Global Information Assurance Certification Paper

*** Northcutt, fine job, John did a lot of research including working out the history on some of these. The submittal has some redundancy and could benefit from a more formal analysis process. Detect one may be the result of a denial of service, this should be considered. 81 ***

Practical 1:

On 3/21/2000 the following detect was posted by a.edu on GIAC:

```
Mar 20 06:36:35.103382 209.203.237.176,22 -> 10.1.9.116,1555 PR tcp len 20 44 -AS
Mar 20 10:39:27.452772 209.203.237.176,22 -> 10.0.3.59,1008 PR tcp len 20 44 -AS
Mar 20 10:53:27.561947 209.203.237.176,22 -> 10.0.2.24,1153 PR tcp len 20 44 -AS
Mar 20 10:57:53.592117 209.203.237.176,22 -> 10.1.8.11,1555 PR tcp len 20 44 -AS
Mar 20 11:11:55.257477 209.203.237.176,22 -> 10.1.7.104,1700 PR tcp len 20 44 -AS
Mar 20 11:35:32.071536 209.203.237.176,22 -> 10.0.1.0,1153 PR tcp len 20 44 -AS
Mar 20 11:39:59.168549 209.203.237.176,22 -> 10.1.7.115,1555 PR tcp len 20 44 -AS
Mar 20 12:22:02.986817 209.203.237.176,22 -> 10.1.6.91,1555 PR tcp len 20 44 -AS
Mar 20 13:27:44.067996 209.203.237.176,22 -> 10.1.9.71,1700 PR tcp len 20 44 -AS
Mar 20 14:33:48.729000 209.203.237.176,22 -> 10.0.2.71,1153 PR tcp len 20 44 -AS
Mar 20 14:38:15.569920 209.203.237.176,22 -> 10.1.8.58,1555 PR tcp len 20 44 -AS
Mar 20 16:02:16.111509 209.203.237.176,22 -> 10.1.6.10,1555 PR tcp len 20 44 -AS
Mar 20 17:36:01.123910 209.203.237.176,22 -> 10.1.9.1,1555 PR tcp len 20 44 -AS
Mar 20 18:13:38.504493 209.203.237.176,22 -> 10.0.2.118,1153 PR tcp len 20 44 -AS
Mar 20 18:18:05.237490 209.203.237.176,22 -> 10.1.8.105,1555 PR tcp len 20 44 -AS
Mar 20 19:14:08.830874 209.203.237.176,22 -> 10.1.6.46,1700 PR tcp len 20 44 -AS
Mar 20 21:11:06.870884 209.203.237.176,22 -> 10.0.3.61,1153 PR tcp len 20 44 -AS
Mar 20 21:15:34.074076 209.203.237.176,22 -> 10.1.9.48,1555 PR tcp len 20 44 -AS
Mar 20 21:29:37.770075 209.203.237.176,22 -> 10.1.8.13,1700 PR tcp len 20 44 –AS
```

On 3/22/2000 a similar detect was submitted to GIAC:

```
15:27:06.961763 209.203.237.176.22 > 10.0.3.40.1153:
1074448401:1074448401(0) ack 674711610 win 8192 <mss 16>
(ttl 51, id 28329)
0000: 4500 002c 6ea9 0000 3306 901a d1cb edb0 E..,n...3.....
0010: 0a00 0328 0016 0481 400a c811 2837 483a ...(....@...(7H:
0020: 6012 2000 778d 0000 0204 0010 0000 `. .w.........

15:55:11.628284 209.203.237.176.22 > 10.0.3.51.1008:
S 1075445345:1075445345(0) ack 674711610 win 8192 <mss 64>
(ttl 51, id 60440)
0000: 4500 002c ec18 0000 3306 12a0 d1cb edb0 E..,....3.....
0010: 0a00 0333 0016 03f0 4019 fe61 2837 483a ...3....@..a(7H:
0020: 6012 2000 4184 0000 0204 0040 0000 `. .A......@..

16:13:41.025829 209.203.237.176.22 > 10.1.8.3.1555:
S 2930416281:2930416281(0) ack 674711610 win 8192 <mss 65532>
(ttl 51, id 4031)
0000: 4500 002c 0fbf 0000 3306 8f1e d1cb edb0 E..,....3.....
0010: 0a01 0803 0016 0613 aeaa 9a99 2837 483a ............(7H:
```

0020: 6012 2000 d4ff 0000 0204 fffc 0000 `. ....

16:27:44.950630 209.203.237.176.22 > 10.1.7.96.1700:
S 2485173332:2485173332(0) ack 674711610 win 8192 <mss 65528>
(ttl 51, id 17767)
0000: 4500 002c 4567 0000 3306 5a19 d1cb edb0 E..,Eg..3.Z.....
0010: 0a01 0760 0016 06a4 9420 bc54 2837 483a ...`..... .T(7H:
0020: 6012 2000 cde4 0000 0204 fff8 0000 `. ....

16:56:00.226219 209.203.237.176.22 > 10.1.7.107.1555:
S 2535447425:2535447425(0) ack 674711610 win 8192 <mss 16>
(ttl 51, id 59846)
0000: 4500 002c e9c6 0000 3306 b5ae d1cb edb0 E..,....3.......
0010: 0a01 076b 0016 0613 971f db81 2837 483a ...k........(7H:
0020: 6012 2000 ac27 0000 0204 0010 0000 `. ..'........

17:09:58.608378 209.203.237.176.22 > 10.1.6.72.1700:
S 2072088364:2072088364(0) ack 674711610 win 8192 <mss 24>
(ttl 51, id 5850)
0000: 4500 002c 16da 0000 3306 89be d1cb edb0 E..,....3.......
0010: 0a01 0648 0016 06a4 7b81 8f2c 2837 483a ...H....{..,(7H:
0020: 6012 2000 14a5 0000 0204 0018 0000 `. ....

17:19:35.570658 209.203.237.176.22 > 10.0.1.3.1008:
S 320282727:320282727(0) ack 674711610 win 8192 <mss 56>
(ttl 51, id 32740)
0000: 4500 002c 7fe4 0000 3306 8104 d1cb edb0 E..,....3.......
0010: 0a00 0103 0016 03f0 1317 2067 2837 483a .... g(7H:
0020: 6012 2000 4eb9 0000 0204 0038 0000 `. .N.....8..

17:38:03.964952 209.203.237.176.22 > 10.1.6.83.1555:
S 2117069193:2117069193(0) ack 674711610 win 8192 <mss 65520>
(ttl 51, id 36081)
0000: 4500 002c 8cf1 0000 3306 139c d1cb edb0 E..,....3.......
0010: 0a01 0653 0016 0613 7e2f e989 2837 483a ...S....~/..(7H:
0020: 6012 2000 b846 0000 0204 fff0 0020 `. ..F.......

Analysis: Initial review indicates that both scans came from the same source address. Trace route indicates that the source system name is ect.de which is owned by a German organization.

The fact that the ACK sequence numbers are always the same in the second detect indicates that the packets were crafted.

It may be some sort of port scan since each detect was directed at ports 1008/1153/1555/1700. There are no legitimate uses registered for these ports, and there are no commonly known Trojans that use these ports. In one of the posts there was one reference to "hellnine2000", but the only reference I could find to hellnine was an IRC crew known as hellnine who had lost some "warfare" but was regaining strength. I could not find any details on their techniques.

More likely it is a host scan because further analysis shows that the scans directed at subnet 10.0 were aimed at ports 1008/1153 while those directed at 10.1 were aimed at 1555/1700. In both cases the delta between port numbers is 145. The scan's purpose may be to solicit resets to

map active hosts. The variation of the port numbers might be to complicate the pattern.

Classification: Targeted, malicious and high risk. The risk is high because the same scan has been seen on 2 separate days at 2 seemingly unrelated sites. This may be just a host scan, but the possibility of a new Trojan cannot be discounted.

Follow up: Review logs for any other activity from this source and look for correlation to these scans. Post to GIAC and ask for any similar activity from this or any other source.

On March 17th the following detects submitted to GIAC:

```
-*> Snort! <*- Version 1.5 By Martin Roesch
(roesch@clark.net, www.clark.net/~roesch)
snaplen="68" Entering readback mode....

03/17-00:35:00.636403 24.141.65.11:1079> MY.NET.221.82:6699
TCP TTL:112 TOS:0x0 ID:7976 DF
SFR***1 Seq: 0x52F40 Ack: 0x1FE Win: 0x8010
TCP Options => EOL EOL NOP NOP Sack: 510@59039 EOL EOL EOL EOL
EOL EOL EOL EOL EOL EOL EOL EOL EOL EOL EOL EOL EOL EOL EOL EOL EOL

03/17-00:53:29.010178 128.187.245.108:1866 -> MY.NET.10.119:6699
TCP TTL:111 TOS:0x0 ID:47952 DF
SFRPA*2 Seq: 0x14D Ack: 0xD76E0020 Win: 0x5010
TCP Options => EOL EOL EOL EOL
00 00 ..

03/17-00:53:30.348973 128.187.245.108:1866 -> MY.NET.10.119:6699
TCP TTL:111 TOS:0x0 ID:56400 DF
SFRPA*2 Seq: 0x14D Ack: 0xD76E0021 Win: 0x5010
TCP Options => EOL EOL EOL EOL EOL EOL EOL NOP NOP NOP TS: 4109107200 0

03/17-00:53:32.388257 128.187.245.108:1866 -> MY.NET.10.119:6699
TCP TTL:111 TOS:0x0 ID:4177 DF
SFRPA*2 Seq: 0x14D Ack: 0xD76E0023 Win: 0x5010
TCP Options => EOL EOL EOL EOL EOL EOL SackOK Opt 55 (40):
6988 0014 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000

03/17-00:58:42.940563 194.159.250.7:27070 -> MY.NET.202.66:27005
TCP TTL:48 TOS:0x0 ID:33118 DF
SF***U2 Seq: 0x470CD1 Ack: 0x7B630000 Win: 0x80
TCP Options => Opt 68 (15): 0107 1C9E FF00 804A 0064 002B 0000
80 4A 00 64 00 2B .J.d.+

03/17-00:59:48.694985 24.7.62.224:6699 -> MY.NET.98.85:1078
TCP TTL:109 TOS:0x0 ID:55178 DF
SF**AU2 Seq: 0x12861BB Ack: 0x118D583 Win: 0x5018
1A 2B 04 36 01 28 61 BB 01 18 D5 83 00 73 50 18 .+.6.(a.....sP.
21 3C 1A 45 00 00 95 68 DB 52 81 C9 A4 1E F4 A3 !<.E...h.R.....
EE 25 .%
```

03/17-01:00:21.420967 194.159.250.7:31516 -> MY.NET.202.66:31501
TCP TTL:48 TOS:0x0 ID:38873 DF
SFR***2 Seq: 0x6C2AB8 Ack: 0xE9160000 Win: 0x80
TCP Options => EOL Opt 197 (16): 0000 0001 0213 0667 4ED0 7D00 0000

03/17-01:12:38.824936 MY.NET.211.154:1255 -> 129.81.147.16:6700
TCP TTL:126 TOS:0x0 ID:6158 DF
SF****21 Seq: 0xFD1F13 Ack: 0x881C700 Win: 0x5010
04 E7 1A 2C 00 FD 1F 13 08 81 C7 00 00 C3 50 10 ...,..........P.
22 38 14 A6 20 20 20 20 20 00 "8.. .

03/17-01:13:05.890424 195.11.243.24:27045 -> MY.NET.97.77:27005
TCP TTL:48 TOS:0x0 ID:58740 DF
SF*PA* Seq: 0x919A3E Ack: 0x391B0000 Win: 0x0
39 1B 00 00 07 03 A4 EA 43 0F 0D 07 11 5F FD F8 9.......C...._..
FF 00 00 10 A4 71 .....q

03/17-02:06:33.464442 194.159.250.7:7744 -> MY.NET.202.54:2111
TCP TTL:48 TOS:0x0 ID:59731 DF
SF***U2 Seq: 0x51D2A6 Ack: 0xFC454DE7 Win: 0xF968
TCP Options => Echo Rep: 3813805284 CCNEW: 506801176

03/17-02:07:03.415797 194.159.250.7:771 -> MY.NET.202.54:57149
TCP TTL:48 TOS:0x0 ID:62328 DF
SFR**U1 Seq: 0x0 Ack: 0x45000025 Win: 0x0
00 00 00 00 45 00 00 25 11 A7 00 00 71 11 34 D4 ....E..%....q.4.
81 3B 1B 6D C2 9F A4 05 0A 52 69 91 00 11 .;.m.....Ri...

03/17-02:08:45.834976 194.159.250.7:27035 -> MY.NET.202.54:27005
TCP TTL:48 TOS:0x0 ID:3063 DF
SFRPAU Seq: 0x564364 Ack: 0xB33F0100 Win: 0x100
TCP Options => Opt 68 (15): 0104 F8EC FF2B 8780 8018 0C14 0000
EOL EOL EOL EOL EOL EOL EOL EOL EOL

There appear to be several different events in this detect so I will break out each event as a separate practical.

Practical 2

03/17-00:35:00.636403 24.141.65.11:1079> MY.NET.221.82:6699
TCP TTL:112 TOS:0x0 ID:7976 DF
SFR***1 Seq: 0x52F40 Ack: 0x1FE Win: 0x8010
TCP Options => EOL EOL NOP NOP Sack: 510@59039 EOL EOL EOL EOL
EOL EOL EOL EOL EOL EOL EOL EOL EOL EOL EOL EOL EOL EOL EOL EOL EOL

03/17-00:53:29.010178 128.187.245.108:1866 -> MY.NET.10.119:6699
TCP TTL:111 TOS:0x0 ID:47952 DF
SFRPA*2 Seq: 0x14D Ack: 0xD76E0020 Win: 0x5010
TCP Options => EOL EOL EOL EOL
00 00 ..

03/17-00:53:30.348973 128.187.245.108:1866 -> MY.NET.10.119:6699
TCP TTL:111 TOS:0x0 ID:56400 DF
SFRPA*2 Seq: 0x14D Ack: 0xD76E0021 Win: 0x5010
TCP Options => EOL EOL EOL EOL EOL EOL EOL NOP NOP NOP TS: 4109107200 0

03/17-00:53:32.388257 128.187.245.108:1866 -> MY.NET.10.119:6699
TCP TTL:111 TOS:0x0 ID:4177 DF
SFRPA*2 Seq: 0x14D Ack: 0xD76E0023 Win: 0x5010
TCP Options => EOL EOL EOL EOL EOL EOL SackOK Opt 55 (40):
6988 0014 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000
03/17-00:59:48.694985 24.7.62.224:6699 -> MY.NET.98.85:1078
TCP TTL:109 TOS:0x0 ID:55178 DF
SF**AU2 Seq: 0x12861BB Ack: 0x118D583 Win: 0x5018
1A 2B 04 36 01 28 61 BB 01 18 D5 83 00 73 50 18 .+.6.(a.....sP.
21 3C 1A 45 00 00 95 68 DB 52 81 C9 A4 1E F4 A3 !<.E...h.R.....
EE 25 .%

The pattern that stands out is that there are 3 connections from to MY.NET.10.119:6699. There is also a connection from 24.141.65.11:1079 to MY.NET.221.82:6699. Finally, there is a connection from 24.7.62.224:6699 to MY.NET.98.85:1075.

Port 6699 has been associated with the use of Napster to exchange MP3 files. A traceroute to 24.141.65.11 ends at co59897-a.kico1.on.wave.home.com. The "wave" in the domain name may reference sound wave files lending a little more credence to this possibility. 128.187.245.108 traces back to Brigham Young University and the universities are known to have a problem with Napster in their networks. Napster has received a lot of attention lately on Bugtraq as a target for buffer overflow attacks.

The probes from 128.187.245.108:1866 all have anomalous TCP flags set consistent with a Xmas tree attack. There also appears to be padding at the end of the packets which may be an attempt at a buffer overflow.

The packet from 24.141.65.11 also has anomalous TCP flags set. It also seems to have padding at the end. The source address belongs to Cogeco Cable Systems in Burlington, Ontario, Canada.

The packet from 24.7.62.224:6699 has anomalous TCP flags set. (Is this the nature of Napster? I know that a lot of network administrators are concerned about the amount of their bandwidth being used by it. Is this an attempt by them to slip through filters intended to block them?)

Classification: Targeted, malicious and medium risk. The risk is medium because these scans are targeted at a service that has recently been reported to have buffer overflow problems. However, this exploit reportedly just crashes the client PC which makes it just a nuisance DOS. However, if the target hosts were servers, then the risk would be higher due to the greater loss of service. If this is an attack and it's not directed at a server, it may just be college kids trying to crash their buddies' system. Further, Napster claims that they have patched their servers to "prevent this from occurring".

Follow up: Verify that the target hosts are not servers which would increase the pain of any DOS. Monitor for further probes of this type. If found, report them to Napster.

Practical 3

```
03/17-00:58:42.940563 194.159.250.7:27070 -> MY.NET.202.66:27005
TCP TTL:48 TOS:0x0 ID:33118 DF
SF***U2 Seq: 0x470CD1 Ack: 0x7B630000 Win: 0x80
TCP Options => Opt 68 (15): 0107 1C9E FF00 804A 0064 002B 0000
80 4A 00 64 00 2B .J.d.+
03/17-01:00:21.420967 194.159.250.7:31516 -> MY.NET.202.66:31501
TCP TTL:48 TOS:0x0 ID:38873 DF
SFR***2 Seq: 0x6C2AB8 Ack: 0xE9160000 Win: 0x80
TCP Options => EOL Opt 197 (16): 0000 0001 0213 0667 4ED0 7D00 0000

03/17-01:13:05.890424 195.11.243.24:27045 -> MY.NET.97.77:27005
TCP TTL:48 TOS:0x0 ID:58740 DF
SF*PA* Seq: 0x919A3E Ack: 0x391B0000 Win: 0x0
39 1B 00 00 07 03 A4 EA 43 0F 0D 07 11 5F FD F8 9.......C.... _..
FF 00 00 10 A4 71 .....q

03/17-02:06:33.464442 194.159.250.7:7744 -> MY.NET.202.54:2111
TCP TTL:48 TOS:0x0 ID:59731 DF
SF***U2 Seq: 0x51D2A6 Ack: 0xFC454DE7 Win: 0xF968
TCP Options => Echo Rep: 3813805284 CCNEW: 506801176

03/17-02:07:03.415797 194.159.250.7:771 -> MY.NET.202.54:57149
TCP TTL:48 TOS:0x0 ID:62328 DF
SFR**U1 Seq: 0x0 Ack: 0x45000025 Win: 0x0
00 00 00 00 45 00 00 25 11 A7 00 00 71 11 34 D4 ....E..%....q.4.
81 3B 1B 6D C2 9F A4 05 0A 52 69 91 00 11 .;.m.....Ri

03/17-02:08:45.834976 194.159.250.7:27035 -> MY.NET.202.54:27005
TCP TTL:48 TOS:0x0 ID:3063 DF
SFRPAU Seq: 0x564364 Ack: 0xB33F0100 Win: 0x100
TCP Options => Opt 68 (15): 0104 F8EC FF2B 8780 8018 0C14 0000
EOL EOL EOL EOL EOL EOL EOL EOL
```

Analysis: In the scan from 195.11.243.24, all 5 packets from this address had anomalous TCP flags. It appears to start with the SYN, FIN, and URG flags set. On many systems this combination will elicit a RST/ACK if the port is open or closed. It also appears that the high order reserved bit is set. The second packet has the same flags set with the addition of the RST flag, and the low order reserved bit is set.

Of the ports probed I could find one ambiguous reference to 27005 in Martin Roesch's port database saying its type was "FLEX" and its description was "LM (1-10)". I not sure what either of those things mean although I wonder if LM isn't an abbreviation for LanMan. I couldn't find any reference to this port in any other database.

This may be a combination host mapping and OS fingerprinting attempt. In the scan of MY.NET.202.54 a third packet was sent with all the TCP flags but neither of the reserved bits set. This lends weight to the OS fingerprinting theory.

Several of these packets were directed at destination port 27005. These packets came from two different source addresses which increases my concern. There is no known common use for port 27005.

I was ready to put these guys on my watch list until a traceroute showed this traffic is originating from demon-net which is known for generating garbage TCP packets

Classification: Targeted, probably innocent and low risk. The risk is low because these scans are coming from demon-net and our systems do not appear to be responding. However, because port 27005 was probed on both destination addresses from two different hosts, it would be wise to watch for other probes to this port.

Follow up: Review logs for other probes from 195.11.243.24. Watch for further probes from it. Watch for further probes of port 27005. If found, look for further correlation.

Practical 4

```
03/17-01:12:38.824936 MY.NET.211.154:1255 -> 129.81.147.16:6700
TCP TTL:126 TOS:0x0 ID:6158 DF
SF****21 Seq: 0xFD1F13 Ack: 0x881C700 Win: 0x5010
04 E7 1A 2C 00 FD 1F 13 08 81 C7 00 00 C3 50 10 ...,..........P.
22 38 14 A6 20 20 20 20 20 00 "8.. .
```

Analysis: The connection between 03/17-01:12:38.824936 MY.NET.211.154:1255 -> 129.81.147.16:6700 has both the SYN/FIN flags set which may be an attempt to evade 129.81.147.16's packet filter or firewall. At the least I would look back through my logs to see if there have been any other connections between these two systems. I would also check for any other anomalous traffic involving MY.NET.211.154, and if appropriate, examine the system. It may be comprised.

Classification: Targeted, intent unknown, moderate risk. The primary reasons I classify this as a medium risk is that one of my systems is sending anomalous TCP flags to unknown target ports.

Follow up: Review logs for other anomalous traffic from this system.

Practical 5

OK Northcutt - I'll bite. On 3/13/2000 the following post was made:

```
Mar 12 16:53:14 morton kernel: Packet log: input DENY eth0 PROTO=17
200.191.3.82:28432 63.224.27.201:28431 L=29 S=0x00 I=54328 F=0x0000 T=111
(#66)
Mar 12 16:53:14 pooky kernel: Packet log: input DENY eth0 PROTO=17
200.191.3.82:28432 63.224.27.204:28431 L=29 S=0x00 I=55096 F=0x0000 T=111
Mar 12 16:53:14 www kernel: Packet log: input DENY eth0 PROTO=17
```

200.191.3.82:28432 63.224.27.205:28431 L=29 S=0x00 I=55352 F=0x0000 T=111 (#53)

On 3/17/2000 Rob Hounsell made the following post to GIAC:

I've been seeing a number of probes for port 28431 in the last month on the @home net. Still no idea what they are. Target node address has been sanitized. Visits from Russia, the US, Austria, France, and Quebec.

#Severity, timestamp (GMT), issueId, issueName, intruderIp, intruderName, victimIp, victimName, parameters, count

59, 2000-03-01 23:08:03, 2003502, UDP port probe, 212.46.35.81, , 24.114.X.X, , port=28431, 1

59, 2000-03-02 00:38:31, 2003502, UDP port probe, 216.203.165.134, 216-203-165-134.ticnet.com, 24.114.X.X, , port=28431, 1

59, 2000-03-11 13:48:51, 2003502, UDP port probe, 212.183.65.147, WIEG, 24.114.X.X, , port=28431, 1

59, 2000-03-11 23:46:08, 2003502, UDP port probe, 212.11.164.81, , 24.114.X.X, , port=28431, 1

59, 2000-03-16 17:19:35, 2003502, UDP port probe, 207.236.189.122, DELL7500, 24.114.X.X, , port=28431, 1

Analysis: On the one hand the scans appear to be related because they both scan UDP port 28431. However, the 3/12 scan appears to be an automated scan due to its rapid fire nature. The Hounsell scan is of the low and slow variety.

The 3/12 scan was targeted at different hosts within the same subnet. It is interesting that it skipped .203 and .204. This may indicate that previous recon has been done on this network. The scrubbing of destination addresses on the Hounsell scan prevents learning anything in this regard for that scan.

The packets in the 3/12 scan appear to be crafted because the source port number is constant. The Hounsell scan appears to be crafted, because the IP ID remains constant.

In addition to the locations that Rob Hounsell found, the 3/12 scan originated in Brazil.

Classification: Targeted, intent unknown, unknown risk.

Follow up: Review logs for similar traffic directed at this port. File in the "let's look at this one again later" drawer.

Practical 6

On 3/9/2000 Adam posted the following on GIAC:

Mar 7 23:55:13 beer kernel: Packet log: input DENY ppp0
PROTO=6 203.37.101.21:2345 139.130.12.177:21
L=48 S=0x00 I=59652 F=0x4000 T=126 SYN (#18)

Mar 7 23:55:13 beer kernel: Packet log: input DENY ppp0
PROTO=6 203.37.101.21:2346 139.130.12.177:22
L=48 S=0x00 I=60420 F=0x4000 T=126 SYN (#18)

Mar 7 23:55:13 beer kernel: Packet log: input DENY ppp0
PROTO=6 203.37.101.21:2347 139.130.12.177:23
L=48 S=0x00 I=62212 F=0x4000 T=126 SYN (#18)

Mar 7 23:55:13 beer kernel: Packet log: input DENY ppp0
PROTO=6 203.37.101.21:2349 139.130.12.177:42
L=48 S=0x00 I=64772 F=0x4000 T=126 SYN (#18)

Mar 7 23:55:13 beer kernel: Packet log: input DENY ppp0
PROTO=6 203.37.101.21:2350 139.130.12.177:53
L=48 S=0x00 I=65028 F=0x4000 T=126 SYN (#18)

Mar 7 23:55:14 beer kernel: Packet log: input DENY ppp0
PROTO=6 203.37.101.21:2351 139.130.12.177:69
L=48 S=0x00 I=1541 F=0x4000 T=126 SYN (#18)

Mar 7 23:55:14 beer kernel: Packet log: input DENY ppp0
PROTO=6 203.37.101.21:2352 139.130.12.177:79
L=48 S=0x00 I=2309 F=0x4000 T=126 SYN (#18)

Mar 7 23:55:14 beer kernel: Packet log: input DENY ppp0
PROTO=6 203.37.101.21:2353 139.130.12.177:80
L=48 S=0x00 I=2821 F=0x4000 T=126 SYN (#18)

Mar 7 23:55:14 beer kernel: Packet log: input DENY ppp0
PROTO=6 203.37.101.21:2354 139.130.12.177:110
L=48 S=0x00 I=3077 F=0x4000 T=126 SYN (#18)

Mar 7 23:55:14 beer kernel: Packet log: input DENY ppp0
PROTO=6 203.37.101.21:2355 139.130.12.177:111
L=48 S=0x00 I=4613 F=0x4000 T=126 SYN (#18)

Mar 7 23:55:14 beer kernel: Packet log: input DENY ppp0
PROTO=6 203.37.101.21:2356 139.130.12.177:119
L=48 S=0x00 I=4869 F=0x4000 T=126 SYN (#18)

Mar 7 23:55:14 beer kernel: Packet log: input DENY ppp0
PROTO=6 203.37.101.21:2357 139.130.12.177:143
L=48 S=0x00 I=6661 F=0x4000 T=126 SYN (#18)

Mar 7 23:55:14 beer kernel: Packet log: input DENY ppp0
PROTO=6 203.37.101.21:2358 139.130.12.177:1080
L=48 S=0x00 I=6917 F=0x4000 T=126 SYN (#18)

Mar 7 23:55:14 beer kernel: Packet log: input DENY ppp0
PROTO=6 203.37.101.21:2359 139.130.12.177:1745
L=48 S=0x00 I=7941 F=0x4000 T=126 SYN (#18)

Mar 7 23:55:14 beer kernel: Packet log: input DENY ppp0
PROTO=6 203.37.101.21:2360 139.130.12.177:2301
L=48 S=0x00 I=8709 F=0x4000 T=126 SYN (#18)

Mar 7 23:55:14 beer kernel: Packet log: input DENY ppp0
PROTO=6 203.37.101.21:2361 139.130.12.177:5190
L=48 S=0x00 I=10245 F=0x4000 T=126 SYN (#18)

Mar 7 23:55:14 beer kernel: Packet log: input DENY ppp0
PROTO=6 203.37.101.21:2362 139.130.12.177:5191
L=48 S=0x00 I=12293 F=0x4000 T=126 SYN (#18)

Mar 7 23:55:14 beer kernel: Packet log: input DENY ppp0
PROTO=6 203.37.101.21:2363 139.130.12.177:5192
L=48 S=0x00 I=13573 F=0x4000 T=126 SYN (#18)

Mar 7 23:55:14 beer kernel: Packet log: input DENY ppp0
PROTO=6 203.37.101.21:2364 139.130.12.177:5193
L=48 S=0x00 I=14853 F=0x4000 T=126 SYN (#18)

Mar 7 23:55:15 beer kernel: Packet log: input DENY ppp0
PROTO=6 203.37.101.21:2366 139.130.12.177:5631
L=48 S=0x00 I=17157 F=0x4000 T=126 SYN (#18)

Mar 7 23:55:15 beer kernel: Packet log: input DENY ppp0
PROTO=6 203.37.101.21:2367 139.130.12.177:5632
L=48 S=0x00 I=18181 F=0x4000 T=126 SYN (#18)

Mar 7 23:55:15 beer kernel: Packet log: input DENY ppp0
PROTO=6 203.37.101.21:2368 139.130.12.177:5800
L=48 S=0x00 I=18437 F=0x4000 T=126 SYN (#18)

Mar 7 23:55:15 beer kernel: Packet log: input DENY ppp0
PROTO=6 203.37.101.21:2369 139.130.12.177:5900
L=48 S=0x00 I=18693 F=0x4000 T=126 SYN (#18)

Mar 7 23:55:15 beer kernel: Packet log: input DENY ppp0
PROTO=6 203.37.101.21:2370 139.130.12.177:6000
L=48 S=0x00 I=18949 F=0x4000 T=126 SYN (#18)

Mar 7 23:55:15 beer kernel: Packet log: input DENY ppp0
PROTO=6 203.37.101.21:2371 139.130.12.177:8000
L=48 S=0x00 I=20741 F=0x4000 T=126 SYN (#18)

Mar 7 23:55:15 beer kernel: Packet log: input DENY ppp0
PROTO=6 203.37.101.21:2372 139.130.12.177:8010
L=48 S=0x00 I=21253 F=0x4000 T=126 SYN (#18)

Mar 7 23:55:15 beer kernel: Packet log: input DENY ppp0
PROTO=6 203.37.101.21:2373 139.130.12.177:8080
L=48 S=0x00 I=21765 F=0x4000 T=126 SYN (#18)

Mar 7 23:55:16 beer kernel: Packet log: input DENY ppp0
PROTO=6 203.37.101.21:2377 139.130.12.177:9100
L=48 S=0x00 I=30725 F=0x4000 T=126 SYN (#18)

Mar 7 23:55:16 beer kernel: Packet log: input DENY ppp0
PROTO=6 203.37.101.21:2379 139.130.12.177:12345
L=48 S=0x00 I=32005 F=0x4000 T=126 SYN (#18)

Analysis: Adam correctly identifies this as a port scan. What makes it somewhat interesting is that it contained some ports with which he was not familiar. Unfortunately, he didn't identify which ones he did not know so I will identify as many as I can:

Port	Description	Port	Description	Port	Description
21	FTP Control	22	Ssh	23	Telnet
42	WINS	53	DNS	69	TFTP
79	Finger	80	HTTP	110	POP3
111	Portmapper	119	NNTP	143	IMAP
1080	SOCKS	1745	Remote-winsock	2301	Compaq Insight Management Web Agents
5190	AIM	5191	AOL 1	5192	AOL 2
5193	AOL 3	5631	PC Anywhere Data	5632	PC Anywhere
5800		5900		6000	X-Windows
8000	Irdmi	8010	Wingate 2.1	8080	HTTP Proxy
9100		12345	Netbus Trojan		

Analysis: This leaves ports 5800, 5900 and 9100 as unidentified. The scan was automated as indicated by its rapid fire nature. The source ports increment by 1 indicating a lightly used system - probably a PC. The source IP address is part of a Class C network administered owned by Ms Ingrid Kyle, 15 Luffman Crescent, Gilmore, ACT 2905, AU. It is administered by Telstra Internet who requests that all reports of security breaches be emailed to abuse@telstra.net.

Classification: Targeted, malicious, low risk. I categorize this as low risk because Adam seems to have his defenses up and his network was not comprised by this attack. This is probably the work of a kiddie script.

Follow up: Report probe to Telstra Internet and let them handle it.

Practical 7

On 3/4/2000 Laurie from .edu posted the following on GIAC:

Mar 3 00:04:56 dns3 portsentry[301]: attackalert:
Connect from host: max3-240.max3.hou.infohwy.com/207.90.225.240
to TCP port: 79
Mar 3 00:05:00 dns3 portsentry[301]: attackalert:
Connect from host: max3-240.max3.hou.infohwy.com/207.90.225.240
to TCP port: 143
Mar 3 00:07:09 dns3 portsentry[301]: attackalert:
Connect from host: max3-240.max3.hou.infohwy.com/207.90.225.240
to TCP port: 79
Mar 3 00:07:11 dns3 in.telnetd[11055]:
refused connect from max3-240.max3.hou.infohwy.com
Mar 3 00:07:17 dns3 portsentry[301]: attackalert:
Connect from host: max3-240.max3.hou.infohwy.com/207.90.225.240
to TCP port: 143
Mar 3 00:07:25 dns3 in.telnetd[11131]:
refused connect from max3-240.max3.hou.infohwy.com
Mar 3 00:08:55 dns3 portsentry[301]: attackalert:
Connect from host: max3-240.max3.hou.infohwy.com/207.90.225.240
to TCP port: 79
Mar 3 00:08:56 dns3 in.telnetd[11133]:
refused connect from max3-240.max3.hou.infohwy.com
Mar 3 00:09:02 dns3 portsentry[301]: attackalert:
Connect from host: max3-240.max3.hou.infohwy.com/207.90.225.240
to TCP port: 143
Mar 3 00:09:09 dns3 in.telnetd[11134]:
refused connect from max3-240.max3.hou.infohwy.com

Analysis: This is a common port scan on a DNS server for 3 vulnerable ports: telnet, finger and IMAP. It is difficult to state definitively that this is an automated scan because of the random time intervals (ranging from 1 second up to 2 minutes and 5 seconds) and random sequencing of ports. This is may be an attempt to disguise an automated scan.

Telnet is susceptible to buffer overflows when it is passed either long login names or passwords.

Versions of IMAP based on University of Washington's implementation are also vulnerable to buffer overflow exploits. Among the affected vendors are Microsoft, IBM AIX, Sun, BSD, Red Hat Linux, Caldera Linux, and Debian Linux. CERT has posted two advisories, CA-98.09.imapd and CA-97.09.imap_pop, describing these vulnerabilities. Successful use of this exploit could lead to root access.

Aside from buffer overflow exploits, there is also the possibility that this is a Trojan scan. Port 23 can host the Tiny Telnet Server (TTS) Trojan. Port 79 can be home to the FireHotker Trojan. FireHotkey affects Win95/98 systems and can spawn processes, listen, open the CD drive, etc. Port 143 is not affected by Trojans.

Classification: Targeted, unknown intent, low risk. I categorize this as low risk because Laurie seems to have her defenses up and her network was not comprised by this attack. This appears to be a script kiddie scan since it targets older vulnerabilities.

Follow up: Put max3-240.max3.hou.infohwy.com on the watch list.

On 4/4/2000 Andy at .edu submitted the following:

```
04/01-15:59:26.043293 158.94.234.51:1674 -> MY.NET.70.227:6346
TCP TTL:117 TOS:0x0 ID:27853 DF
SFR**U21 Seq: 0x97FCBA Ack: 0x1141D Win: 0x5018
TCP Options => EOL EOL Opt 80 (40): 579C BBE0 E44A 83B0 0EC3
0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0E C3 ..
04/01-16:00:33.741385 158.94.234.51:230 -> MY.NET.70.227:1674
TCP TTL:117 TOS:0x0 ID:3310 DF
SF**** Seq: 0x18CA0098 Ack: 0x5C0B141D Win: 0x5018
TCP Options => EOL EOL Opt 163 (40): E9E3 DC07 D411 A275 0060
0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
04/01-16:04:40.716885 158.94.234.51:1674 -> MY.NET.70.227:6346
TCP TTL:117 TOS:0x0 ID:61266 DF
SFRP**1 Seq: 0x996CFA Ack: 0x141D Win: 0x5018
TCP Options => EOL EOL Opt 238 (26): 0AE5 E007 D411 9F79 0010
0000 0000 0000 0000 0000 0000 0000 EOL EOL EOL EOL EOL EOL EOL
EOL EOL EOL EOL EOL
04/01-16:06:18.182252 158.94.234.51:1674 -> MY.NET.70.227:6346
TCP TTL:117 TOS:0x0 ID:46459 DF
SF*P*U1 Seq: 0xC30099 Ack: 0xDBD9141E Win: 0x5018
06 8A 18 CA 00 C3 00 99 DB D9 14 1E 06 AB 50 18 ..............P.
00 00 D3 0A 00 00 A0 15 49 6C C4 07 D4 11 9F 25 ........Il.....%
00 10 ..
04/01-16:07:17.708685 24.201.15.107:0 -> MY.NET.202.6:4623
TCP TTL:112 TOS:0x0 ID:53039 DF
SF**AU2 Seq: 0x4C0A90 Ack: 0x9B8D0564 Win: 0x5010
TCP Options => EOL EOL EOL EOL EOL EOL Opt 98 (39): 1E61 040C 000A
0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000
04/01-16:10:45.964767 158.94.234.51:1674 -> MY.NET.70.227:6346
TCP TTL:117 TOS:0x0 ID:5343 DF
SFRPAU21 Seq: 0xDB009A Ack: 0x7786141E Win: 0x5018
39 FF 50 18 00 00 EC A2 00 00 7B 15 49 6C C4 07 9.P.......{.Il..
D4 11 9F 25 00 10 ...%..
04/01-16:15:31.394180 24.201.15.107:4623 -> MY.NET.202.6:76
TCP TTL:112 TOS:0x0 ID:34903 DF
SF*P** Seq: 0xA909B8D Ack: 0x5A063E Win: 0x5010
00 00 00 00 00 00 ......
04/01-16:38:37.904840 129.123.236.50:1116 -> MY.NET.70.227:6346
TCP TTL:110 TOS:0x0 ID:2907 DF
SFR***1 Seq: 0x47D9DA59 Ack: 0x1C81443 Win: 0x5010
TCP Options => EOL EOL EOL EOL EOL EOL Opt 85 (40): 2054 5950 453D
0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000
04/01-17:15:33.055773 24.112.44.237:6688 -> MY.NET.205.106:4042
TCP TTL:115 TOS:0x0 ID:27570 DF
SF*P*U21 Seq: 0x405819 Ack: 0xF01F38 Win: 0xA010
22 38 BD CB 00 00 01 01 05 12 1F 38 64 37 1F 38 "8.........8d7.8
```

69 EB i.
04/01-17:49:35.202459 24.68.74.248:6699 -> MY.NET.206.202:2019
TCP TTL:114 TOS:0x0 ID:24611 DF
SF*P*U21 Seq: 0x12F710 Ack: 0x485 Win: 0x8010
TCP Options => EOL EOL NOP NOP Sack: 1157@54251 EOL EOL EOL EOL
EOL EOL EOL EOL EOL EOL EOL EOL EOL EOL

Practical 8

The following traffic all came from the same source address:

04/01-15:59:26.043293 158.94.234.51:1674 -> MY.NET.70.227:6346
TCP TTL:117 TOS:0x0 ID:27853 DF
SFR**U21 Seq: 0x97FCBA Ack: 0x1141D Win: 0x5018
TCP Options => EOL EOL Opt 80 (40): 579C BBE0 E44A 83B0 0EC3
0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0E C3 ..
04/01-16:00:33.741385 158.94.234.51:230 -> MY.NET.70.227:1674
TCP TTL:117 TOS:0x0 ID:3310 DF
SF**** Seq: 0x18CA0098 Ack: 0x5C0B141D Win: 0x5018
TCP Options => EOL EOL Opt 163 (40): E9E3 DC07 D411 A275 0060
0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
04/01-16:04:40.716885 158.94.234.51:1674 -> MY.NET.70.227:6346
TCP TTL:117 TOS:0x0 ID:61266 DF
SFRP**1 Seq: 0x996CFA Ack: 0x141D Win: 0x5018
TCP Options => EOL EOL Opt 238 (26): 0AE5 E007 D411 9F79 0010
0000 0000 0000 0000 0000 0000 0000 EOL EOL EOL EOL EOL EOL EOL
EOL EOL EOL EOL EOL
04/01-16:06:18.182252 158.94.234.51:1674 -> MY.NET.70.227:6346
TCP TTL:117 TOS:0x0 ID:46459 DF
SF*P*U1 Seq: 0xC30099 Ack: 0xDBD9141E Win: 0x5018
06 8A 18 CA 00 C3 00 99 DB D9 14 1E 06 AB 50 18 ....P.
00 00 D3 0A 00 00 A0 15 49 6C C4 07 D4 11 9F 25 ....Il.....%
00 10 ..
04/01-16:10:45.964767 158.94.234.51:1674 -> MY.NET.70.227:6346
TCP TTL:117 TOS:0x0 ID:5343 DF
SFRPAU21 Seq: 0xDB009A Ack: 0x7786141E Win: 0x5018
39 FF 50 18 00 00 EC A2 00 00 7B 15 49 6C C4 07 9.P.......{.Il..
D4 11 9F 25 00 10 ...%..

Analysis: The packets are directed from the same source host to the same destination host. Four out of the five packets are from the same source port to the same destination port. The source port of 1674 is registered to Intel Proshare Multicast. I could not find any known use for the other ports.

For the exceptional packet the sequence and ACK sequence are way out of line with the others which progress somewhat normally. However, in the other packets there is a strange pattern to the last four positions of the ACK sequence number. The first three packets end with 141D and the next two end with 141E. The leading digits vary wildly going up and down. That makes the packets appear crafted.

Another factor that lends credence to packet crafting is the various anomalous TCP flags set in each packet. This could be a WinNuke attack. However, while WinNuke can be effective on any Windows port listening for data, it is usually targeted at port 139. More likely this is an attempt at OS fingerprinting.

It's a low and slow scan. This can imply evaluation between scans.

While the Intel Proshare Multicast protocol has been demonstrated with Microsoft Netmeeting, I would expect more traffic for an application of that nature. I would also expect to see responses from our system although we may not be logging them. My only guess is OS fingerprinting by evaluating the system's response to the anomalous TCP flags.

Classification: Targeted, unknown intent, medium risk. I categorize this as medium risk because while it is not targeted any known vulnerability, it may be directed by somebody with above average skills.

Follow up: Put 158.94.234.51 on the watch list. Check logging configuration to see if traffic out of our network is being logged. If so, review traffic back to 158.94.234.51.

```
04/01-16:07:17.708685 24.201.15.107:0 -> MY.NET.202.6:4623
TCP TTL:112 TOS:0x0 ID:53039 DF
SF**AU2 Seq: 0x4C0A90 Ack: 0x9B8D0564 Win: 0x5010
TCP Options => EOL EOL EOL EOL EOL EOL Opt 98 (39): 1E61 040C 000A
0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000

04/01-16:15:31.394180 24.201.15.107:4623 -> MY.NET.202.6:76
TCP TTL:112 TOS:0x0 ID:34903 DF
SF*P** Seq: 0xA909B8D Ack: 0x5A063E Win: 0x5010
00 00 00 00 00 00 ......
```

Analysis: The first of these packets originates from source port 0 which immediately causes it to stick out. It is targeted to destination port 4623. Strangely, this becomes the source port for the next scan. The packet ID decrements which is unusual. The TCP flags are anomalous.

Classification: Targeted, unknown intent, low risk. I classify this as low risk because it is not directed at any known vulnerabilities and did not appear to generate a response.

Follow up: Check logs for any other activity related to this source address. Check logging configuration to verify that outbound traffic is being logged.

Practical 10

On 3/4/2000 Laurie posted the following on GIAC:

OLS Co Ltd, Hong Kong

```
Mar 31 19:09:35 hosth snort[75541]:
spp_portscan: PORTSCAN DETECTED from 203.85.30.129
Mar 31 19:09:41 hosth snort[75541]: spp_portscan:
portscan status from 203.85.30.129: 14 connections
across 14 hosts: TCP(14), UDP(0)
Mar 31 19:09:47 hosth snort[75541]: spp_portscan:
End of portscan from 203.85.30.129
--------
Mar 31 19:09:34 203.85.30.129:1542 -> A.B.C.30:98 SYN **S*****
Mar 31 19:09:34 203.85.30.129:1545 -> A.B.C.33:98 SYN **S*****
Mar 31 19:09:38 203.85.30.129:1710 -> A.B.C.197:98 SYN **S*****
Mar 31 19:09:38 203.85.30.129:1714 -> A.B.C.201:98 SYN **S*****
Mar 31 19:09:38 203.85.30.129:1717 -> A.B.C.204:98 SYN **S*****
Mar 31 19:09:38 203.85.30.129:1720 -> A.B.C.207:98 SYN **S*****
Mar 31 19:09:38 203.85.30.129:1727 -> A.B.C.214:98 SYN **S*****
Mar 31 19:09:38 203.85.30.129:1728 -> A.B.C.215:98 SYN **S*****
Mar 31 19:09:38 203.85.30.129:1731 -> A.B.C.218:98 SYN **S*****
Mar 31 19:09:36 203.85.30.129:1748 -> A.B.C.235:98 SYN **S*****
Mar 31 19:09:36 203.85.30.129:2021 -> A.B.D.252:98 SYN **S*****
```

Mar 31 19:09:37 203.85.30.129:1531 -> A.B.C.19:98 SYN **S*****
Mar 31 19:09:39 203.85.30.129:2006 -> A.B.D.237:98 SYN **S*****
Mar 31 19:09:39 203.85.30.129:2073 -> A.B.E.48:98 SYN **S*****

Analysis: This is a SYN scan targeting port 98 – linuxconf. It appears to be an automated scan because of the rapid fire nature especially when the destination addresses are close together.

Linuxconf is a GUI administration tool for Linux based systems. If a system is improperly configured, this could lead to a root compromise and complete victory for the attacker.

The gap in the destination addresses being scanned concerns me. It leads me to wonder if recon has already been done and this is a highly targeted scan. However, the time differences between probes argues against that. Of course, that could be for stealth purposes.

Classification: Targeted, highly malicious , high risk. The primary reasons I classify this as a high risk is that it is targeted at a specific port that could lead to a root compromise. A strong argument can also be made that it is targeted at specific hosts which indicates previous recon has been done.

Follow up: Notify CIRT of potential attack. Review logs for other probes by this source address or to port 98.

I would like to close with a few words upon completing this exercise. First, I appreciate the opportunity to practice what you covered at SANS Orlando. It really does help make the material sink in deeper. Any feedback would be helpful so I can make the necessary corrections to my approach to analysis.

Second, in reviewing the many ports in these scans, I found several good sites that listed port registration and known usage. However, the listings tended to reflect either Trojan usages or registered usages. Very few showed usage of unregistered ports by various major software developers. For instance, older versions of pcAnywhere used port 65301. RealAudio uses port 32000, but I also found a vulnerability in Artisoft Xtramail that occurs on this same port. None of the lists I checked mentioned Artisoft's use of that port. Finally, Macromedia Director uses port 1626. Again, no mention of this is found anywhere.

Could SANS host a repository for ports used by vendors and integrate it with the other port listings that are published? That way if I install a new software package that uses default ports I can share that knowledge with everyone else. This would certainly make the process of evaluating scans more efficient and effective because we could research any known vulnerabilities for the targeted software application.

Finally, I want to thank all the people who shared their traces with GIAC especially Laurie who was very generous in her postings. Since I do not currently have access to these kinds of traces, I could not have completed this exercise without their assistance.

Last Updated: May 22nd, 2017

Upcoming Training

Mentor Session - SEC503	**Oceanside, CA**	**May 29, 2017 - Jun 29, 2017**	**Mentor**
Security Operations Center Summit & Training	**Washington, DC**	**Jun 05, 2017 - Jun 12, 2017**	**Live Event**
SANS Houston 2017	**Houston, TX**	**Jun 05, 2017 - Jun 10, 2017**	**Live Event**
SANS Columbia, MD 2017	**Columbia, MD**	**Jun 26, 2017 - Jul 01, 2017**	**Live Event**
SANS London July 2017	**London, United Kingdom**	**Jul 03, 2017 - Jul 08, 2017**	**Live Event**
SANSFIRE 2017	**Washington, DC**	**Jul 22, 2017 - Jul 29, 2017**	**Live Event**
SANSFIRE 2017 - SEC503: Intrusion Detection In-Depth	**Washington, DC**	**Jul 24, 2017 - Jul 29, 2017**	**vLive**
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**
Baltimore September 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 Seattle 2017	**Seattle, WA**	**Oct 30, 2017 - Nov 04, 2017**	**Live Event**
SANS San Diego 2017	**San Diego, CA**	**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**
SIEM & Tactical Analytics Summit & Training	**Scottsdale, AZ**	**Nov 28, 2017 - Dec 05, 2017**	**Live Event**
SANS OnDemand	**Online**	**Anytime**	**Self Paced**
SANS SelfStudy	**Books & MP3s Only**	**Anytime**	**Self Paced**