GIAC Intrusion Detection Practical Assignment

Glenn Williamson

SANS Parliament Hill 2000

Contents

Submitted: September 22 2000
Conference: SANS Parliament Hill 2000

Detect One

1. The Fields Below Represent:

   a. ID   (Alarm Number)
   b. Type (FireWall Input)
   c. Date (yyyy/mm/dd)
   d. Time (GMT-00:00)
   e. Source IP (xxx.xxx.xxx.xxx)
   f. Source Port (xxx.xxx.xxx.xxx)
   g. Destination IP (111.222.333.444)
   h. Destination Port
   i. Transport Protocol (ICMP/TCP/UDP/IGMP)

```
 A    B        C          D             E            F            G          H
          I
557  FWIN  2000/07/25  20:09    212.254.234.64  42133  111.222.333.444  11
       TCP
558  FWIN  2000/07/25  20:09    212.254.234.64  42134  111.222.333.444  12
       TCP
559  FWIN  2000/07/25  20:09    212.254.234.64  42135  111.222.333.444  13
       TCP
560  FWIN  2000/07/25  20:09    212.254.234.64  42136  111.222.333.444  14
       TCP
561  FWIN  2000/07/25  20:09    212.254.234.64  42137  111.222.333.444  15
       TCP
562  FWIN  2000/07/25  20:09    212.254.234.64  42139  111.222.333.444  17
       TCP
563  FWIN  2000/07/25  20:09    212.254.234.64  42140  111.222.333.444  18
       TCP
564  FWIN  2000/07/25  20:09    212.254.234.64  42141  111.222.333.444  19
       TCP
565  FWIN  2000/07/25  20:09    212.254.234.64  42142  111.222.333.444  20
       TCP
575  FWIN  2000/07/25  20:09    212.254.234.64  42145  111.222.333.444  32
       TCP
576  FWIN  2000/07/25  20:09    212.254.234.64  42146  111.222.333.444  33
       TCP
577  FWIN  2000/07/25  20:09    212.254.234.64  42148  111.222.333.444  34
       TCP
578  FWIN  2000/07/25  20:09    212.254.234.64  42149  111.222.333.444  35
       TCP
579  FWIN  2000/07/25  20:09    212.254.234.64  42150  111.222.333.444  36
       TCP
580  FWIN  2000/07/25  20:09    212.254.234.64  42151  111.222.333.444  37
       TCP
581  FWIN  2000/07/25  20:09    212.254.234.64  42152  111.222.333.444  38
       TCP
582  FWIN  2000/07/25  20:09    212.254.234.64  42153  111.222.333.444  39
       TCP
583  FWIN  2000/07/25  20:09    212.254.234.64  42154  111.222.333.444  40
       TCP
594  FWIN  2000/07/25  20:09    212.254.234.64  42156  111.222.333.444  52
       TCP
595  FWIN  2000/07/25  20:09    212.254.234.64  42157  111.222.333.444  53
       TCP
596  FWIN  2000/07/25  20:09    212.254.234.64  42158  111.222.333.444  51
       TCP
597  FWIN  2000/07/25  20:09    212.254.234.64  42159  111.222.333.444  54
       TCP
598  FWIN  2000/07/25  20:09    212.254.234.64  42160  111.222.333.444  55
```

```
     TCP
599  FWIN  2000/07/25  20:09    212.254.234.64  42161  111.222.333.444  56
     TCP
600  FWIN  2000/07/25  20:09    212.254.234.64  42162  111.222.333.444  57
     TCP
601  FWIN  2000/07/25  20:09    212.254.234.64  42165  111.222.333.444  60
     TCP
612  FWIN  2000/07/25  20:09    212.254.234.64  42169  111.222.333.444  72
     TCP
613  FWIN  2000/07/25  20:09    212.254.234.64  42170  111.222.333.444  73
     TCP
614  FWIN  2000/07/25  20:09    212.254.234.64  42171  111.222.333.444  71
     TCP
615  FWIN  2000/07/25  20:09    212.254.234.64  42172  111.222.333.444  74
     TCP
616  FWIN  2000/07/25  20:09    212.254.234.64  42173  111.222.333.444  75
     TCP
617  FWIN  2000/07/25  20:09    212.254.234.64  42174  111.222.333.444  76
     TCP
618  FWIN  2000/07/25  20:09    212.254.234.64  42175  111.222.333.444  77
     TCP
619  FWIN  2000/07/25  20:09    212.254.234.64  42176  111.222.333.444  78
     TCP
620  FWIN  2000/07/25  20:09    212.254.234.64  42178  111.222.333.444  80
     TCP
629  FWIN  2000/07/25  20:09    212.254.234.64  42180  111.222.333.444  93
     TCP
630  FWIN  2000/07/25  20:09    212.254.234.64  42181  111.222.333.444  91
     TCP
631  FWIN  2000/07/25  20:09    212.254.234.64  42182  111.222.333.444  94
     TCP
632  FWIN  2000/07/25  20:09    212.254.234.64  42183  111.222.333.444  95
     TCP
633  FWIN  2000/07/25  20:09    212.254.234.64  42184  111.222.333.444  96
     TCP
634  FWIN  2000/07/25  20:09    212.254.234.64  42185  111.222.333.444  97
     TCP
635  FWIN  2000/07/25  20:09    212.254.234.64  42186  111.222.333.444  98
     TCP
636  FWIN  2000/07/25  20:09    212.254.234.64  42187  111.222.333.444  99
     TCP
637  FWIN  2000/07/25  20:09    212.254.234.64  42188  111.222.333.444  100
     TCP
```

1. **Source of Trace:**

   a. This Upwards Port Scan was conducted against a @home connected machine.

2. **Detect was generated by:**

   a. Zone Alarm ZoneAlarm Basic Logging Client v2.1.25. It is currently configured to not allow anything but initiated connections required by the @home user. Any attempt at pinging and or scanning the machine 111.222.333.444 are denied via the host machine dropping the packet and not responding to the connection request via the mentioned ports. The fields are explained in the first paragraph on the previous page.

3. **Probability the source address was spoofed:**

   a. Unlikely. The protocol used to conduct the upwards port was tcp(i.e. this requires a 3-way handshake). This attacker is looking/appearing to be gathering information. The source address can be pinged. The unusual event is the IP Address is a cgi server which allows users to set up accounts. It also acts as a DNS server and allows users to use (SMTP,TELNET,FTP). There is the high possibility that the machine that conducted this scan/probe has been compromised(hacked).

4. **Description of attack:**

a. Reconnaissance – Information Gathering. This is a simple upwards port scan
for open services offered by the Operating System of this machine. A port scan such as this will provide a list of open ports or services and with this information an attacker can look for and exploit possible security vulnerabilities in the Operating System. This is the most basic form of TCP scanning. The connect system call provided by an operating system is used to open a connection to every interesting port on the scanned machine. If the port is listening, connect will succeed, otherwise the port remains unreachable.

b. Port scanning is like ringing the doorbell to see whether someone's home. The police usually can't and won’t do anything about it. They usually have to wait until a crime is committed or has been committed. In Germany and Singapore, port scanning cannot be prosecuted. There is the possibility, if a computer system is affected too much by a port scan, one can view it that the port scan was, in fact, a denial-of-service (DoS) attack, which is usually an offense.

5. **Attack Mechanism:**

a. The attacker is attempting to establish a 3-way handshake on ports 11 – 100. This does not provide an in-depth, nor a very stealthy scan. It does not hide the scan or conduct it as a slow-scan should. This is the 3rd such scan originating out of the same IP address. He has left out some well known ports (Telnet,FTP), this may be a way of not sending suspicion of active port scanning for any Telnet or FTP services that this IP address may be providing. This Port Scan actually took a total of 24 seconds.

6. **Correlations:**

a. Lots of information resides on the Internet about Port Scans, they are the one of the first steps in trying to exploit a host machine. NMAP is one of the tools to do Port Scanning. This scan does not look nor act like a NMAP scan as NMAP usually does not scan in specific port sequence.

7. **Evidence of active targeting:**

a. Being as how the attacker is scanning ports 11-100, this appears to

actively targeting this IP address, unfortunately this scan can be targeted against a range of hosts operating in the same network. The IP address that was scanned was a simple @home machine, it was probably picked from a list of addresses that were actively scanned for open ports.

8. **Severity:**

a. (Criticality + Lethality) – (system countermeasure + countermeasure)

```
                 2      +     1                  4            +
2 = -3
```

9. **Defensive Recommendation:**

a. A standard firewall or router will automatically drop or block these connections unless it is poorly configured. Being as this was a @home machine a tool such as Conseal (Firewall), Snort, or BlackIce will easily show and certain IDS/Firewalls can deny access to any of these ports as long as no service is currently using these ports.

10. Multiple choice Test question:

What are the Telnet and FTP Ports that were not scanned during this trace?

a. 21 , 53
b. 98 , 25
b. 21 , 23
d. Not sure, don't know what ports are used for Telnet and FTP

Answer is C, if you answered D, then you missed Stephen Northcutt telling you what ports to always remember.

Detect Two

```
1.
a.
[**] IDS197 - DDoS - Trin00 [**]
08/28-11:07:47.986342 aaa.bbb.ccc:2923 -> eee.fff.ggg:27444
UDP TTL:128 TOS:0x0 ID:25287
Len: 19
70 6E 67 20 6C 34 34 61 64 73 6C D8 6C D8 6C D8  png l44adsl.l.l.
6C D8                                            l.

[**] IDS197 - DDoS - Trin00 [**]
08/28-11:07:47.977198 aaa.bbb.ccc:2923 -> eee.fff.ggg:27444
UDP TTL:128 TOS:0x0 ID:25287
Len: 19
png l44adsl.l.l.l.

A            B                C                   D
2000-07-24  08:47:17  Trinoo master activity   Bugs Bunny 4
E                                                   F             G
H
port=27444|34555&data=png_[]|44|png_|44ads|   xxx.xxx.xxx      001524
59
```

1. **Source of Trace:**

   a. This is a Trin00 Master Server against potential daemon ("broadcast") host on a Class B Network host machine.

2. **Detect was generated by:**

   a. **SNORT**

   c. Explanation of Fields of Interest For Second Example

   BlackICE

   A. Date (yyyy-mm-dd)
   B. Time (hh:mm:ss)
   C. Detect (Attack)
   D. Src Host (Name)
   E. Dst Port (Number)
   F. Dst Host (IP Address)
   G. Attack ID (Number)
   H. Severity (Number)

3. **Probability the source address was spoofed:**

   a. Extremely low, the attacker was on the same internal Intranet, it is a trusted machine by this host

4. **Description of attack:**

   a. Over a period of one hour multiple attacks were tried against this IP address on the same Intranet. It was done to determine the difference in which events SNORT detected and which ones BlackICE detected. Trin00 networks have potentionaly been set up on thousands of machines across the Internet. Most of the machines that were compromised across the Internet were either Sun Solaris 2.X or Linux operating system machines. This was targeted against a Windows NT machine with SP6 with the latest hot fixes in place.

b. The master and the daemons are usually password protected, this is done to prevent system administrators and other hacker groups from being able to take over the network that compromises the machines that can make up a Trin00 network.

5. **Attack Mechanism:**

a. A Trin00 network is made up of a master server (master.c) and the trin00
daemon (ns.c). An attacker(s) may control one or more master servers, each master server can control multiple daemons. The daemons carry out coordinated packet based attacks, against one or more victim systems. The network usually is run with (attacker(s)→ master(s)→daemon(s)→ against victim or victims.

b. All that is needed is an ability to establish TCP connections to the master hosts using a "telnet" session and the password to the master server to be able to wage a somewhat massive, highly coordinated, DOS (Denial of Service) attack against a target. Communication from the master to the daemon is accomplished via UDP packets on port 27444. Had a TCP connection been made back to the master from the potential daemon(this machine) it would have been via port 31335/udp.

d. The initial "png" command sent to the daemon by the master would have
replied with a "PONG" on the port 31335/udp, had the daemon(host) been previously compromised. Also noted in the activity above, is the default password "144ads1". No activity of this nature was noted as a sniffer was being run on the host machine to see if it replied.

e. A Trin00 network is established usually by compromising machines in the exploitation of the Remote Procedure Call (RPC) buffer overrun bugs, these can be found in some of the following services, statd, cmsd, and ttdbserverd. If an attacker modifies the source code prior to setting up a Trin00 network, he(she/it) can or may change may of the details such as prompts, commands, passwords and TCP/UDP port numbers.

6. **Correlations:**

a. There is a great deal of reference material written about DDOS Trinoo (Trin00). Most of the information can be found at the following sights. CERT Incident Note IN-99-07 http://www.cert.org/incident_notes/IN-99-07.html#trinoo. It can also be found and reference at the Mitre organization under the CVE list, CAN-2000-0138 (under review) this is a candidate for inclusion in CVE, http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2000-0138.

7. **Evidence of active targeting:**

a. This was a single machine that the initial "png" "144ads1" was sent to. If the Master wanted to or had the need to, he could address the previous command line to multiple hosts (potential daemons) to establish, or control a Trin00 network.

8. **Severity:**

a. (criticality + Lethality) – (system countermeasure + countermeasure)
2 + 1 5 +
4 = -6

9. **Defensive Recommendation:**

a. One step is to Deny Invalid Source IP Addresses leaving from this

network, This will minimize the chance that this network will be the source of a Spoofed DDOS attack. An up to date Intrusion detection system (SNORT/SHADOW) can be configured to easily recognize the initial "png" coming into the host machines, this connection can either then be dropped or shunned from the router.

10. Can the configuration of Trin00 be configured to operate on different ports then those normally associated with it?

a. Yes
b. No
c. Not Sure
d. All of the Above

Answer is A, unless just a script kiddie then the answer would be C Not Sure.

Detect Three

```
Origin        Source       Destination
07:34:03.480321   62.0.41.179.111 > 111.222.0.1.111: SF 1515234464:1515234464(0) win 1028 (ttl 29, id 39426)
07:34:03.481342   111.222.255.130 > 62.0.41.179: icmp: host 111.222.0.1 unreachable - admin prohibited filter (ttl 255,
id 1565)
07:34:03.484963   62.0.41.179.111 > 111.222.0.0.111: SF 1515234464:1515234464(0) win 1028 (ttl 30, id 39426)
07:34:03.506296   62.0.41.179.111 > 111.222.0.2.111: SF 1515234464:1515234464(0) win 1028 (ttl 29, id 39426)
07:34:03.517026   62.0.41.179.111 > 111.222.0.3.111: SF 1515234464:1515234464(0) win 1028 (ttl 29, id 39426)
07:34:03.540531   62.0.41.179.111 > 111.222.0.4.111: SF 1515234464:1515234464(0) win 1028 (ttl 29, id 39426)
07:34:03.561377   62.0.41.179.111 > 111.222.0.5.111: SF 1515234464:1515234464(0) win 1028 (ttl 29, id 39426)
07:34:03.585732   62.0.41.179.111 > 111.222.0.6.111: SF 1515234464:1515234464(0) win 1028 (ttl 29, id 39426)
07:34:03.596453   62.0.41.179.111 > 111.222.0.7.111: SF 1515234464:1515234464(0) win 1028 (ttl 29, id 39426)
07:34:03.621052   62.0.41.179.111 > 111.222.0.8.111: SF 1515234464:1515234464(0) win 1028 (ttl 29, id 39426)
07:34:03.635757   62.0.41.179.111 > 111.222.0.9.111: SF 1515234464:1515234464(0) win 1028 (ttl 29, id 39426)
07:34:03.658459   62.0.41.179.111 > 111.222.0.10.111: SF 1515234464:1515234464(0) win 1028 (ttl 29, id 39426)
07:34:03.677883   62.0.41.179.111 > 111.222.0.11.111: SF 1515234464:1515234464(0) win 1028 (ttl 29, id 39426)
07:34:03.703343   62.0.41.179.111 > 111.222.0.12.111: SF 1515234464:1515234464(0) win 1028 (ttl 29, id 39426)
07:34:03.723630   62.0.41.179.111 > 111.222.0.13.111: SF 1515234464:1515234464(0) win 1028 (ttl 29, id 39426)
07:34:03.776910   62.0.41.179.111 > 111.222.0.15.111: SF 1515234464:1515234464(0) win 1028 (ttl 29, id 39426)
07:34:03.777180   62.0.41.179.111 > 111.222.0.14.111: SF 1515234464:1515234464(0) win 1028 (ttl 29, id 39426)
>>>>
07:34:03.918987   62.0.41.179.111 > 111.222.0.23.111: SF 1515234464:1515234464(0) win 1028 (ttl 29, id 39426)
07:34:03.939883   62.0.41.179.111 > 111.222.0.24.111: SF 140872056:140872056(0) win 1028 (ttl 29, id 39426)
07:34:04.000165   62.0.41.179.111 > 111.222.0.25.111: SF 140872056:140872056(0) win 1028 (ttl 29, id 39426)
07:34:04.000889   111.222.255.130 > 62.0.41.179: icmp: host 111.222.0.25 unreachable - admin prohibited filter (ttl
255, id 1566)
07:34:04.007136   62.0.41.179.111 > 111.222.0.26.111: SF 140872056:140872056(0) win 1028 (ttl 29, id 39426)
07:34:04.007732   62.0.41.179.111 > 111.222.0.27.111: SF 140872056:140872056(0) win 1028 (ttl 29, id 39426)
07:34:04.019649   62.0.41.179.111 > 111.222.0.28.111: SF 140872056:140872056(0) win 1028 (ttl 29, id 39426)
>>>
08:17:58.299301  62.0.41.179.111 > 111.222.255.240.111: SF 79549353:79549353(0) win 1028 (ttl 29, id 39426)
08:17:58.300309  111.222.255.130 > 62.0.41.179: icmp: host 111.222.255.240 unreachable - admin prohibited filter (ttl
255, id 11339)
08:17:58.314825  62.0.41.179.111 > 111.222.255.241.111: SF 79549353:79549353(0) win 1028 (ttl 29, id 39426)
08:17:58.337114  62.0.41.179.111 > 111.222.255.242.111: SF 79549353:79549353(0) win 1028 (ttl 29, id 39426)
08:17:58.354157  62.0.41.179.111 > 111.222.255.243.111: SF 79549353:79549353(0) win 1028 (ttl 29, id 39426)
```

//note// This scan was conducted against an entire Class B Network, most of the trace was left out for brevity purposes, and to save printing and making this document.

1. **Source of Trace:**

   a. Internal Class B Network

2. **Detect was generated by:**

   a. Netranger (Cisco Secure Intrusion Detection System), SNORT.

3. **Probability the source address was spoofed:**

   a. Low, since these are SYN/FYN scan packets being directed at this class B network, this scan would not be very useful if replies were not sent back to the source address. Having dealt with the machine that originated this probe, the machine that conducted this scan had been previously compromised(hacked).

4. **Description of attack:**

   a. An RPC scan is used by a remote user (hacker) to determine the presence of UNIX based hosts on the targeted network (s). The attacker is looking for the existence and exploitability of the RPC service statd on any of the hosts he scanned.

b. Systems found to be offering RPC services are often exploited through well-known vulnerabilities in both the RPC portmapper and other applications that may be available through the Portmap service.

f. Unix servers and workstations use many applications of RPC, there are daemons, lock managers and license managers. The first step in the attack is to determine if it is running s specific service on any of these machines.

5. **Attack Mechanism:**

a. This scan used the anomalous flag bit settings. By setting both the syn and fin bits in the TCP header, the attacker may be attempting to gather operating system information using a remote OS fingerprinting technique. Because each system would respond differently based on Operating System, the attacker can attempt to identify the OS installed on the targeted host(s)

6. **Correlations:**

a. This type of activity has been recorded many times over the past couple of months, there are many articles written on it specifically CVE –1999-0208(rpc.ypupdated (NIS) allows remote users to execute arbitrary commands, CVE-199-0493(rpc.statd allows remote attackers to forward RPC calls to the local operating system via the SM_MON and SM_NOTIFY commands, which in turn could be used to exploit other bugs such as in automound)

7. **Evidence of active targeting:**

a. As the attacker(prober) scanned an entire Class B network for RPC
services, he was not activily targetting specific machines but general scan of the entire network. This can be seen as specifically targeting several machines but hiding it into the complete scan. This was neither a slow nor hidden scan.

8. **Severity:**

a. (criticality + Lethality) – (system countermeasure + countermeasure)

4 + 1 - 5 + 5 = -5

9. **Defensive Recommendation:**

a. An interesting note, the border routers have sent back a msg “unreachable –admin prohibited filter”. This does provide the attacker what certain machines are there, this would enable him to try and further exploit those actual hosts. This router has an admin prohibited filter in place that safeguards High-Value hosts on connected networks from this type of scanning. It would be better to actually have the routers shun the packet instead of sending the message. A better enhancement would be to cease admin prohibited filter messages. These messages will indicate which hosts are active on the shielded networks and provide and estimate of the importance. If this filter is removed the internet address space will become a "IP PACKET BLACKHOLE" for incoming RPC connection requests.

g. Ensuring all Unix based machines have the RPC service turned off if not needed. Also ensure the the router does not allow incoming packets to port 111 on the inside host machines. An IDS/Firewall/Router are all able to provide information and service if they are specifically configured to disallow port 111 service inside and outside of the Class B Network.

10. Multiple choice Question:

On which port does RPC services normally reside

a. 143
b. 98
c. 111
d. 100

Answer: C

Detect Four

```
Origin      Source        Destination

16:54:58.460997 207.152.157.6.10451 > 111.222.0.1.635: S 1048977893:1048977893(0) win 512 <mss 1460> (ttl 48, id
34125)
16:55:01.456957 207.152.157.6.10451 > 111.222.0.1.635: S 1048977893:1048977893(0) win 32120 <mss 1460> (ttl 48,
id 34208)
16:55:01.457625 111.222.255.130 > 207.152.157.6: icmp: host 111.222.0.1 unreachable - admin prohibited filter (ttl 255,
id 32847)
16:55:01.504965 207.152.157.6.12301 > 111.222.0.1.110: S 2951021530:2951021530(0) win 512 <mss 1460> (ttl 48, id
34210)
16:55:04.498811 207.152.157.6.12301 > 111.222.0.1.110: S 2951021530:2951021530(0) win 32120 <mss 1460> (ttl 48,
id 34299)
16:55:04.499494 111.222.255.130 > 207.152.157.6: icmp: host 111.222.0.1 unreachable - admin prohibited filter (ttl 255,
id 32852)
16:55:04.557462 207.152.157.6.12802 > 111.222.0.1.143: S 3801774974:3801774974(0) win 512 <mss 1460> (ttl 48, id
34300)
16:55:07.545576 207.152.157.6.12802 > 111.222.0.1.143: S 3801774974:3801774974(0) win 32120 <mss 1460> (ttl 48,
id 34397)
16:55:07.546262 111.222.255.130 > 207.152.157.6: icmp: host 111.222.0.1 unreachable - admin prohibited filter (ttl 255,
id 32865)
16:55:07.598197 207.152.157.6.14062 > 111.222.0.1.53: S 899410416:899410416(0) win 512 <mss 1460> (ttl 48, id
34398)
16:55:10.584800 207.152.157.6.14062 > 111.222.0.1.53: S 899410416:899410416(0) win 32120 <mss 1460> (ttl 48, id
34510)
16:55:10.585463 111.222.255.130 > 207.152.157.6: icmp: host 111.222.0.1 unreachable - admin prohibited filter (ttl 255,
id 32870)
16:55:10.639389 207.152.157.6.16435 > 111.222.0.1.21: S 640757607:640757607(0) win 512 <mss 1460> (ttl 48, id
34512)
16:55:13.624887 207.152.157.6.16435 > 111.222.0.1.21: S 640757607:640757607(0) win 32120 <mss 1460> (ttl 48, id
34602)
16:55:13.625522 111.222.255.130 > 207.152.157.6: icmp: host 111.222.0.1 unreachable - admin prohibited filter (ttl 255,
id 32876)

17:45:37.267762 207.152.157.6.21083 > 111.222.1.1.635: S 21325939:21325939(0) win 512 <mss 1460> (ttl 48, id
46032)
17:45:37.268797 111.222.255.130 > 207.152.157.6: icmp: host 111.222.1.1 unreachable - admin prohibited filter (ttl 255,
id 38162)
17:45:37.313006 207.152.157.6.21084 > 111.222.1.1.110: S 4256948489:4256948489(0) win 512 <mss 1460> (ttl 48, id
46033)
17:45:40.309811 207.152.157.6.21084 > 111.222.1.1.110: S 4256948489:4256948489(0) win 32120 <mss 1460> (ttl 48,
id 46094)
17:45:40.310448 111.222.255.130 > 207.152.157.6: icmp: host 111.222.1.1 unreachable - admin prohibited filter (ttl 255,
id 38172)
17:45:40.354737 207.152.157.6.21757 > 111.222.1.1.143: S 3568089615:3568089615(0) win 512 <mss 1460> (ttl 48, id
46095)
17:45:43.350180 207.152.157.6.21757 > 111.222.1.1.143: S 3568089615:3568089615(0) win 32120 <mss 1460> (ttl 48,
id 46187)
17:45:43.350839 111.222.255.130 > 207.152.157.6: icmp: host 111.222.1.1 unreachable - admin prohibited filter (ttl 255,
id 38176)
17:45:43.395434 207.152.157.6.22921 > 111.222.1.1.53: S 1605178929:1605178929(0) win 512 <mss 1460> (ttl 48, id
46188)
17:45:46.389079 207.152.157.6.22921 > 111.222.1.1.53: S 1605178929:1605178929(0) win 32120 <mss 1460> (ttl 48,
id 46280)
17:45:46.389827 111.222.255.130 > 207.152.157.6: icmp: host 111.222.1.1 unreachable - admin prohibited filter (ttl 255,
id 38180)
17:45:46.434134 207.152.157.6.24163 > 111.222.1.1.21: S 274300058:274300058(0) win 512 <mss 1460> (ttl 48, id
46283)
17:45:49.429667 207.152.157.6.24163 > 111.222.1.1.21: S 274300058:274300058(0) win 32120 <mss 1460> (ttl 48, id
46401)

17:46:16.318976 207.152.157.6.31877 > 111.222.2.1.635: S 1488440166:1488440166(0) win 512 <mss 1460> (ttl 48, id
47028)
17:46:16.319723 111.222.255.130 > 207.152.157.6: icmp: host 111.222.2.1 unreachable - admin prohibited filter (ttl 255,
id 38214)
17:46:16.364558 207.152.157.6.31931 > 111.222.2.1.110: S 1124542946:1124542946(0) win 512 <mss 1460> (ttl 48, id
47030)
17:46:19.360351 207.152.157.6.31931 > 111.222.2.1.110: S 1124542946:1124542946(0) win 32120 <mss 1460> (ttl 48,
id 47100)
17:46:19.361007 111.222.255.130 > 207.152.157.6: icmp: host 111.222.2.1 unreachable - admin prohibited filter (ttl 255,
id 38225)
17:46:19.405241 207.152.157.6.1927 > 111.222.2.1.143: S 3847054876:3847054876(0) win 512 <mss 1460> (ttl 48, id
47106)
17:46:22.399138 207.152.157.6.1927 > 111.222.2.1.143: S 3847054876:3847054876(0) win 32120 <mss 1460> (ttl 48,
id 47196)
17:46:22.399794 111.222.255.130 > 207.152.157.6: icmp: host 111.222.2.1 unreachable - admin prohibited filter (ttl 255,
id 38232)
17:46:22.444805 207.152.157.6.3733 > 111.222.2.1.53: S 3419423810:3419423810(0) win 512 <mss 1460> (ttl 48, id
47197)
17:46:25.439292 207.152.157.6.3733 > 111.222.2.1.53: S 3419423810:3419423810(0) win 32120 <mss 1460> (ttl 48, id
47329)
17:46:25.439921 111.222.255.130 > 207.152.157.6: icmp: host 111.222.2.1 unreachable - admin prohibited filter (ttl 255,
id 38240)
17:46:25.484687 207.152.157.6.4436 > 111.222.2.1.21: S 3077082541:3077082541(0) win 512 <mss 1460> (ttl 48, id
47331)
17:46:28.479548 207.152.157.6.4436 > 111.222.2.1.21: S 3077082541:3077082541(0) win 32120 <mss 1460> (ttl 48, id
47433)

17:46:52.880952 207.152.157.6.13394 > 111.222.3.1.635: S 2638751503:2638751503(0) win 512 <mss 1460> (ttl 48, id
48008)
17:46:52.881691 111.222.255.130 > 207.152.157.6: icmp: host 111.222.3.1 unreachable - admin prohibited filter (ttl 255,
id 38311)
17:46:52.928156 207.152.157.6.13398 > 111.222.3.1.110: S 2502966003:2502966003(0) win 512 <mss 1460> (ttl 48, id
48009)
17:46:55.920916 207.152.157.6.13398 > 111.222.3.1.110: S 2502966003:2502966003(0) win 32120 <mss 1460> (ttl 48,
id 48098)
17:46:55.921710 111.222.255.130 > 207.152.157.6: icmp: host 111.222.3.1 unreachable - admin prohibited filter (ttl 255,
id 38316)
17:46:55.967389 207.152.157.6.14499 > 111.222.3.1.143: S 592621807:592621807(0) win 512 <mss 1460> (ttl 48, id
48099)
17:46:58.961727 207.152.157.6.14499 > 111.222.3.1.143: S 592621807:592621807(0) win 32120 <mss 1460> (ttl 48, id
48200)
17:46:58.962376 111.222.255.130 > 207.152.157.6: icmp: host 111.222.3.1 unreachable - admin prohibited filter (ttl 255,
id 38325)
17:46:59.009401 207.152.157.6.15688 > 111.222.3.1.53: S 1159945659:1159945659(0) win 512 <mss 1460> (ttl 48, id
48203)
17:47:02.001298 207.152.157.6.15688 > 111.222.3.1.53: S 1159945659:1159945659(0) win 32120 <mss 1460> (ttl 48,
id 48318)
```

111.222.4.1 – 111.222.22.1 were also scanned but have been left out for brevity sakes.

```
17:59:37.864164 207.152.157.6.14943 > 111.222.23.1.635: S 4196990321:4196990321(0) win 512 <mss 1460> (ttl 48,
id 2371)
17:59:37.865178 111.222.255.130 > 207.152.157.6: icmp: host 111.222.23.1 unreachable - admin prohibited filter (ttl
255, id 39666)
17:59:37.909435 207.152.157.6.14956 > 111.222.23.1.110: S 2709717675:2709717675(0) win 512 <mss 1460> (ttl 48,
id 2372)
17:59:40.908184 207.152.157.6.14956 > 111.222.23.1.110: S 2709717675:2709717675(0) win 32120 <mss 1460> (ttl
48, id 2457)
17:59:40.908857 111.222.255.130 > 207.152.157.6: icmp: host 111.222.23.1 unreachable - admin prohibited filter (ttl
255, id 39673)
17:59:40.953228 207.152.157.6.16310 > 111.222.23.1.143: S 1422298785:1422298785(0) win 512 <mss 1460> (ttl 48,
id 2458)
17:59:43.951252 207.152.157.6.16310 > 111.222.23.1.143: S 1422298785:1422298785(0) win 32120 <mss 1460> (ttl
48, id 2529)
17:59:43.955722 111.222.255.130 > 207.152.157.6: icmp: host 111.222.23.1 unreachable - admin prohibited filter (ttl
255, id 39678)
17:59:44.003433 207.152.157.6.16642 > 111.222.23.1.53: S 2389051786:2389051786(0) win 512 <mss 1460> (ttl 48, id
2531)
17:59:46.998263 207.152.157.6.16642 > 111.222.23.1.53: S 2389051786:2389051786(0) win 32120 <mss 1460> (ttl 48,
id 2609)
17:59:46.998910 111.222.255.130 > 207.152.157.6: icmp: host 111.222.23.1 unreachable - admin prohibited filter (ttl
255, id 39682)
17:59:47.042004 207.152.157.6.17812 > 111.222.23.1.21: S 83594792:83594792(0) win 512 <mss 1460> (ttl 48, id
2614)
17:59:50.038634 207.152.157.6.17812 > 111.222.23.1.21: S 83594792:83594792(0) win 32120 <mss 1460> (ttl 48, id
2723)
17:59:50.039297 111.222.255.130 > 207.152.157.6: icmp: host 111.222.23.1 unreachable - admin prohibited filter (ttl
255, id 39690)
```

1. **Source of Trace:**

   a. Internal Class B Network

2. **Detect was generated by:**

   a. Netranger (Cisco Secure Intrusion Detection System), activity was first noted on a Netranger Alarm, then the traffic was captured with TCP Dump.

3. **Probability the source address was spoofed:**

   a. Low, this attack against these hosts were targeted at specific network based hosts, this was not a DOS attack or there would be little point at spoofing the source IP address and the responses would not be sent back to the originator unless the originator is the source IP address.

4. **Description of attack:**

   a. This was specifically targeted at a class B Network starting at 111.222.0.1 and the final packet was sent to 111.222.23.1. He sent a SYN(S) packet against these ports 635(MountD), 110(POP3), 143 (IMAPD), 53(DNS), 21(Telnet). The scan/probe may have been stopped, if attacker is not getting the response that he is looking for, why continue?

   b. This was an exploit driven scripted attack.

5. **Attack Mechanism:**

   a. The tool that did this enables the user(attacker) to scan whole domains and complete ranges of IP addresses to discover well-known vulnerabilities in the services that were scanned. This is possible for the Intruders to use a tool called "Multiscan or Mscan".

   b. The check in mscan for vulnerable systems is very simple, All it does is an "rpcinfo –p system" and searchs for the text "rstatd"

   c. This attack was specifially targeted at the ports that were found in the above trace, the unusual item is the way it scans the network looking for more than likely web servers. This assumes that everyone puts there web servers at their .1 address space. Had they been Web Servers running Linux older versions, they it would have been easier to compromise them ie; buffer overflows on specific services, if they had replied to any of the connections requests.

6. **Correlations:**

a. Information on specific attacks against certain ports using "mscan" can be found at http://www.ja.net/CERT/JANET-CERT/mscan.html

b. Information on some of the ports that were scanned and the vulnerabilities can be found at:

http://www.cert.org.advisories/CA-98.12.mountd.html
http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0002 Buffer overflow in NFS mountd gives remote access to remote attackers, mostly in Linux systems.
Lance Spitzner also submitted information on the port 635 scan, it can be found at:
http://www.shmoo.com/mail/fw1/jun99/msg00004.html

7. **Evidence of active targeting:**

a. By the amount of Syn packets sent to each of the specific IP addresses, he specifically is looking for active targets in this scan, but being as he targeted a number of machines, this is mainly information gathering (establishing connection on designated port) for each of the IP Addresses scanned.

8. **Severity:**

a. (criticality + Lethality) – (system countermeasure + countermeasure)
5 + 1 - 5 + 4 = -3

9. **Defensive Recommendation:**

a. Border routers will activily drop(shun) these connection requests from outside of the network if they have been configured this way. I noticed the way they only respond once with the ICMP message no matter how many requests were sent to that specific port in the scan. This is accomplished due to the fact the setting on the router is set for 2 seconds before it will send another ICMP message for the same connection request.

b. Many firewalls including Firewall 1 are setup to deny connections to these ports from outside → in, this should be made a default setting for any firewall installation. The Deny All → Deny All is getting to be the pretty much defacto standard in firewall installation but it will be a while before all deployable firewall take advantage of this default setting.

c. Many Operating System vendors have implemented patches dealing with Multiscan or Mscan network scanning activity. All patches should be applied to the operating system.

10. **Multiple choice Question:**

What ports will Mcsn/Multiscan tool scan for?

a. 1000,1001,1002,1003
b. 33664,33665,33337,12345
c. 1234,1080,8080,31337
d. 23,80,143,53,110,111

Answer: d

Assignment 2

Evaluate an Attack

The attack being presented here is Trinity V3, originally this Distributed Denial of Service Attack that was first discovered on two Host machines residing at XXX University.

Originally, these machines had been compromised by a Server belonging to a small ISP in another state. How this tool (Trinity V3) was originally discovered while working on the RPC scans conducted by two different sights. I have decided to do this evaluation based on the original files that were pulled from one of the compromised host machines, vice what has been posted on the web, if the information is found to be different from what has been posted on the Internet, it is due to the configuration of Trinity V3 that was installed on the 2 host machines. It was originally sent to CERT and the FBI, references to it can now be found at:
http://www.securiteam.com/securitynews/Trinity_v3_Distributed_Denial_of_Service_tool.html
Also at: http://xforce.iss.net/alerts/advise59.php

A current listing of hosts that have Trinity v3 infected boxes can be found at http://www.documents.cyberabuse.org/?doc=3

This list found on cyberabuse may not be a complete list of all machines, having gone through the logs captured off of one of the machines, it seems the hacker was able to use the IRIX exploitable buffer overflow that has been discovered in telnetd daemon which can lead to root compromise. SGI has investigated the issue and recommends certain steps that neutralize the exposer. The telnetd daemon can be exloited remotely over an untrusted network.

1. A certain XXX university (xx.58.1.36) conducted an Automated Remote Procedure Call scan targeting the entire Class-B Internet Network owned by this company. This was done on the 17th Aug 00 at 1040L EST by a certain server at XXX university.

   a. The two machines were almost certainly copromised via the rpc.statd problem: http://www.redhat.com/support/errata/RHSA-2000-043-03.html

b. There were a total of twelve machines compromised at XXX University, but only two were found to have the Trinity V3 program also on them.

2. A certain private company (xxx.126.95.202) in Kansas City, MO conducted an Automated Remoted scan targeting the entire Class-B Internet Network owned by this company. This was done on the 16th Aug 00 at 1542L EST.

3. Tracking through system logs both machines were interconnected by the same attacker(script kiddie/hacker). He had compromised the Server (email/web) at a small ISP, after compromising that machine, he proceeded to compromise multiple machines at various Universities and Privat Orgainizations. He used an RPC scan for various machines as well as hunting out individual machines at the schools. He/She was able to telnet in and run automount daemon agains some of these machines.

4. During this write up Trinity V3 may also be referred to as a program idle.so

5. Originally when discovered on the XXX University machines the program was thought to be called Trinity V3 by Self.

Trinity V3 currently supports the following types of attacks:

Tudp is used for **UDP Flood**
Tfrag is used for **Fragment Flood**
Tsyn is used for **Syn Flood**
Trst is used for **Reset Flood**
Trnd is used for **RandomFlag Flood**
Tack is used for **Ack Flood**
Tnull is used for **Null Flood**
Testab is used for Unknown
? is used for **Establish Flood**

//note**//** Currently it is not known what Testab is meant for, it maybe another Flood attempt. The current Establish Flood the acronym is currently not known.

4. The program idle.so was run on a Red Hat 6.2 that is not connected to the Internet, and found this is the program's initial behavior.

   a. The Trinity V3 program will currently try to make a TCP connection to port 6667 on the commonly used IRC servers listed below:

204.127.145.17	irc2.worldnet.att.net, newbrunwick.nj.us.undernet.or
216.24.134.10	irc.lvdi.net, lasvegas.nv.us.undernet.org
208.51.158.10	newyork.ny.us.undernet.org
199.170.91.114	irc.io.com, austin.tx.us.undernet.or
207.173.16.33	irc.aros.net, saltlake.ut.us.undernet.org
207.96.122.250	irc.erols.com, arlington.va.us.undernet.org
205.252.46.98	irc.cais.net, mclean.va.us.undernet.org

216.225.7.155	undernet.freei.net, seattle.wa.us.unernet.org
205.188.149.3	irc-i02.irc.aol.com, washington.dc.us.undernet.org
207.69.200.131	atlanta.ga.us.undernet.org
207.114.4.35	u2.abs.net, baltimore.md.us.undernet.org

b. If after the initial connection attempt is made, if it does not succeed, the Trinity program will sleep for 5 seconds, then select another (not necessarily different) server at random from the list, and try to connect to that one on tcp port 6667. As far as can be found this loop will continue to find a server and can run forever or until it is manually stopped.

C. When a successful tcp connection is made, the trinity program will compose a 9-character username using the following rules:

1. The first part of the username is the hostname of the locale machine, truncated to at most 6 characters. The remaining characters (at least 3, and at most are 8) are randomly selected lowercase letters.

For example:
1. if the hostname is "wyoming", the username might be "wyominqzj.
2. if the hostname is "a", the username might be "akrpbefxh".

2. When the username is "wyominqzj" the Trinity progam would send following to IRC server over the existing port-6667 tcp connection:

```
USER: wyominqzj wyominqzj wyominqzj :wyominqzj
NICK: wyominqzj
```

C. It will then wait for a response from the IRC server(using the "select" system call). If the IRC server responds, the trinity program will then send (over the existing port-6667 tcp connection:

```
MODE wyominqzj +i
JOIN #b3eblebr0x zerblat
MODE #b3eblebr)x +sk zerblat
```

C. It is not known whether the attack commands are exclusively received over this IRC connection, or whether they might be received from clients who connect to the Trinity program, using one of it's listening ports (e.g., tcp port 39168 as was found in the original logs from one of the compromised machines)

C. The following is hardcoded in the binary:

1. The channel name #b3eblebr0x
2. Various message strings indicating when attacks begin e.g,

PRIVMSG PRIVMSG %s :(trinity) udpflood started

PRIVMSG %s :(trinity) randomflagsflood started
Various message strings indicating when attacks end, e.g.,
PRIVMSG %s :(trinity) Udpflood completed. %d packets/sec
PRIVMSG %s :(trinity) randomflags flood completed. %d packets/sec

4) The numerical IP address of the 11 IRC servers
5) The key zerblat that is used to join the channel #b3eblebr0x.
6) Other strings associated with the trinity programs' operation.

G. On August 17 at 17:00 GMT. XXX received a brief excerpt of an IRC session that involved use of trinity program:

```
        <self> port 0
        <ns2_loyth> (trinity) i will now hit on random ports
        <mail_uuxb> (trinity) i will now hit on random ports
        <ns2_alpmw> (trinity) i will now hit on random ports
        <self> tsyn apekatt 213.112.57.4:213.112.57.4 60
        <ns2_loyth> (trinity) {:self!self@cvx-sto-2-102.ppp.netlink.se}: tsyn goodpass
213.112.57.4:213.112.57.4 60
        <ns2_loyth> (trinity) {:self!self@cvx-sto-2-102.ppp.netlink.se}: tsyn goodpass
        213.112.57.4:213.112.57.4 60^M
        <dns1-bwax> (trinity) synflood stated
        <www-whwtu> (trinity) synflood started
        <stetsoxrp> (trinity) synflood started.^M
        <ns2_loyth> (trinity) synflood completed. 13972 packets/sec
        <comtrdgud> (trinity) synflood completed 2285 packets/sec
        <proxy_cky> (trinity) synflood startd
        <gw-tjdheh> (trinity) synflood completed. 91 packets/sec
```

C. The XXX university hosts, were running RED Hat 6.0 i386 and Red Hat 6.2 1386, The Trinity program was named /usr/lib/idsl.so and was a 248004-bye statically linked ELF binary.

Another binary was found in /var/spool/uucp/uucico. This is a simple backdoor program that listens on TCP ports for connections. When connections are established, the attacker sends a passwork to get a root shell . This uucico file is not the one normally associated with Linux, When a connection is established then the ports for the source machine changes. Some of the ports found to be listening are: 996, 39168 and then when connected the port for the source IP address is 4383. This occurs when the program idle.so is running.

The two uucico processes running on both of the Hosts were listening on ports 996, 4248 39168, and 33270.

The following files were changed/compromised on the 2 host machines:

--A 3392-byte dynamicaly linked i386 ELF binary named /usr/sbin/inetd that contained the following strings:
/usr/bib/inetd
/var/spool/uucp
uucici
/var/spool/uucp/ucico
/usr/lib
idle.so
/usr/lib/idle.so
exec1
chdir
fork

execv

--A 4312-byte dynamically linked i386 ELF binary named /var/spool/uucp/uucico that contained the following strings:
exec1
dup2
socket
accept
bind
signal
read
strncmp
setpgrp
listen
fork
htons
-csh
/bin/chs

- A modifies version of /bin/ps was found on the compromised machine that does not list the uucico and the idle.so processes.
- On one of the two systems, the standard version of /usr/sbin/inetd was copied to /usr/lib/inetd, and the standard version of /bin/ps was copied to /usr/lib/libsup.a

The current state of Trinity V3 is unknown, there are known to be currently at least 100 machines that were infected, when Trinity V3 was first found there were over 390 machines compromised around the world. Although written up in Xforce, Security Focus and SANS, having worked on this from the beginning, it was first discovered on 2 machines at the University. This as was first reported to CERT and the FBI(NPIC), all reports that have been found do not contain all the available information. The person who first launched this is suspected to be from Sweden, the Swedish version of CERT (TaliaCirt) is currently looking into the person who may have first launched this DDOS. There is now another version(change) of Trinity that is currently operating on the Internet, this one has the potential to do more harm and it is currently using ADSL machines that are connected 24/7 on the Internet. DDOS can and will become a major disturbance on the internet that are becoming harder and harder to protect against.

Assignment 3

Analyse This

The scans to be shown in this analysis are the Snort Detects from http://www.sans.org/PH2000/snort/index.htm which held a large number of Snort detects. After a specific period of time the various detects will no longer be available for public analysis. These detects appear to be from what would assume to be the 30,000 ft view, as we have no correlating data. Some analysis may appear to be incomplete, but from the data given this is some of the things that have been found in the researched logs. From time to time, the power may have failed or the disk was full so the logs do not properly represent all of the activity that occurred over the specific time frame.

DETECT 1 Analyzed

Date	Time	Source IP	Destination IP	Bit
Jun 30	4:23:48 AM	195.132.120.31:4159	MY.NET.130.7:21	**S*****
Jun 30	4:23:51 AM	195.132.120.31:4164	MY.NET.130.12:21	**S*****
Jun 30	4:23:48 AM	195.132.120.31:4161	MY.NET.130.9:21	**S*****
Jun 30	4:23:48 AM	195.132.120.31:4163	MY.NET.130.11:21	**S*****
Jun 30	4:23:49 AM	195.132.120.31:4165	MY.NET.130.13:21	**S*****
Jun 30	4:23:49 AM	195.132.120.31:4166	MY.NET.130.14:21	**S*****
Jun 30	4:23:49 AM	195.132.120.31:4168	MY.NET.130.16:21	**S*****
Jun 30	4:23:49 AM	195.132.120.31:4170	MY.NET.130.18:21	**S*****

This activity which occurred on Jun 30 appears to be a port scan specifically targeting Port 21 which started at 04:23 on the 30th of June and continued till 04:47(same day), he continued scanning till he reached network MY.NET.143.220. He may have been specifically looking for FTP service and establishing a 3-way hand shake as only the SYN bit was set. The snort rule that detects this appears to look for the specific port. As long as no machine responds to this connection request there is nothing to fear, but if certain machines are responding there is the potential that they can be compromised.

Recommendation: If FTP service is not required then disallow the service on all host machines. Being that we have not done a complete discovery of network structure and of host machines, we should have a look at our firewall and ensure it is correctly blocking services not required. We should deny all that is not absolutely required to maintain security integrity.

DETECT 2 Analysis

Month	Time	Source IP	Source	Dest IP	Dest	Prot	0x13
Aug 4	5:39:27	24.7.157.43	3722	MY.NET.226.151	27374	SYN	**S*****
Aug 4	5:39:27	24.7.157.43	3723	MY.NET.226.152	27374	SYN	**S*****
Aug 4	5:39:27	24.7.157.43	3724	MY.NET.226.153	27374	SYN	**S*****
Aug 4	5:39:27	24.7.157.43	3725	MY.NET.226.154	27374	SYN	**S*****
Aug 4	5:39:27	24.7.157.43	3726	MY.NET.226.155	27374	SYN	**S*****
Aug 4	5:39:27	24.7.157.43	3727	MY.NET.226.156	27374	SYN	**S*****

1. Although I have only pasted a portion of the activity targeted against Destination Port 27374, I have brought it to the attention of the people that own this network. There have been multiple instances of port scans specifically looking to establish a 3 way handshake with this network (MY.NET.XXX.XXX).

2. Port 27374 is known to be used by Subseven Ver 2.1. The SubSeven backdoor is one of many backdoor programs that attackers can use to access your computer system without your knowledge or consent. With the SubSeven backdoor, an attacker can do the following:

   a. Shut down or restart you computer,
   b. Retrieve most saved and cached passwords,
   d. Modify your system registry,
   e. Upload, Download, and delete files from your System.

3. SubSeven is a powerful backdoor that is widely used against Windows systems. With the most recent versions, a remote attacker can do anything to a victim’s computer that could be done locally. For these reasons, SubSeven should be removed immediately if found on this network.

4. Recommendation: The SubSeven backdoor can be very difficult to remove manually, because the executable is difficult to locate and identify on your system. You can and should use an anti-virus program to remove the SubSeven backdoor if it is found residing on host machines. You should download and install one of these virus scanners:

   Http://www.symantec.com/nav/indexA.html
   http://software.mcafee.com/centers/download/
   http://www.trend.com/pc-cillin/2

   Run the anti-virus program to scan your system for this backdoor, the virus scanner should find and remove the backdoor from your computer. The Consequences of this program is to **Gain Access**.

DETECT 3 Analysis

Month	Time	Source IP	Source	Dest IP	Dest Port	Proto	0x13
Jul 27	2:04:25	211.60.222.3	3881	MY.NET.104.93	53	SYN	**S*****
Jul 27	2:04:25	211.60.222.3	3884	MY.NET.104.96	53	SYN	**S*****
Jul 27	2:04:25	211.60.222.3	3885	MY.NET.104.97	53	SYN	**S*****
Month	**Time**	**Source IP**	**Source**	**Dest IP**	**Dest Port**	**Proto**	**0x13**
Jul 29	21:02:5	207.155.88.2	1567	MY.NET.7.60	53	SYN	**S*****
Jul 29	21:02:5	207.155.88.2	1568	MY.NET.7.61	53	SYN	**S*****
Jul 29	21:02:5	207.155.88.2	1569	MY.NET.7.62	53	SYN	**S*****
Month	**Time**	**Source IP**	**Source**	**Dest IP**	**Dest Port**	**Proto**	**0x13**
Jul 29	21:11:1	207.155.88.2	2347	MY.NET.25.223	53	SYN	**S*****
Jul 29	21:11:1	207.155.88.2	2352	MY.NET.25.228	53	SYN	**S*****
Jul 29	21:11:1	207.155.88.2	2356	MY.NET.25.232	53	SYN	**S*****

1. What we have here is the potential of one or two items, it seems over the period of time with these Snort logs, there have been numerous attempts at sending the SYN Bit at multiple host/network based machines. There can be multiple reasons for this to occur. One of the potential problems if the users of any of these host machines are running AntiSniff version 1.01

2. AntiSniff was a program that was written and released by L0pht Heavy Industries in the middle of last year. It attempts to determine if a machine on a local network segment is listening to traffic that it is not directed to it.

3. The possibility is if AntiSniff is configured to run the DNS test, and only during the time the test is running is there the potential for problems. There is a potential to cause a buffer overflow on the system running AntiSniff, if the packet is crafted appropriately this overflow scenario can be exploited to execute arbitrary code on the system. The same problem occurs in both the Unix and Windows version of this program.

4. This is a vulnerability that should not be ignored and has even been found in other promiscuous mode detection programs as well.

5. Recommendations: If running AntiSniff ensure you do not run the DNS tests on version 1.01 or the Researchers version 1.0. If you need to run the program and tests you should download the newer version of AntiSniff version 1.02 or version 1-1

6. There is also the possibility that this was simply a SYN request against Host/Network machines for DNS service requests to do a DNS Zone Transfer if the hosts are DNS Servers. This should not be allowed from outside IP addresses as the potential is there for DNS poisoning or to capture your whole IP/Name structure for your network. This can be accomplished via your installed Firewall.

7. It was also noted that there was a SYN-FIN scan conducted by 202.0.178.98 on the 28 of June @ 20:07. This was done against

MY.NET.XXX.XXX, this is another DNS probe but this time with 2 of the bits sent to try and get the machines to respond.

DETECT 4 Analysis

Month	Time	Source IP	Source	Dest IP	Dest	Proto	0x13
Jul 11	10:07:1	211.112.142.	2209	MY.NET.1.2	98	SYN	**S*****
Jul 11	10:07:1	211.112.142.	2214	MY.NET.1.7	98	SYN	**S*****
Jul 11	10:07:1	211.112.142.	2220	MY.NET.1.13	98	SYN	**S*****
Month	**Time**	**Source IP**	**Source**	**Dest IP**	**Dest**	**Proto**	**0x13**
Jul 26	11:08:2	209.61.158.2	4649	MY.NET.254.234	98	SYN	**S*****
Jul 26	11:08:2	209.61.158.2	4710	MY.NET.254.242	98	SYN	**S*****
Jul 26	11:08:2	209.61.158.2	4850	MY.NET.254.246	98	SYN	**S*****
Jul 24	21:56:5	209.123.109.	1305	MY.NET.98.118	98	SYN	**S*****

1. As we can see in the above trace it appears to be multiple IP addresses attempting to connect to Port 98 on these host machines. As a matter of record there was a total of 34,631 connection requests against these machines. There is note of this type of scan detected with the Dragon IDS and the correlation on this type of scan can be found at http://www.sans.org/y2k/122599.htm

2. Port 98 is associated with (linuxconf) this is a Linux problem that enables the machine to be compromised if it has not been properly been configured and protected. Linuxconf is an administration system for the Linux operating system, it is a relatively new, GUI approach and will sometimes be run by newer less experience admins. It is a configuration and an activator for Linux.

3. If there is no need to allow Port service requests against this port then it should be immediately blocked at either the firewall or depending at the router it can be blocked(shunned) from there if there is a need to. If none of the aforementioned hosts/network based machines are currently running Linux then the need for protection is limited, however the tighter some security policies the less chance for machines to be compromised.

DETECT 5 Analysis

Field1	DATE	Field5	Field8	Field10
1303	06/29-	Queso	129.21.145.131:577	MY.NET.217.98:113
1541	06/29-	Queso	24.3.29.155:1344	MY.NET.6.44:110
1307	06/29-	Queso	129.21.145.131:747	MY.NET.217.98:20

Field1	DATE	Field5	Field7	Field9	Field10
10014	07/08-	NMAP	ping!	209.218.228.46:5	MY.NET.1.8:53
10013	07/08-	NMAP	ping!	209.218.228.46:8	MY.NET.1.8:53
34677	07/11-	NMAP	ping!	195.54.105.6:53	MY.NET.1.8:53
3426	06/27-	NMAP	ping!	195.54.105.6:80	MY.NET.1.9:53

1. As you can see by this trace route it looks like 2 different IP addresses tried to actively scan 2 of MY.NET IP addresses. This has been noted in the Snort logs as QUESO, this is a tool used for scanning IP addresses. These are not all of the Queso or NMAP entries, only a portion is shown to show the potential problems.

2. The second on seen are for intercepts that are seen as a NMAP ping gaisnt port 53 of MY.NET. This can again be used to look for DNS servers on our host machines.

3. As long as the personal that maintain the firewall have a complete understanding of how Nmap and Queso work to actively scan your system then there will be no apparent problems to identify these scans in the wild and respond accordingly.

4. Information in regards to these scans as well as Nmap scans can be found at http://www.securityfocus.com/frames/?focus=ids&content=/focus/ids/articles/portscan.htm.

DETECT 6 Analysis

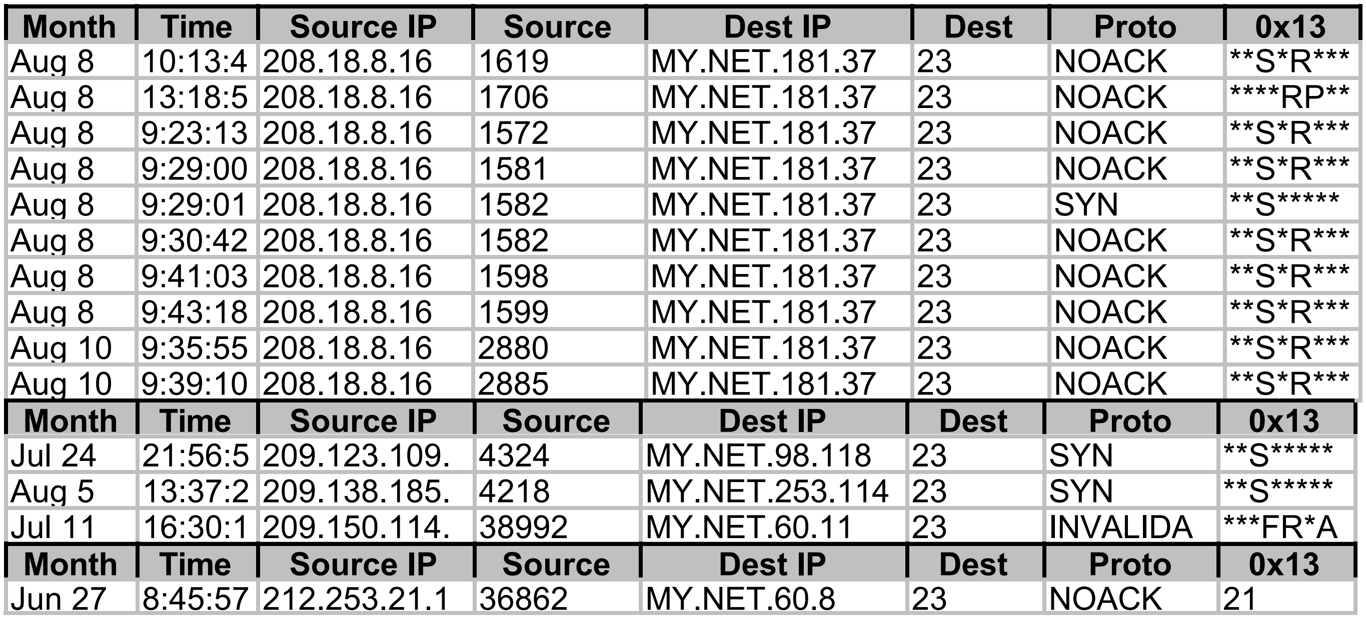
Mont	Day	Time	Source IP	Source	Dest IP	Dest	0x13
Jul	24	23:50:4	211.7.235.4	109	MY.NET.1.5	109	**SF****
Jul	24	23:50:4	211.7.235.4	109	MY.NET.1.15	109	**SF****
Jul	24	23:50:4	211.7.235.4	109	MY.NET.1.57	109	**SF****
Jul	24	23:50:4	211.7.235.4	109	MY.NET.1.65	109	**SF****
Jul	24	23:50:4	211.7.235.4	109	MY.NET.1.111	109	**SF****
Jul	24	23:50:4	211.7.235.4	109	MY.NET.1.151	109	**SF****
Jul	24	23:50:5	211.7.235.4	109	MY.NET.2.193	109	**SF****
Jul	24	23:50:5	211.7.235.4	109	MY.NET.2.221	109	**SF****
Jul	24	23:50:5	211.7.235.4	109	MY.NET.4.15	109	**SF****
Jul	24	23:55:4	211.7.235.4	109	MY.NET.60.231	109	**SF****
Jul	24	23:56:2	211.7.235.4	109	MY.NET.68.16	109	**SF****
Jul	24	23:57:0	211.7.235.4	109	MY.NET.76.72	109	**SF****
Jul	24	23:57:0	211.7.235.4	109	MY.NET.76.134	109	**SF****
Jul	24	23:57:5	211.7.235.4	109	MY.NET.85.49	109	**SF****
Jul	24	23:57:5	211.7.235.4	109	MY.NET.85.59	109	**SF****
Jul	24	23:57:5	211.7.235.4	109	MY.NET.85.120	109	**SF****
Jul	24	23:58:3	211.7.235.4	109	MY.NET.94.8	109	**SF****
Jul	24	23:58:3	211.7.235.4	109	MY.NET.94.95	109	**SF****
Jul	24	23:58:3	211.7.235.4	109	MY.NET.94.112	109	**SF****
Jul	24	23:58:4	211.7.235.4	109	MY.NET.94.230	109	**SF****

1. For the purpose of brevity I have left out the total number of SYN/FIN connection attempts generated by 211.7.235.4(unresolved). It seems this IP address is not out there as far as being able to trace back. It belongs to GFI-NET2-JP but you are unable to ping it or resolve host name. This actually belongs to the Asian Pacific Network.

2. During analysis of the SYN-FIN scan a total of 99 alerts to be from this external IP address were found. As this does not show the host machines sending a reset or any type of traffic what we can assume from this is the IP address was trying to intrude (do a recon) of the machines in a sense if able he/she would then have the opportunity, to potentially compromise the machines that responded.

3. Recommendation: Port 109 happens to be used for Post Office Protocol – Version 2, it is highly recommended if you are currently using POP2 then you upgrade systems to Post Office Protocol Version 3, this would stop any potential compromise associated with POP2.

DETECT 7 Analyzed



Month	Time	Source IP	Source	Dest IP	Dest	Proto	0x13
Aug 8	10:13:4	208.18.8.16	1619	MY.NET.181.37	23	NOACK	**S*R***
Aug 8	13:18:5	208.18.8.16	1706	MY.NET.181.37	23	NOACK	****RP**
Aug 8	9:23:13	208.18.8.16	1572	MY.NET.181.37	23	NOACK	**S*R***
Aug 8	9:29:00	208.18.8.16	1581	MY.NET.181.37	23	NOACK	**S*R***
Aug 8	9:29:01	208.18.8.16	1582	MY.NET.181.37	23	SYN	**S*****
Aug 8	9:30:42	208.18.8.16	1582	MY.NET.181.37	23	NOACK	**S*R***
Aug 8	9:41:03	208.18.8.16	1598	MY.NET.181.37	23	NOACK	**S*R***
Aug 8	9:43:18	208.18.8.16	1599	MY.NET.181.37	23	NOACK	**S*R***
Aug 10	9:35:55	208.18.8.16	2880	MY.NET.181.37	23	NOACK	**S*R***
Aug 10	9:39:10	208.18.8.16	2885	MY.NET.181.37	23	NOACK	**S*R***

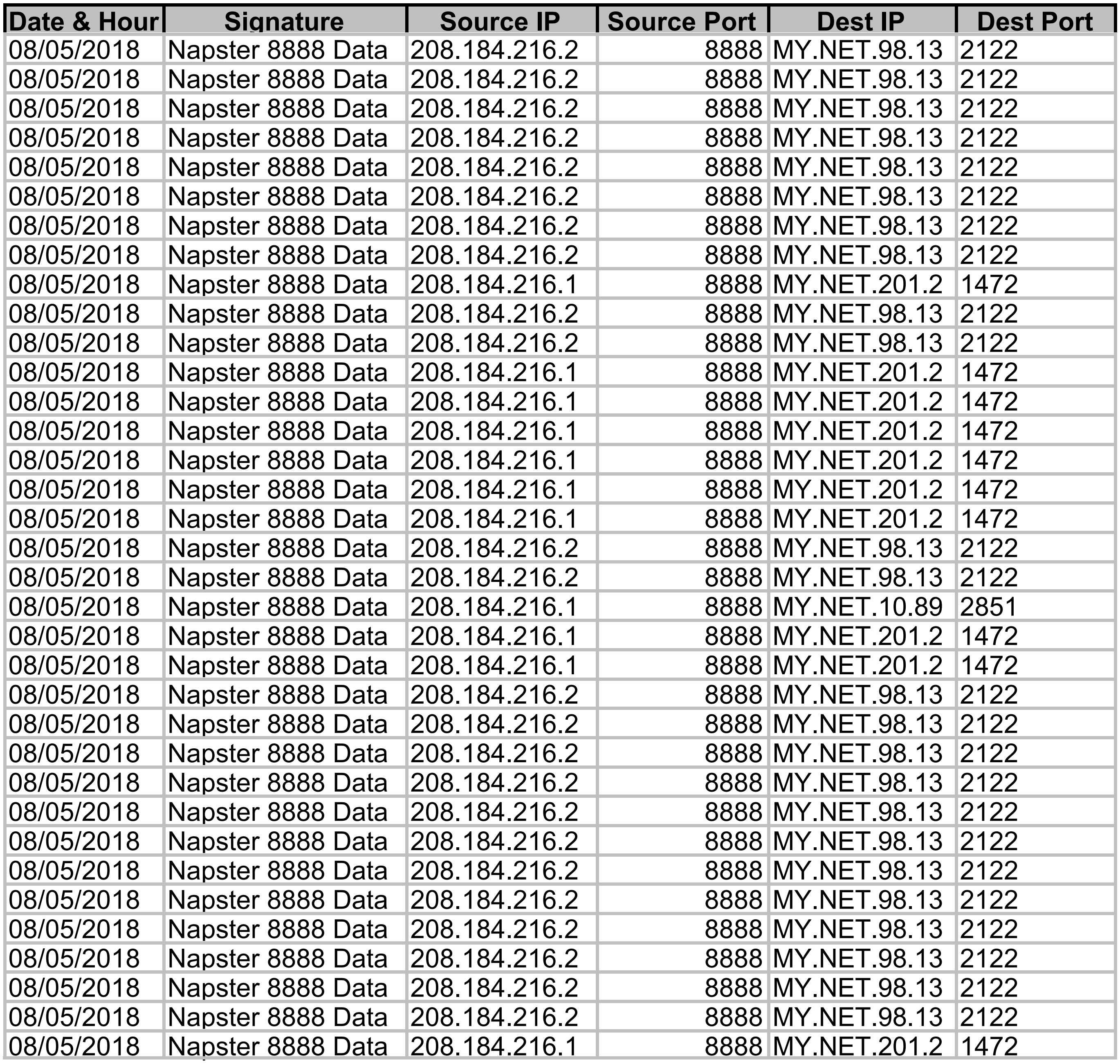
Month	Time	Source IP	Source	Dest IP	Dest	Proto	0x13
Jul 24	21:56:5	209.123.109.	4324	MY.NET.98.118	23	SYN	**S*****
Aug 5	13:37:2	209.138.185.	4218	MY.NET.253.114	23	SYN	**S*****
Jul 11	16:30:1	209.150.114.	38992	MY.NET.60.11	23	INVALIDA	***FR*A

Month	Time	Source IP	Source	Dest IP	Dest	Proto	0x13
Jun 27	8:45:57	212.253.21.1	36862	MY.NET.60.8	23	NOACK	21

1. By the above detects we can see that 208.18.8.16 has tried to get into and compromise MY.NET.181.37. The unusual part of this is the SYN-RESET bit set when initially trying the connection. There is the potential if the machine MY.NET.181.37 is offering telnet services then the source IP is trying various methods to compromise or obtain root shell through a buffer overflow on this machine, without seeing the raw data associated with these packets, at this time we are unable to nail down the real cause of this apparent attack.

2. This type of activity is very often used for reconnaissance and intelligence gathering or denial of service attacks. The attacker repetitively sends the SYN-Reset to the source host and waits for any type of reply. From these responses the attacker can gain valuable information for future attacks.

DETECT 8 Analyzed



Date & Hour	Signature	Source IP	Source Port	Dest IP	Dest Port
08/05/2018	Napster 8888 Data	208.184.216.2	8888	MY.NET.98.13	2122
08/05/2018	Napster 8888 Data	208.184.216.2	8888	MY.NET.98.13	2122
08/05/2018	Napster 8888 Data	208.184.216.2	8888	MY.NET.98.13	2122
08/05/2018	Napster 8888 Data	208.184.216.2	8888	MY.NET.98.13	2122
08/05/2018	Napster 8888 Data	208.184.216.2	8888	MY.NET.98.13	2122
08/05/2018	Napster 8888 Data	208.184.216.2	8888	MY.NET.98.13	2122
08/05/2018	Napster 8888 Data	208.184.216.2	8888	MY.NET.98.13	2122
08/05/2018	Napster 8888 Data	208.184.216.2	8888	MY.NET.98.13	2122
08/05/2018	Napster 8888 Data	208.184.216.1	8888	MY.NET.201.2	1472
08/05/2018	Napster 8888 Data	208.184.216.2	8888	MY.NET.98.13	2122
08/05/2018	Napster 8888 Data	208.184.216.2	8888	MY.NET.98.13	2122
08/05/2018	Napster 8888 Data	208.184.216.1	8888	MY.NET.201.2	1472
08/05/2018	Napster 8888 Data	208.184.216.1	8888	MY.NET.201.2	1472
08/05/2018	Napster 8888 Data	208.184.216.1	8888	MY.NET.201.2	1472
08/05/2018	Napster 8888 Data	208.184.216.1	8888	MY.NET.201.2	1472
08/05/2018	Napster 8888 Data	208.184.216.1	8888	MY.NET.201.2	1472
08/05/2018	Napster 8888 Data	208.184.216.1	8888	MY.NET.201.2	1472
08/05/2018	Napster 8888 Data	208.184.216.2	8888	MY.NET.98.13	2122
08/05/2018	Napster 8888 Data	208.184.216.2	8888	MY.NET.98.13	2122
08/05/2018	Napster 8888 Data	208.184.216.1	8888	MY.NET.10.89	2851
08/05/2018	Napster 8888 Data	208.184.216.1	8888	MY.NET.201.2	1472
08/05/2018	Napster 8888 Data	208.184.216.1	8888	MY.NET.201.2	1472
08/05/2018	Napster 8888 Data	208.184.216.2	8888	MY.NET.98.13	2122
08/05/2018	Napster 8888 Data	208.184.216.2	8888	MY.NET.98.13	2122
08/05/2018	Napster 8888 Data	208.184.216.2	8888	MY.NET.98.13	2122
08/05/2018	Napster 8888 Data	208.184.216.2	8888	MY.NET.98.13	2122
08/05/2018	Napster 8888 Data	208.184.216.2	8888	MY.NET.98.13	2122
08/05/2018	Napster 8888 Data	208.184.216.2	8888	MY.NET.98.13	2122
08/05/2018	Napster 8888 Data	208.184.216.2	8888	MY.NET.98.13	2122
08/05/2018	Napster 8888 Data	208.184.216.2	8888	MY.NET.98.13	2122
08/05/2018	Napster 8888 Data	208.184.216.2	8888	MY.NET.98.13	2122
08/05/2018	Napster 8888 Data	208.184.216.2	8888	MY.NET.98.13	2122
08/05/2018	Napster 8888 Data	208.184.216.2	8888	MY.NET.98.13	2122
08/05/2018	Napster 8888 Data	208.184.216.2	8888	MY.NET.98.13	2122
08/05/2018	Napster 8888 Data	208.184.216.1	8888	MY.NET.201.2	1472

1. As we can see in the above trace this appears to be a Napster connection to various host machines that are connected to MY.NET.XXX.XXX. I have only included a portion of the alerts for brevity purposes but it shows us a potential problem.

2. Napster is music at Internet speed, Napster is free and full of potential holes. It seems there is the possibility that multiple host machines on your network have Napster installed.

3. One of the problems with Napster is it allows the user to select a proxy to run on, this cause the problem because network administrators are usually local network admins. Specifically, they only thave the power to filter(prevent) packeets that leave your computer destined for some other specific set of computers. So when it appears they have blocked Napster, what they have actually done is we've shut of direct access to the napster servers. The problem is. It is easy to find a proxy that must be on a network that can directly access.

DETECT 9 Analyzed

Date & Hour	Min	Se	Signature	Source IP	Dest
06/28/2006	37	13	Tiny Fragments - Possible Hostile Activity	63.236.34.174	MY.NET.
07/11/2003	33	54	Tiny Fragments - Possible Hostile Activity	208.61.144.55	MY.NET
06/28/2006	35	13	Tiny Fragments - Possible Hostile Activity	63.236.34.174	MY.NET.
06/28/2006	35	13	Tiny Fragments - Possible Hostile Activity	63.236.34.174	MY.NET.
06/28/2006	35	13	Tiny Fragments - Possible Hostile Activity	63.236.34.174	MY.NET.
06/28/2006	37	13	Tiny Fragments - Possible Hostile Activity	63.236.34.174	MY.NET.
06/28/2006	37	13	Tiny Fragments - Possible Hostile Activity	63.236.34.174	MY.NET.

1. What we have in the about trace can be construed as having the potential to be harmful traffic.

2. A normal TCP header is a minimum of 20 byes in length, however a packet may be crafted so that these 20 bytes are fragmented in an attempt to bypass firewalls or intrusion detection systems. This is a form of reconnaissance. Although through all seen alerts/alarms, there is not a large amount of this type of traffic, its mere presence indicates a very interested visitor is using this recon for a specific query. Additional information can be found at http://www.sans.org/y2k/092000.htm

3. With many IP implementations it is possible to impose an unusually small fragment size on outgoing packets. If the fragment size is made small enough to force some of a TCP packet’s TCP header fields into the second fragment, filter rules that specify patterns for those fields will not match. If the filtering implementation does not enforce a minimum fragment size, a disallowed packet might be passed because it didn’t him a match in the filter.

DETECT 10 Analyzed

Date & Hour	Min	Sec	Signature	Source IP	Dest IP
06/29/2017	55	42	spp portscan portscan status from	MY.NET.1.3	
06/30/2006	33	38	spp portscan PORTSCAN DETECTED from	MY.NET.1.3	
06/30/2006	33	38	spp portscan portscan status from	MY.NET.1.3	
06/30/2006	33	39	spp portscan portscan status from	MY.NET.1.3	
06/30/2006	33	39	spp portscan portscan status from	MY.NET.1.3	
06/30/2010	23	46	spp portscan PORTSCAN DETECTED from	MY.NET.1.3	
06/30/2010	23	48	spp portscan portscan status from	MY.NET.1.3	
06/30/2010	32	32	spp portscan PORTSCAN DETECTED from	MY.NET.1.3	
06/30/2010	32	33	spp portscan portscan status from	MY.NET.1.3	
06/30/2015	40	44	spp_portscan PORTSCAN DETECTED from	MY.NET.1.3	

1. What we have here is MY.NET.1.3 doing portscans against ? Destination IP addresses. This is unusual unless a system administrator as why would one do portscans on your own network unless specifically looking for services operating on any host machince scanned.

2. There is also the potential that MY.NET.1.3 could be a compromised machine, if this is the case then an investigation into this needs to be done.

Summary Of Most Events

For the purpose of this Analyze this portion of the practical, I have placed below a general summary of most of the Alarms/Alerts. During work with Access at times it became difficult to keep track of all Alarms/Alerts, hence putting in a table for quick reference. This table was made with the assistance of fellow workers as we all searched through the detects. There are some detects that have been left, those that were were not as vital as what has been previously reported. Some of the ones previously reported are just to show the potential of security problems. There are/is more security alarms in this table than was reported but this was just to show the customer some of the problems on their Network.

Alarm/Alert	# of alarms
Possible wuftpd giac 000623	4
IDS246 misc large icmp	5
IDS127 telenet login incorrect	7
Site exec possible wu	8
Tiny fragments	9
Queso finger print	11
Napster client data	12
Sunrpc high port access	20
Nmap tcp ping	51
Null scan	99
Napster 7777 data	170
GIAC 218 VA CIRT port	190
SMB name wild card	208
GIAC 000218 VA CIRT Port	214
Napster 8888 data	323
Snmp public access	1080
IDS247 misc large upd	1170
Wingate 1080 attempt	2284
Attempted sun rpc high port	2311
Wingate 880 attempt	4214
Watchlist 222 NET	4795
Ping icmp time exceeded	6690
Spp	8181
Ping icmp dest unreachable	12313
Watchlist 220 ILISDNNET	13976
Syn/fin scan	20068

Additional Information

1. It seems there is a lot of scanning going on looking for backdoors that may be installed on the network. One of the main ones, is Port 31337, this happens to be used for Back Orifice. This has the potential to cause multiple problems if the user finds a machine he can connect to. There is a possiblity if connection is made using Back Orifice then the victim machine can cause potential damage to the inside network, also, the user who controls this machine may cause additional problems to MY.Net networks.

2. Going through all of the logs we have seen multiple attempts to either compromise machines or to literally control certain machines. I have left out the names(IP’s) of these machines as our intention here was to provide specific and or fairly generic ideas of the potential problems that reside with the company the owns MY.NET.XXX.XXX

3. Having been unable to do in-depth analysis of the logs as they do not show traffic data, a lot of these packets do not reveal additional information as to why this activity continues.

4. A lot of traffic targeted against this network is caused by script kiddies, if the firewall, router and IDS system are properly configured and maintained this will over come most of the potential problems of running services on the Internet.

5. One item of note, if there is the possibility of being able to obtain or look at the missing logs, or correlating the snort logs with the router/firewall logs this may be able to free up certain pieces of missing information.

6. Knowing nothing about the net structure of this company at times it becomes difficult to provide an indepth accurate report of the security structure of this company. One would need to look at the security policy and all of the IP addresses and host machines to know what is the potential for any rogue user to provide company information to the outside world whether intentially or by accidental mis-use of his host machine.

7. It was noted during going through the logs there are a number of restrictive addresses that are specifically targeting MY.NET.XXX.XXX. These sources addresses have specific rules built that guard/protect against them establishing a connection against MY.NET.XXX.XXX. This should continue and more addresses may have to be added to properly protect any malicious activity that originates from known IP Addresses.

8. Having seen all this information here are some of the things that now have to thought about.

   a. How will you secure the access?
   b. Do you have a need for multiple access point and central management?
   c. Do you have a need for internal network access control?
   d. Do you require Internet access control by time of day, site, content?
   e. Are you using un-authorized IP address on your

internal Network?

f. Do you have remote employees looking for encrypted access to your Intranet?

9. Once the above questions have been answered then your security policy can be written to reflect the changes needed to properly ensure the security of your network.

ASSIGNMENT 4- Analysis Process.

1. The process that was used to analyze the data in assignment 3 was time consuming and not very user friendly.

g. First step was to look at Snort logs- This didn't work and was extremely time consuming as there were multiple logs and correlating IP addresses would have worn out any computer screen or piece of paper.

h. Putting them all into a Word document seemed like a good idea, extremely difficult to do this as the cutting and pasting of txt would have taken a fair bit of manipulation to correlate all of the needed information.

i. Hence the need for a Dbase program to use this, the only one available that was free to use, (thank-you) Dept of National Defence was Microsoft Access.

j. Running this was simple yet time consuming, I was able to do various tables and queries to obtain the results that were presented in this paper.

2. There are various other means of analyze the data that was presented in these Snort logs. If time had not been an issue perl scrpts would have been written to fully analyze the data that was presented, a fuller and more comprehensible writing of data may have been able to be shown. It can also be sent over to a Linux/Unix machine to be able to grep and look for specific patterns of activity.

3. It would have been advantageous to be able to capture and play back any TCPDump traffic that these Snort logs represents. This will give a lot of information as to the packet structure/header/ and data, without it, this was not a full evaluation of the data presented.