Attempt [**]
04/05-10:47:10.543453 194.217.242.91:3773 -> x.y.z.22:8080
TCP TTL:52 TOS:0x0 ID:16143  DF
**S***** Seq: 0x4C601F   Ack: 0x0   Win: 0x2000
TCP Options => MSS: 1460
```

Analysis: This was flagged as an attempt to find a proxy server that could be used to redirect traffic, and hide the originator's identity.

The source address here is a mail server at demon.net (anchor-post-33.mail.demon.net). A review of traffic shows this packet occurred in the midst of a mail transfer session.

Looking at the traffic in more detail:

```
3434207970:3434209430(1460) ack 3311662681 win 17520 (DF) (ttl 52, id 16051)
10:47:10.360791 anchor-post-33.mail.demon.net.4739 > abc.smtp: . 3434209430:3434210890(1460) ack 3311662681 win 17520 (DF)
(ttl 52, id 16052)
10:47:10.361200 abc.smtp > anchor-post-33.mail.demon.net.4739: . ack 3434210890 win 8760 (DF) (ttl 255, id 49901)
10:47:10.389763 anchor-post-33.mail.demon.net.4739 > abc.smtp: . 3434210890:3434212350(1460) ack 3311662681 win 17520 (DF)
(ttl 52, id 16072)
10:47:10.397623 anchor-post-33.mail.demon.net.4739 > abc.smtp: . 3434212350:3434213810(1460) ack 3311662681 win 17520 (DF)
(ttl 52, id 16073)
10:47:10.398049 abc.smtp > anchor-post-33.mail.demon.net.4739: . ack 3434213810 win 8760 (DF) (ttl 255, id 49902)
10:47:10.511980 anchor-post-33.mail.demon.net.4739 > abc.smtp: . 3434213810:3434215270(1460) ack 3311662681 win 17520 (DF)
(ttl 52, id 16114)
10:47:10.519822 anchor-post-33.mail.demon.net.4739 > abc.smtp: . 3434215270:3434216730(1460) ack 3311662681 win 17520 (DF)
(ttl 52, id 16115)
10:47:10.529623 abc.smtp > anchor-post-33.mail.demon.net.4739: . ack 3434216730 win 8760 (DF) (ttl 255, id 49903)
10:47:10.535587 anchor-post-33.mail.demon.net.4739 > abc.smtp: . 3434216730:3434218190(1460) ack 3311662681 win 17520 (DF)
(ttl 52, id 16142)
10:47:10.543453 anchor-post-33.mail.demon.net.3773 > abc.8080: S 5005343:5006799(1456) win 8192 <mss 1460> (DF) (ttl 52, id
16143)
10:47:10.559132 anchor-post-33.mail.demon.net.4739 > abc.smtp: . 3434219650:3434221110(1460) ack 3311662681 win 17520 (DF)
(ttl 52, id 16155)
10:47:10.559514 abc.smtp > anchor-post-33.mail.demon.net.4739: . ack 3434218190 win 8760 (DF) (ttl 255, id 49904)
10:47:10.566984 anchor-post-33.mail.demon.net.4739 > abc.smtp: . 3434221110:3434222570(1460) ack 3311662681 win 17520 (DF)
(ttl 52, id 16156)
10:47:10.567330 abc.smtp > anchor-post-33.mail.demon.net.4739: . ack 3434218190 win 8760 (DF) (ttl 255, id 49905)
```

```
10:47:10.672144 anchor-post-33.mail.demon.net.4739 > abc.smtp: . 3434222570:3434224030(1460) ack 3311662681 win 17520 (DF)
(ttl 52, id 16216)
10:47:10.672516 abc.smtp > anchor-post-33.mail.demon.net.4739: . ack 3434218190 win 8760 (DF) (ttl 255, id 49906)
10:47:10.679984 anchor-post-33.mail.demon.net.4739 > abc.smtp: . 3434224030:3434225490(1460) ack 3311662681 win 17520 (DF)
(ttl 52, id 16217)
10:47:10.680314 abc.smtp > anchor-post-33.mail.demon.net.4739: . ack 3434218190 win 8760 (DF) (ttl 255, id 49907)
10:47:10.680314 abc.smtp > anchor-post-33.mail.demon.net.4739: . ack 3434218190 win 8760 (DF) (ttl 255, id 49907)
10:47:10.699150 anchor-post-33.mail.demon.net.4739 > abc.smtp: . 3434225490:3434226950(1460) ack 3311662681 win 17520 (DF)
(ttl 52, id 16227)
10:47:10.699507 abc.smtp > anchor-post-33.mail.demon.net.4739: . ack 3434218190 win 8760 (DF) (ttl 255, id 49908)
10:47:10.731564 anchor-post-33.mail.demon.net.4757 > abc.smtp: S 3437062831:3437062831(0) win 16384 <mss 1460> (DF) (ttl 52,
id 16246)
10:47:10.731967 abc.smtp > anchor-post-33.mail.demon.net.4757: S 3314832479:3314832479(0) ack 3437062832 win 8760 <mss 1460>
(DF) (ttl 255, id 49909)
10:47:10.819260 anchor-post-33.mail.demon.net.4739 > abc.smtp: . 3434218190:3434219650(1460) ack 3311662681 win 17520 (DF)
(ttl 52, id 16285)
10:47:10.819699 abc.smtp > anchor-post-33.mail.demon.net.4739: . ack 3434226950 win 8760 (DF) (ttl 255, id 49910)
10:47:10.862681 anchor-post-33.mail.demon.net.4757 > abc.smtp: . ack 3314832480 win 17520 (DF) (ttl 52, id 16310)
```

I see the smtp traffic coming in groups of two packets with sequential IP Ids, with ACK from our side. After the mangled 2nd packet, we ACK the 1st one only and things get out of sync, the other end keeps going, then resends the mangled packet and the ACKs catch up.
Given the surrounding traffic I am inclined to write this off as another mangled demon.net packet.

Note: *Severity is based on initial detect only, as it would be used to determine how much analysis to devote to the detect.*

Severity		(Criticality		Lethality)		(System Countermeasures		Network Countermeasures)
	=	Web/Proxy server	+	Could be used as intermediary	-	Not running proxy, Up to date Firewall patches	+	Restrictive Firewall
-1		4		4		5		4

Severity after analysis: NA - not an active target

Detect # 7

Initial Detect: This detect is from a firewall log analysis program for logs of 4/5/00

```
***IP_UNKNOWN*** 216.148.179.186
    6 - 216.148.179.186 / Count SPort  Destination IP:DPort Ptcl/Alert
                        :     2  3786   x.y.z.22: 1433  tcp/on unserved port
                        :     4  1275   x.y.z.22: 1521  tcp/on unserved port
-------------------------------------------------------------------------------
Apr  5 15:41:26 abc.com unix: securityalert: tcp if=qe0 from 216.148.179.186:1275 to x.y.z.22 on unserved port 1521
Apr  5 15:41:26 abc.com unix: securityalert: tcp if=qe0 from 216.148.179.186:1275 to x.y.z.22 on unserved port 1521
Apr  5 15:41:27 abc.com unix: securityalert: tcp if=qe0 from 216.148.179.186:1275 to x.y.z.22 on unserved port 1521
Apr  5 15:41:27 abc.com unix: securityalert: tcp if=qe0 from 216.148.179.186:1275 to x.y.z.22 on unserved port 1521
Apr  5 21:38:55 abc.com unix: securityalert: tcp if=qe0 from 216.148.179.186:3786 to x.y.z.22 on unserved port 1433
Apr  5 21:38:56 abc.com unix: securityalert: tcp if=qe0 from 216.148.179.186:3786 to x.y.z.22 on unserved port 1433
```

Analysis: This script summarizes all firewall securityalert messages by source IP.
This is active targeting.
Scanning two ports 1433 (MS SQL Server) and 1521 (Unassigned in RFC 1700, nCube license manager according to rapidnet port search)
The intent of the scan could be to find a vulnerable MS SQL server (1433), default configurations of which are known to have vulnerabilities. Port 1521 could be a search for a trojan compiled on that port.
Source looks to be hidden behind a firewall or router doing address translation (nat179-186.alldata.net).
This was the first contact from this location, but there have been additional hits to the same ports on 4/6-4/9.

Note: *Severity is based on initial detect only, as it would be used to determine how much analysis to devote to the detect.*

Severity		(Criticality		Lethality)		(System Countermeasures		Network Countermeasures)
	=	Firewall targeted	+	SQL server holes	-	Up to date Firewall patches, services blocked	+	Restrictive Firewall
0		5		4		5		4

Severity after analysis: -2 (Target is apparently a MS SQL server which does not exist, or a desktop trojan screened by the firewall)

Detect # 8

Initial Detect: From firewall logs

```
***IP_UNKNOWN*** 198.22.121.120
    8 - 198.22.121.120  / Count SPort  Destination IP:DPort Ptcl/Alert
                        :     4 34166   x.y.z.22: 5190  tcp/on unserved port
                        :     4 34936   x.y.z.22: 5190  tcp/on unserved port
-------------------------------------------------------------------------------
Apr  5 13:22:30 abc.com unix: securityalert: tcp if=qe0 from 198.22.121.120:34166 to x.y.z.22 on unserved port 5190
Apr  5 13:22:30 abc.com unix: securityalert: tcp if=qe0 from 198.22.121.120:34166 to x.y.z.22 on unserved port 5190
Apr  5 13:22:31 abc.com unix: securityalert: tcp if=qe0 from 198.22.121.120:34166 to x.y.z.22 on unserved port 5190
Apr  5 13:22:31 abc.com unix: securityalert: tcp if=qe0 from 198.22.121.120:34166 to x.y.z.22 on unserved port 5190
Apr  5 13:23:31 abc.com unix: securityalert: tcp if=qe0 from 198.22.121.120:34936 to x.y.z.22 on unserved port 5190
Apr  5 13:23:31 abc.com unix: securityalert: tcp if=qe0 from 198.22.121.120:34936 to x.y.z.22 on unserved port 5190
Apr  5 13:23:32 abc.com unix: securityalert: tcp if=qe0 from 198.22.121.120:34936 to x.y.z.22 on unserved port 5190
Apr  5 13:23:32 abc.com unix: securityalert: tcp if=qe0 from 198.22.121.120:34936 to x.y.z.22 on unserved port 5190
```

Analysis: Probing a port where we don't provide service.

This is active targeting.

This appears to be two tries to determine if AOL service (port 5190) is available from (or through) an external machine. Intent could be to exploit some vulnerability in the AOL client, there is a rumored buffer overflow in the AOL IM client which will also use this port by default.

The apparent source of the scan is helios.compusa.com; however, as the firewall just resets the connection there is no way to know for sure if they are the real source.

Note: *Severity is based on initial detect only, as it would be used to determine how much analysis to devote to the detect.*

Severity		(Criticality		Lethality)		(System Countermeasures		Network Countermeasures)
	=	Desktop running AOL or Firewall proxy	+	AOL exploits	-	Up to date Firewall patches, AOL service blocked	+	Restrictive Firewall
-2		3		4		5		4

Severity after analysis: no change

Detect # 9

Initial Detect:

```
[**] Reserved flags [**]
04/10-10:19:33.004352 216.35.123.108:80 -> x.y.z.22:32999
TCP TTL:49 TOS:0x0 ID:64441  DF
2*S***A* Seq: 0x0   Ack: 0x3EDEA181   Win: 0x5B4
TCP Options => MSS: 1460

[**] Reserved flags [**]
```

```
04/10-10:19:35.542560 216.35.123.108:80 -> x.y.z.22:33023
TCP TTL:49 TOS:0x0 ID:2579  DF
2*S***A* Seq: 0x0   Ack: 0x3F14CEAE   Win: 0x5B4
TCP Options => MSS: 1460

[**] Reserved flags [**]
04/10-10:21:50.793899 216.35.123.108:80 -> x.y.z.22:34262
TCP TTL:49 TOS:0x0 ID:11949  DF
2*S***A* Seq: 0x0   Ack: 0x49C49D5E   Win: 0x5B4
TCP Options => MSS: 1460

[**] Reserved flags [**]
04/10-10:25:09.544969 216.35.123.108:80 -> x.y.z.22:35759
TCP TTL:49 TOS:0x0 ID:26779  DF
2*S***A* Seq: 0x0   Ack: 0x56C10193   Win: 0x5B4
TCP Options => MSS: 1460

[**] Reserved flags [**]
04/10-10:29:42.379264 216.35.123.108:80 -> x.y.z.22:37356
TCP TTL:49 TOS:0x0 ID:30928  DF
2*S***A* Seq: 0x0   Ack: 0x655AC671   Win: 0x5B4
TCP Options => MSS: 1460
```

Analysis: Random port scan using invalid TCP flags.

This looks like a scan using invalid TCP flags; however, detailed analysis of the captured tcp traffic shows that we initiate a connection to the web server, it responds with this messed up Syn/Ack, but the proxy treats it as a normal Syn/Ack and Acks it. Then the session continues normally, with data and a Fin closure.

It is possible that this web server has been modified to send the illegal bit to try to get information on connecting hosts, based on how they handle the illegal packet, but I think that is unlikely.

Source Exodus Commnications Inc. (NETBLK-ECI-7) Netblock: 216.32.0.0 - 216.35.255.255

Note: *Severity is based on initial detect only, as it would be used to determine how much analysis to devote to the detect.*

Severity		(Criticality		Lethality)		(System Countermeasures		Network Countermeasures)
	=	Firewall Scan	+	Random High Ports	-	Up to date Firewall patches	+	Restrictive Firewall
-1		4		3		4		4

Severity after analysis: NA, not a scan

Detect # 10

Initial Detect: This detect is from a firewall log analysis program for logs of 4/10/00

```
18 - 207.71.92.221   / Count SPort  Destination IP:DPort Ptcl/Alert
                         :     2  4645   x.y.z.22:  110  tcp/on unserved port
                         :     4  4657   x.y.z.22:  139  tcp/on unserved port
                         :     4  4637   x.y.z.22:   79  tcp/on unserved port
                         :     4  4664   x.y.z.22:  143  tcp/on unserved port
                         :     4  4149   x.y.z.22:  139  tcp/on unserved port
-----------------------------------------------------------------
```

tcpdump for this detect:

```
19:47:20.548561 207.71.92.221.4149 > abc.139: S 376658775:376658775(0) win 8192 <mss 1460> (DF)
19:47:20.549127 abc.139 > 207.71.92.221.4149: R 0:0(0) ack 376658776 win 0 (DF)
19:47:21.101437 207.71.92.221.4149 > abc.139: S 376658775:376658775(0) win 8192 <mss 1460> (DF)
19:47:21.101809 abc.139 > 207.71.92.221.4149: R 0:0(0) ack 1 win 0 (DF)
```

```
19:47:21.602418 207.71.92.221.4149 > abc.139: S 376658775:376658775(0) win 8192 <mss 1460> (DF)
19:47:21.602775 abc.139 > 207.71.92.221.4149: R 0:0(0) ack 1 win 0 (DF)
19:47:22.105204 207.71.92.221.4149 > abc.139: S 376658775:376658775(0) win 8192 <mss 1460> (DF)
19:47:22.105566 abc.139 > 207.71.92.221.4149: R 0:0(0) ack 1 win 0 (DF)
19:49:06.377451 207.71.92.221.4634 > abc.ftp: S 376764673:376764673(0) win 8192 <mss 1460> (DF)
19:49:06.377942 abc.ftp > 207.71.92.221.4634: S 1391484162:1391484162(0) ack 376764674 win 8760 <mss 1460> (DF)
19:49:06.462988 207.71.92.221.4634 > abc.ftp: . ack 1 win 8760 (DF)
19:49:06.463178 207.71.92.221.4634 > abc.ftp: F 1:1(0) ack 1 win 8760 (DF)
19:49:06.463374 abc.ftp > 207.71.92.221.4634: . ack 2 win 8760 (DF)
19:49:06.475164 207.71.92.221.4635 > abc.telnet: S 376764683:376764683(0) win 8192 <mss 1460> (DF)
19:49:06.475544 abc.telnet > 207.71.92.221.4635: S 1391589630:1391589630(0) ack 376764684 win 8760 <mss 1460> (DF)
19:49:06.556875 207.71.92.221.4635 > abc.telnet: . ack 1 win 8760 (DF)
19:49:06.566713 abc.telnet > 207.71.92.221.4635: P 1:64(63) ack 1 win 8760 (DF)
19:49:06.580852 207.71.92.221.4635 > abc.telnet: F 1:1(0) ack 1 win 8760 (DF)
19:49:06.581145 abc.telnet > 207.71.92.221.4635: . ack 2 win 8760 (DF)
19:49:06.601750 207.71.92.221.4636 > abc.smtp: S 376764839:376764839(0) win 8192 <mss 1460> (DF)
19:49:06.602117 abc.smtp > 207.71.92.221.4636: S 1391699840:1391699840(0) ack 376764840 win 8760 <mss 1460> (DF)
19:49:06.648800 207.71.92.221.4635 > abc.telnet: R 376764685:376764685(0) win 0 (DF)
19:49:06.663466 207.71.92.221.4635 > abc.telnet: R 376764685:376764685(0) win 0
19:49:06.682886 207.71.92.221.4636 > abc.smtp: . ack 1 win 8760 (DF)
19:49:06.683459 207.71.92.221.4636 > abc.smtp: F 1:1(0) ack 1 win 8760 (DF)
19:49:06.684112 abc.smtp > 207.71.92.221.4636: . ack 2 win 8760 (DF)
19:49:06.687540 207.71.92.221.4637 > abc.finger: S 376764947:376764947(0) win 8192 <mss 1460> (DF)
19:49:06.687999 abc.finger > 207.71.92.221.4637: R 0:0(0) ack 376764948 win 0 (DF)
19:49:07.252780 207.71.92.221.4637 > abc.finger: S 376764947:376764947(0) win 8192 <mss 1460> (DF)
19:49:07.253172 abc.finger > 207.71.92.221.4637: R 0:0(0) ack 1 win 0 (DF)
19:49:07.432658 abc.ftp > 207.71.92.221.4634: F 1:1(0) ack 2 win 8760 (DF)
19:49:07.516583 207.71.92.221.4634 > abc.ftp: . ack 2 win 8760 (DF)
19:49:07.778509 207.71.92.221.4637 > abc.finger: S 376764947:376764947(0) win 8192 <mss 1460> (DF)
19:49:07.778833 abc.finger > 207.71.92.221.4637: R 0:0(0) ack 1 win 0 (DF)
19:49:08.355085 207.71.92.221.4637 > abc.finger: S 376764947:376764947(0) win 8192 <mss 1460> (DF)
19:49:08.355476 abc.finger > 207.71.92.221.4637: R 0:0(0) ack 1 win 0 (DF)
19:49:08.443863 207.71.92.221.4644 > abc.80: S 376766656:376766656(0) win 8192 <mss 1460> (DF)
19:49:08.471528 abc.80 > 207.71.92.221.4644: S 1392718365:1392718365(0) ack 376766657 win 8760 <mss 1460> (DF)
19:49:08.558206 207.71.92.221.4644 > abc.80: . ack 1 win 8760 (DF)
19:49:08.558554 207.71.92.221.4644 > abc.80: F 1:1(0) ack 1 win 8760 (DF)
19:49:08.558985 abc.80 > 207.71.92.221.4644: . ack 2 win 8760 (DF)
19:49:08.561133 207.71.92.221.4645 > abc.pop3: S 376766799:376766799(0) win 8192 <mss 1460> (DF)
19:49:08.561653 abc.pop3 > 207.71.92.221.4645: R 0:0(0) ack 376766800 win 0 (DF)
19:49:08.767134 abc.80 > 207.71.92.221.4644: F 1:1(0) ack 2 win 8760 (DF)
19:49:08.850098 207.71.92.221.4644 > abc.80: . ack 2 win 8760 (DF)
19:49:09.058925 207.71.92.221.4645 > abc.pop3: S 376766799:376766799(0) win 8192 <mss 1460> (DF)
19:49:09.059294 abc.pop3 > 207.71.92.221.4645: R 0:0(0) ack 1 win 0 (DF)
19:49:09.556577 207.71.92.221.4645 > abc.pop3: S 376766799:376766799(0) win 8192 <mss 1460> (DF)
19:49:09.556981 abc.pop3 > 207.71.92.221.4645: R 0:0(0) ack 1 win 0 (DF)
19:49:10.059196 207.71.92.221.4645 > abc.pop3: S 376766799:376766799(0) win 8192 <mss 1460> (DF)
19:49:10.059523 abc.pop3 > 207.71.92.221.4645: R 0:0(0) ack 1 win 0 (DF)
19:49:10.154529 207.71.92.221.4656 > abc.113: S 376768368:376768368(0) win 8192 <mss 1460> (DF)
19:49:10.154971 abc.113 > 207.71.92.221.4656: S 1393281632:1393281632(0) ack 376768369 win 8760 <mss 1460> (DF)
19:49:10.237161 207.71.92.221.4656 > abc.113: . ack 1 win 8760 (DF)
19:49:10.237608 207.71.92.221.4656 > abc.113: F 1:1(0) ack 1 win 8760 (DF)
19:49:10.246223 207.71.92.221.4657 > abc.139: S 376768465:376768465(0) win 8192 <mss 1460> (DF)
```

```
19:49:10.329758 abc.113 > 207.71.92.221.4656: . ack 2 win 8760 (DF)
19:49:10.330051 abc.139 > 207.71.92.221.4657: R 0:0(0) ack 376768466 win 0 (DF)
19:49:10.390140 abc.113 > 207.71.92.221.4656: P 1:38(37) ack 2 win 8760 (DF)
19:49:10.390286 abc.113 > 207.71.92.221.4656: F 38:38(0) ack 2 win 8760 (DF)
19:49:10.470796 207.71.92.221.4656 > abc.113: R 376768370:376768370(0) win 0 (DF)
19:49:10.472347 207.71.92.221.4656 > abc.113: R 376768370:376768370(0) win 0
19:49:10.678931 abc.smtp > 207.71.92.221.4636: P 1:45(44) ack 2 win 8760 (DF)
19:49:10.681320 abc.smtp > 207.71.92.221.4636: F 45:45(0) ack 2 win 8760 (DF)
19:49:10.760066 207.71.92.221.4636 > abc.smtp: R 376764841:376764841(0) win 0 (DF)
19:49:10.762190 207.71.92.221.4636 > abc.smtp: R 376764841:376764841(0) win 0
19:49:10.857258 207.71.92.221.4657 > abc.139: S 376768465:376768465(0) win 8192 <mss 1460> (DF)
19:49:10.857666 abc.139 > 207.71.92.221.4657: R 0:0(0) ack 1 win 0 (DF)
19:49:11.366902 207.71.92.221.4657 > abc.139: S 376768465:376768465(0) win 8192 <mss 1460> (DF)
19:49:11.367331 abc.139 > 207.71.92.221.4657: R 0:0(0) ack 1 win 0 (DF)
19:49:11.859104 207.71.92.221.4657 > abc.139: S 376768465:376768465(0) win 8192 <mss 1460> (DF)
19:49:11.859475 abc.139 > 207.71.92.221.4657: R 0:0(0) ack 1 win 0 (DF)
19:49:11.956102 207.71.92.221.4664 > abc.143: S 376770241:376770241(0) win 8192 <mss 1460> (DF)
19:49:11.975603 abc.143 > 207.71.92.221.4664: R 0:0(0) ack 376770242 win 0 (DF)
19:49:12.459907 207.71.92.221.4664 > abc.143: S 376770241:376770241(0) win 8192 <mss 1460> (DF)
19:49:12.460287 abc.143 > 207.71.92.221.4664: R 0:0(0) ack 1 win 0 (DF)
19:49:12.960876 207.71.92.221.4664 > abc.143: S 376770241:376770241(0) win 8192 <mss 1460> (DF)
19:49:12.961224 abc.143 > 207.71.92.221.4664: R 0:0(0) ack 1 win 0 (DF)
19:49:13.460995 207.71.92.221.4664 > abc.143: S 376770241:376770241(0) win 8192 <mss 1460> (DF)
19:49:13.461432 abc.143 > 207.71.92.221.4664: R 0:0(0) ack 1 win 0 (DF)
19:49:13.555183 207.71.92.221.4673 > abc.443: S 376771799:376771799(0) win 8192 <mss 1460> (DF)
19:49:13.580429 abc.443 > 207.71.92.221.4673: S 1394482099:1394482099(0) ack 376771800 win 8760 <mss 1460> (DF)
19:49:13.661926 207.71.92.221.4673 > abc.443: . ack 1 win 8760 (DF)
19:49:13.665380 207.71.92.221.4673 > abc.443: F 1:1(0) ack 1 win 8760 (DF)
19:49:13.665601 abc.443 > 207.71.92.221.4673: . ack 2 win 8760 (DF)
19:49:14.017934 abc.443 > 207.71.92.221.4673: F 1:1(0) ack 2 win 8760 (DF)
19:49:14.106127 207.71.92.221.4673 > abc.443: . ack 2 win 8760 (DF)
```

Analysis: Port scan of frequently used (open) ports.

The technique here is a TCP Syn scan; however, when the scan finds an open port, it completes the 3-way handshake and send a Fin to close the connection. This unusual (non-stealth/friendly) behavior prompted me to look more closely at the firewall logs where I saw:

```
Apr 10 19:42:40 abc.com ssl-gw-0[765]: gethostbyaddr: shieldsup.grc.com. != 207.71.92.221
```

The source appears to be associated with ShieldsUP!. ShieldsUP! is a web based 'service' that will scan your machine and provide a report of what it finds, including a brief summary of issues with any open ports it finds. Users of the service must click buttons that authorize the scanning of their machine, the web page also contains a statement that "*Information gained will not be retained, viewed or used by us in any way for any purpose whatsoever*".

This is active targeting because the user requested the scan.

The scan of port 139 is associated with the 'Test My Shields" button, and tests for windows shares. Thus the initial minute and a half delay before the rest of the ports are scanned, that is triggered by a second button ('Probe My Ports!').

Back to the firewall logs, I can find what IP was accessing grc.com just prior to the scan, and using internal logs can find out what user was assigned that address. At least with internal problems I can work issues directly and don't have to worry about non-response from the detect source ☺.

Source: Gibson Research Corp. (NETBLK-NINT-CF475CC0) NINT-CF475CC0 207.71.92.192 - 207.71.92.223

Note: *Severity is based on initial detect only, as it would be used to determine how much analysis to devote to the detect.*

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

Low		Scan of Firewall		Targeted scan of commonly open ports		Up to date Firewall patches		Restrictive Firewall policy
1		5		4		4		4

Severity after analysis: I don't know enough about Gibson Research Corp to reduce the severity.

Last Updated: July 22nd, 2017

Upcoming Training

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