GIAC
CERTIFICATIONS

Global Information Assurance Certification Paper

Copyright SANS Institute
Author Retains Full Rights

This paper is taken from the GIAC directory of certified professionals. Reposting is not permited without express written permission.

Interested in learning more?

Check out the list of upcoming events offering
"Network Monitoring and Threat Detection In-Depth (Security 503)"
at http://www.giac.org/registration/gcia

GIAC PRACTICAL

David Singer

SANS DARLING HARBOUR 2001

© SANS Institute 2000 - 2002, Author retains full rights.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

© SANS Institute 2000 - 2002, Author retains full rights.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

Assignment 1 – Network Detects

Detect No. 1

```
Snort portscan logs:

Mar 23 03:59:10 194.133.121.2:53  -> MY.NET.3.154:53 SYNFIN ******SF
Mar 23 03:59:10  194.133.121.2:53  -> MY.NET.3.155:53 SYNFIN ******SF
Mar 23 03:59:10 194.133.121.2:53  -> MY.NET.3.156:53 SYNFIN ******SF
Mar 23 03:59:10 194.133.121.2:53  -> MY.NET.3.157:53 SYNFIN ******SF
Mar 23 03:59:10 194.133.121.2:53  -> MY.NET.3.158:53 SYNFIN ***** *SF
```

Both Syn and Fin bits set tcp[13] = 3

Site: Site1 Host lookup: , Dates: 03/20/01 - 03/25/01 Pattern: tcp[13] = 3

```
03/20/01 21:25:59.985556 qitek.com.tw.ftp > MY.NET.3.154.ftp: SF
151998664:151998664(0) win 1028
03/20/01 21:26:00.005306 qitek.com.t  w.ftp > MY.NET.3.155.ftp: SF
151998664:151998664(0) win 1028
03/20/01 21:26:00.021521 qitek.com.tw.ftp > MY.NET.3.156.ftp: SF
151998664:151998664(0) win 1028
03/20/01 21:26:00.042222 qitek.com.tw.ftp > MY.NET.3.157.ftp: SF
151998664:151998664(0) win 1028
03/20/01 21:26:00.058559 qitek.com.tw.ftp > MY.NET.3.158.ftp: SF
151998664:151998664(0) win 1028
03/22/01 08:18:10.399694 www.magnitude.co.uk.511 > MY.NET.3.154.511: SF
352603438:352603438(0) win 1028
03/22/01 08:18:10.420353 www.magnitude.co.uk.511 > MY.NE  T.3.155.511: SF
352603438:352603438(0) win 1028
03/22/01 08:18:10.439075 www.magnitude.co.uk.511 > MY.NET.3.156.511: SF
352603438:352603438(0) win 1028
03/22/01 08:18:10.460880 www.magnitude.co.uk.511 > MY.NET.3.157.511: SF
352603438:352603438(0) win 1028
03/22/01 08:18:10.480851 www.magnitude.co.uk.511 > MY.NET.3.158.511: SF
352603438:352603438(0) win 1028
03/22/01 18:15:38.773171 211.21.102.12.ssh > MY.NET.3.154.ssh: SF
1583399634:1583399634(0) win 1028
03/22/01 18:15:38.787805 211.21.102.12.ssh > MY.NET.  3.155.ssh: SF
1583399634:1583399634(0) win 1028
03/22/01 18:15:38.799156 211.21.102.12.ssh > MY.NET.3.156.ssh: SF
1583399634:1583399634(0) win 1028
03/22/01 18:15:38.826821 211.21.102.12.ssh > MY.NET.3.157.ssh: SF
1583399634:1583399634(0) win 1028
03/22/01  18:15:38.872483 211.21.102.12.ssh > MY.NET.3.158.ssh: SF
1583399634:1583399634(0) win 1028
03/23/01 02:59:10.257999 194.133.121.2.domain > MY.NET.3.154.domain: SF
596888598:596888598(0) win 1028
03/23/01 02:59:10.281897 194.133.121.2.domain > MY.NET.3.155   .domain: SF
596888598:596888598(0) win 1028
03/23/01 02:59:10.306984 194.133.121.2.domain > MY.NET.3.156.domain: SF
596888598:596888598(0) win 1028
03/23/01 02:59:10.319478 194.133.121.2.domain > MY.NET.3.157.domain: SF
596888598:596888598(0) win 1028
03/23/01 02:59:10.343419 194.133.121.2.domain > MY.NET.3.158.domain: SF
596888598:596888598(0) win 1028
```


© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

1. Source of Trace.

My network.

2. Detect was generated by:

Initial detect generated by Snort. I then used the search capability of shadow to look for more evidence of this attack.

tcpdump filter:
Both Syn and Fin bits set tcp[13] = 3

3. Probability the source address was spoofed:

Unlikely
With a SYN/FIN scan the attacker relies on the receiving the replies for their reconnaissance so a source address is u sually not spoofed. However, here we see the same pattern with 4 different source addresses over the period of 4 days. If this is one attacker then maybe the sourced address is spoofed. If is 4 different attackers then the source address is probably not s poofed. I looked through my tcpdump logs for evidence of source routing but did not find any so would assume that I am dealing with 4 different attackers.

4. Description of attack:

The following was noticed:

SF set
Source Port and Destination Port the sam e.
Same TCP sequence number used
Attack originated from four different source addresses, from different countries.

IP Header greater than 20 bytes in Byte 0
IP option field with value of x83 or x89 in Byte 20 of the IP Header

tcpdump filter:

ip[0] & 0x0f > 5 and (ip[20] =0x83 or ip[20] =0x89)

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

5. Attack mechanism:

This is a SYN/FIN probe probably looking for active TCP ports. A SYN/FIN is sent to elicit the expected response of a RST ACK from an inactive port. Windows NT/95/98 would respond with a FIN/AC K for an inactive port.

	Operating System	Normal Response
closed port	Unix	RST ACK
closed port	Windows NT/95/98	FIN ACK
open port	Unix	no response
open port	Windows NT/95/98	FIN ACK

The intention is to find open ports behind a firewall. The techniq ue of using SF and well known source ports is to try and fool some packet filtering firewalls which don't keep state.
(Ref 1)

6. Correlations:

This is a well known use of either hping or nmap.

http://www.whitehats.com/info/IDS441

7. Evidence of active targ eting:

General scan of entire network
All addresses on network were probed

8. Severity:

(Critical + Lethal) – (System + Net Countermeasures) = Severity
(3 + 2) – (3 + 4) = -2

Critical = 3 – a mix of hosts
Lethal = 2 – reconnaissance
System = 3 – some test systems and older operating systems
Net Countermeasures = 4 only mail DNS let through the firewall

© SANS Institute 2000 - 2002, Author retains full rights.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

9. Defensive recommendation:

Use an IDS to filter for SF flags set.
Both Syn and Fin bits set tcp[13] = 3
Monitor firewall logs.
Check firewall configurat ion for open ports.

10. Multiple choice test question:

```
Mar 23 03:59:10 194.133.121.2:53  -> MY.NET.3.154:53 SYNFIN ******SF
```

What would be the normal response from a unix host if the above stimulus was sent to an open port?

a) No response
b) FIN ACK
c) RST ACK
d) SYN

Answer: a

© SANS Institute 2000 - 2002, Author retains full rights.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

Detect No. 2

Gauntlet Firewall Logs:

Apr 9 19:04:26 gwsyd kernel: securityalert: tcp if=exp1 from 216.229.235.156:2666 to MY.NET.3.154 on unserved port 27374
Apr 9 19:04:26 gwsyd kernel: securityalert: tcp if=exp1 from 216.229.235.156:2668 to MY.NET.3.154 on unserved port 12345
Apr 9 19:04:26 gwsyd kernel: securityalert: tcp if=exp1 from 216.229.235.156:2689 to MY.NET.3.154 on unserved port 139
Apr 9 19:04:27 gwsyd kernel: securityalert: tcp if=exp1 from 216.229.235.156:2666 to MY.NET.3.15 4 on unserved port 27374
Apr 9 19:04:27 gwsyd kernel: securityalert: tcp if=exp1 from 216.229.235.156:2668 to MY.NET.3.154 on unserved port 12345
Apr 9 19:04:27 gwsyd kernel: securityalert: tcp if=exp1 from 216.229.235.156:2689 to MY.NET.3.154 on unserve d port 139
Apr 9 19:04:28 gwsyd kernel: securityalert: tcp if=exp1 from 216.229.235.156:2668 to MY.NET.3.154 on unserved port 12345
Apr 9 19:04:28 gwsyd kernel: securityalert: tcp if=exp1 from 216.229.235.156:2666 to MY.NET.3.154 on unserved port 27374
Apr 9 19:04:28 gwsyd kernel: securityalert: tcp if=exp1 from 216.229.235.156:2689 to MY.NET.3.154 on unserved port 139

© SANS Institute 2000 - 2002, Author retains full rights.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

Shadow Search Tool:

```
19:03:42.386597 host -00010366832C.public.southern.edu.2666 >
MY.NET.3.154.asp: S 803120325:803120325(0) win 16384    (DF)
19:03:42.386832 MY.NET.3.154.asp > host  -
00010366832C.public.southern.edu.2666: R 0:0(0) ack 803120326 win 0
19:03:42.393390 host -00010366832C.public.southern.edu.2668 >
MY.NET.3.154.12345: S 803180636:803180636(0) win 16384  (DF)
19:03:42.393574 MY.NE T.3.154.12345 > host -
00010366832C.public.southern.edu.2668: R 0:0(0) ack 803180637 win 0
19:03:42.400402 host -00010366832C.public.southern.edu.2689 >
MY.NET.3.154.netbios -ssn: S 803226259:803226259(0) win 16384  (DF)
19:03:42.400596 MY.NET.3.154.netbios  -ssn > host-
00010366832C.public.southern.edu.2689: R 0:0(0) ack 803226260 win 0
19:03:43.256420 host -00010366832C.public.southern.edu.2666 >
MY.NET.3.154.asp: S 803120325:803120325(0) win 16384  (DF)
19:03:43.256618 MY.NET.3.154.asp > host  -
00010366832C.public .southern.edu.2666: R 0:0(0) ack 803120326 win 0
19:03:43.372170 host -00010366832C.public.southern.edu.2668 >
MY.NET.3.154.12345: S 803180636:803180636(0) win 16384  (DF)
19:03:43.372369 MY.NET.3.154.12345 > host  -
00010366832C.public.southern.edu.2668: R 0:  0(0) ack 803180637 win 0
19:03:43.392125 host -00010366832C.public.southern.edu.2689 >
MY.NET.3.154.netbios -ssn: S 803226259:803226259(0) win 16384  (DF)
19:03:43.392322 MY.NET.3.154.netbios  -ssn > host-
00010366832C.public.southern.edu.2689: R 0:0(0) ack 803  226260 win 0
19:03:44.250151 host -00010366832C.public.southern.edu.2668 >
MY.NET.3.154.12345: S 803180636:803180636(0) win 16384  (DF)
19:03:44.250347 MY.NET.3.154.12345 > host  -
00010366832C.public.southern.edu.2668: R 0:0(0) ack 803180637 win 0
19:03:44.25 9735 host -00010366832C.public.southern.edu.2666 >
MY.NET.3.154.asp: S 803120325:803120325(0) win 16384  (DF)
19:03:44.259916 MY.NET.3.154.asp > host  -
00010366832C.public.southern.edu.2666: R 0:0(0) ack 803120326 win 0
19:03:44.266746 host -00010366832C.publi c.southern.edu.2689 >
MY.NET.3.154.netbios -ssn: S 803226259:803226259(0) win 16384  (DF)
19:03:44.266916 MY.NET.3.154.netbios  -ssn > host-
00010366832C.public.southern.edu.2689: R 0:0(0) ack 803226260 win 0
```

1. Source of Trace.

My network.

2. Detect was generat ed by:

Initial detect generated by Gauntlet Firewall. I then used the search capability of shadow to look for more evidence of this attack.

3. Probability the source address was spoofed:

Unlikely

Attacker's address resolves to host -00010366832C.public. southern.edu
When using Trojans attacker normally wants a response.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

4. Description of attack:

Connection attempts to suspicious unserved ports on the firewall were observed.

Port 12345 – Trojan - netbus

Port 27374 – Trojan - subseven
worm-ramen-asp

5. Attack mechanism:

The attacker is looking for Trojans, either netbus, subseven.
He sends a SYN to attempt to connect but the firewall replies with a RESET. The connection is then dropped.

6. Correlations:

Subseven:
http://www.whitehats.com/info/IDS279
Netbus:
http://www.whitehats.com/info/IDS401

7. Evidence of active targeting:

Only one address was targeted. Logs did not show any further activity, would assume that attacker gave up and moved on.

8. Severity:

(Critical + Lethal) – (System + Net Countermeasure s) = Severity
(5 + 5) – (3 + 5) = 2

Critical = 5 – firewall
Lethal = 5 – looking for compromised host
System = 3 – older operating system, some patches missing.
Net Countermeasures = 5 – firewall effective in blocking attack

© SANS Institute 2000 - 2002, Author retains full rights.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

9. Defensive recommendation:

Known signatures - Snort or other IDS can detect.
Monitor firewall logs.

10. Multiple choice test question:

```
19:03:42.393390  216.229.235.156 .2668 > MY.NET.3.154.12345: S
803180636:803180636(0) win 16384  (DF)

19:03:42.393574 MY.NET.3.154.12345 >  216.229.235.1 56.2668: R 0:0(0) ack
803180637 win 0
```

TCP port 12345 is a signature for the Trojan netbus. From the above trace and given that there is no other traffic logged to port 12345 would you conclude that;

a) The host MY.NET.3.154 is compromised
b) The host 216.229.235.156 is compromised
c) The host 216.229.235.156 is not compromised
d) The host MY.NET.3.154 is not compromised

Answer: d

© SANS Institute 2000 - 2002, Author retains full rights.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

Detect No. 3

Gauntlet Firewall Logs:

Apr 10 05:21:16 gwsyd kernel: securityalert: tcp if=exp1 from 202.103.198.146:1677 to MY.NET.3.154 on unserved port 3128
Apr 10 05:21:16 gwsyd kernel: securityalert: tcp if=exp1 from 202.103.198.146:1680 to MY.NET.3.155 on unserved port 3128
Apr 10 05:21:17 gwsyd kernel: securityalert: tcp if=exp1 from 202.103.198.146:1677 to MY.NET.3.154 on unserved po rt 3128
Apr 10 05:21:17 gwsyd kernel: securityalert: tcp if=exp1 from 202.103.198.146:1680 to MY.NET.3.155 on unserved port 3128
Apr 10 05:21:17 gwsyd kernel: securityalert: tcp if=exp1 from 202.103.198.146:1677 to MY.NET.3.154 on unserved port 3128
Apr 10 05:21:17 gwsyd kernel: securityalert: tcp if=exp1 from 202.103.198.146:1680 to MY.NET.3.155 on unserved port 3128

Shadow Search Tool:

```
05:20:30.312193 202.103.198.146 > MY.NET.3.154: icmp: echo request
05:20:30.312367 MY.NET.3.154 > 202.103.198.146: icmp  : echo reply
05:20:30.327664 202.103.198.146 > MY.NET.3.156: icmp: echo request
05:20:30.337168 202.103.198.146 > MY.NET.3.155: icmp: echo request
05:20:30.337311 MY.NET.3.155 > 202.103.198.146: icmp: echo reply
05:20:30.346070 202.103.198.146 > MY.NET.3.1  57: icmp: echo request
05:20:30.346142 MY.NET.3.157 > 202.103.198.146: icmp: echo reply
05:20:30.355061 202.103.198.146 > MY.NET.3.158: icmp: echo request
05:20:31.891993 202.103.198.146.1675 > MY.NET.3.154.webcache: S
3202866206:3202866206(0) win 16384  ( DF)
05:20:31.892219 MY.NET.3.154.webcache > 202.103.198.146.1675: S
1859372819:1859372819(0) ack 3202866207 win 8760  (DF)
05:20:31.898992 202.103.198.146.1676 > MY.NET.3.154.www: S
3202922480:3202922480(0) win 16384  (DF)
05:20:31.899159 MY.NET.3.154.www  > 202.103.198.146.1676: S
1859407025:1859407025(0) ack 3202922481 win 8760  (DF)
05:20:31.905900 202.103.198.146.1677 > MY.NET.3.154.3128: S
3202967753:3202967753(0) win 16384  (DF)
05:20:31.906089 MY.NET.3.154.3128 > 202.103.198.146.1677: R 0:0(0) ack
3202967754 win 0
05:20:31.912900 202.103.198.146.1678 > MY.NET.3.155.webcache: S
3203002685:3203002685(0) win 16384  (DF)
05:20:31.913078 MY.NET.3.155.webcache > 202.103.198.146.1678: S
1859451739:1859451739(0) ack 3203002686 win 8760  (DF)
05:20:31.919926 20 2.103.198.146.1679 > MY.NET.3.155.www: S
3203068106:3203068106(0) win 16384  (DF)
05:20:31.920105 MY.NET.3.155.www > 202.103.198.146.1679: S
1859496884:1859496884(0) ack 3203068107 win 8760  (DF)
05:20:31.926757 202.103.198.146.1680 > MY.NET.3.155.3128: S
3203104498:3203104498(0) win 16384  (DF)
05:20:31.926945 MY.NET.3.155.3128 > 202.103.198.146.1680: R 0:0(0) ack
3203104499 win 0
05:20:31.933756 202.103.198.146.1681 > MY.NET.3.157.webcache: S
3203164948:3203164948(0) win 16384  (DF)
05:20:31.933799 MY.NET .3.157.webcache > 202.103.198.146.1681: R 0:0(0) ack
3203164949 win 0
05:20:31.940768 202.103.198.146.1682 > MY.NET.3.157.www: S
3203222910:3203222910(0) win 16384  (DF)
```


© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

1. Source of Trace.

My network.

2. Detect was generated by:

Initial detect generated by Gauntlet Firewall. I then used the search capability of shadow to look for more evidence of this attack.

3. Probability the source address was spoofed:

Unlikely

Attacker's address is found in apnic whois:

```
inetnum:     202.103.192.0  - 202.103.255.255
netname:     CHINANET -GX
descr:       CHINANET Guangxi province network
descr:       Data Communication Division
descr:       China Telecom
country:     CN
admin-c:     CH93 -AP
tech-c:      DZ7 -AP
mnt-by:      MAINT -CHINANET
mnt-lower:   MAINT -CHINANET -GX
changed:     hostmaster@ns.chinanet.cn.net 20000101
source:      APNIC
```

4. Description of attack:

Attacker pings all addresses on our network looking for replies. He then attempts connection port 80 (www) and port 3128 (Squid Proxy Server) on all hosts that reply to ping requests.

5. Attack mechanism:

The attacker is looking web servers or Squid proxy servers. If successful he may then continue to gather information regarding OS and versions and then target a server with known vulnerabilities. Attacker may als o be trying to use the Squid proxy as a method of hiding his address when attacking other sites.

Another explanation is that this might be the Ring Zero scan which looks for proxies at ports 80, 8080 and 3128. I don't think this is the case here as the at tack was preceded by pings. Also, port 8080 was not scanned.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

6. Correlations:

Ring Zero

http://www.sans.org/newlook/resources/IDFAQ/ring_zero.htm

7. Evidence of active targeting:

Packets were sent to a range of addresses. Not active targeting at this stag e.

8. Severity:

(Critical + Lethal) – (System + Net Countermeasures) = Severity
(3 + 3) – (3 + 5) = -2

Critical = 3 – a range of hosts
Lethal = 3 – may try a vulnerability
System = 3 – older operating system, some patches missing.
Net Countermeasures = 5 – firewall effective in blocking attack

9. Defensive recommendation:

If not used block ports 80, 8080 and 3128 at the router and/or firewall.
Block traffic from 202.103.192.0 - 202.103.255.255
Monitor firewall logs.

10. Multiple choice test question:

```
05:20:31.891993 202.103.198.146.1675 > MY.NET.3.154.webcache: S
3202866206:3202866206(0) win 16384  (DF)

05:20:31.892219 MY.NET.3.154.webcache > 202.103.198.146.1675: S
1859372819:1859372819(0) ack 3202866207 win 8760  (DF)

05:20:31.898992 202.103.198.146.1676 >  MY.NET.3.154.www: S
3202922480:3202922480(0) win 16384  (DF)

05:20:31.899159 MY.NET.3.154.www > 202.103.198.146.1676: S
1859407025:1859407025(0) ack 3202922481 win 8760  (DF)
```

What is the best answer for the intention of the attacker from the above trace ?

a) This could be a Ring Zero scan
b) The attacker could be scanning for web proxies.
c) The attacker could be scanning for web servers or squid proxies.
d) Any of the above is true

Answer: d

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

Detect No. 4

Gauntlet Firewall Logs:

Apr 10 18:42:57 gwsyd kernel: sec urityalert: tcp if=exp1 from 38.136.180.4:1319 to MY.NET.3.154 on unserved port 515
Apr 10 18:42:57 gwsyd kernel: securityalert: tcp if=exp1 from 38.136.180.4:1320 to MY.NET.3.155 on unserved port 515

Shadow Search Tool:

```
18:42:11.913522 38.136.180.4.1319  > MY.NET.3.154.printer: S
3557083570:3557083570(0) win 32120  (DF)
18:42:11.913748 MY.NET.3.154.printer > 38.136.180.4.1319: R 0:0(0) ack
3557083571 win 0
18:42:11.922445 38.136.180.4.1320 > MY.NET.3.155.printer: S
3559623777:3559623777(0) win 32120  (DF)
18:42:11.922626 MY.NET.3.155.printer > 38.136.180.4.1320: R 0:0(0) ack
3559623778 win 0
18:42:11.930249 38.136.180.4.1321 > MY.NET.3.156.printer: S
3557246736:3557246736(0) win 32120  (DF)
18:42:11.939313 38.136.180.4.1322 > MY.NET.3.157.printer: S
3560716241:3560716241(0) win 32120  (DF)
18:42:11.939391 MY.NET.3.157.printer > 38.136.180.4.1322: R 0:0(0) ack
3560716242 win 0
18:42:11.947674 38.136.180.4.1323 > MY.NET.3.158.printer: S
3551146915:3551146915(0) win 32120  (DF)
18:42:14.030138 38.136.180.4.132 3 > MY.NET.3.158.printer: S
3551146915:3551146915(0) win 32120  (DF)
18:42:15.827200 38.136.180.4.1321 > MY.NET.3.156.printer: S
3557246736:3557246736(0) win 32120 (DF)
```

© SANS Institute 2000 - 2002. Author retains full rights.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

1. Source of Trace.

My network.

2. Detect was generated by:

Initial detect generated by Gauntlet Firewall. I then used the search capability of shadow to look for more evidence of this attack.

3. Probability the source address was spoofed:

Unlikely

Attacker would require a response to the scan.

```
[whois.arin.net]
Performance Systems Intern ational (NET -PSINETA)
   510 Huntmar Park Drive
          Herndon, VA  22070
   US

   Netname: PSINETA
   Netblock: 38.0.0.0  - 38.255.255.255
   Maintainer: PSI

   Coordinator:
      PSINet,Inc.  (PSI -NISC-ARIN)  hostinfo@psi.com
      (518) 283 -8860
```

4. Description of attack:

Attacker attempts to connect to port 515 on our hosts.

5. Attack mechanism:

The attacker may be looking for vulnerabilities for the LPR service.
The LPRng port on BSD and Linux, has a potential vulnerability which may allow root compromise.

© SANS Institute 2000 - 2002, Author retains full rights.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

6. Correlations:

Alert: Increased probes
to TCP port 515
Posted: 14:00 November 20, 2000

http://www.sans.org/newlook/alerts/port515.htm

7. Evidence of active targeting:

Packets were sent to a range of addresses. Not active targeting at this stage.

8. Severity:

(Critical + Lethal) – (System + Net Countermeasures) = Severity
(3 + 5) – (3 + 5) = 0

Critical = 3 – a range of hosts
Lethal = 5 – potential root access
System = 3 – older operating system, some patches missing.
Net Countermeasures = 5 – firewall effective in blocking attack

9. Defensive recommendation:

If not used block port 515 at the router and/or firewall.
Upgrade to latest LPR version

10. Multiple choice test question:

Apr 10 18:42:57 gwsyd kernel: securityalert: tcp if=exp1 from 38.136.180 .4:1319 to MY.NET.3.154 on unserved port 515
Apr 10 18:42:57 gwsyd kernel: securityalert: tcp if=exp1 from 38.136.180.4:1320 to MY.NET.3.155 on unserved port 515

The above is a trace from a Gauntlet firewall. Which of the following is true:

a) Port 515 is not listening on address MY.NET.3.154
b) Port 515 is not listening on address 38.136.180.4
c) Port 515 is not listening on the firewall
d) Port 1319 is not listening on the firewall

Answer: c

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

Detect No. 5

Shadow:
Site: Site1 - Date: Apr12 - EST: 15:00.

```
203.50.1 .12 > MY.NET.3.154
15:06:50.657731 host.badguy.net.43675 > MY.NET.3.154.33453: udp 12 [ttl 1]
15:06:50.688011 host.badguy.net.43675 > MY.NET.3.154.33454: udp 12 [ttl 1]
15:06:50.715540 host.badguy.net.43675 > MY.NET.3.154.33455: udp 12 [ttl 1]
```

Shadow Search Tool:

```
15:06:50.657731 host.badguy.net.43675 > MY.NET.3.154.33453: udp 12 [ttl 1]
15:06:50.657959 MY.NET.3.154 > host.badguy.net: icmp: MY.NET.3.154 udp port
33453 unreachable
15:06:50.688011 host.badguy.net.43675 > MY.NET.3.154.33454: udp 12 [ttl 1]
15:06:50.688211 MY.NET.3.154 > host.badguy.net: icmp: MY.NET.3.154 udp port
33454 unreachable
15:06:50.715540 host.badguy.net.43675 > MY.NET.3.154.33455: udp 12 [ttl 1]
15:06:50.715734 MY.NET.3.154 > host.badguy.net: icmp: MY.NET.3.154 udp port
33455 unreach able
15:07:20.731108 host.badguy.net > MY.NET.3.154: icmp: echo request
15:07:20.731370 MY.NET.3.154 > host.badguy.net: icmp: echo reply
15:07:21.746244 host.badguy.net > MY.NET.3.154: icmp: echo request
15:07:21.746418 MY.NET.3.154 > host.badguy.net: icmp  : echo reply
15:07:34.373486 host.badguy.net.4415 > MY.NET.3.154.smtp: S
3295563626:3295563626(0) win 16384  (DF) [tos 0x10]
15:07:34.373716 MY.NET.3.154.smtp > host.badguy.net.4415: S
2832985901:2832985901(0) ack 3295563627 win 8760  (DF)
15:07:34.399478  host.badguy.net.4415 > MY.NET.3.154.smtp: . ack 2832985902
win 17520 (DF) [tos 0x10]
15:07:34.481880 MY.NET.3.154.smtp > host.badguy.net.4415: P
2832985902:2832985947(45) ack 3295563627 win 8760 (DF)
15:07:34.531248 host.badguy.net.4415 > MY.NET.3.154.smtp  : . ack 2832985947
win 17520 (DF) [tos 0x10]
15:07:45.813134 host.badguy.net.4415 > MY.NET.3.154.smtp: P
3295563627:3295563642(15) ack 2832985947 win 17520 (DF) [tos 0x10]
15:07:45.813446 MY.NET.3.154.smtp > host.badguy.net.4415: P
2832985947:2832985984(37 ) ack 3295563642 win 8760 (DF)
15:07:45.930929 host.badguy.net.4415 > MY.NET.3.154.smtp: . ack 2832985984
win 17520 (DF) [tos 0x10]
```

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

Analysis of tcpdump using snort:
=+

04/12-15:08:36.606749 203.50.1.12:4415 -> MY.NET.3.154:25
TCP TTL:58 TOS:0x10 ID:48492 IpLen:20 DgmLen:69 DF
AP Seq: 0xC46E4F99 Ack: 0xA8DBEFA5 Win: 0x4470 TcpLen: 20
52 43 50 54 20 54 4F 3A 20 62 72 79 63 65 RCPT TO: bryce

=+ =+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

04/12-15:08:36.608060 MY.NET.3.154:25 -> 203.50.1.12:4415
TCP TTL:64 TOS:0x0 ID:51534 IpLen:20 DgmLen:111 DF
AP Seq: 0xA8DBEFA5 Ack: 0xC46E4FB6 Win: 0x2238 TcpLen: 20
35 35 30 20 4D 61 69 6C 62 6F 78 20 75 6E 550 Mailbox un

=+

1. Source of Trace.

My network.

2. Detect was generated by:

Initial detect generated by Shadow. I then used the search capability of shadow to look for more evidence of this atta ck. I then used snort to further analyze the tcpdump file and examine the payload.

3. Probability the source address was spoofed:

Very unlikely

Address of attacker resolved to a well known unix host. Address has been sanitized to host.badguy.net

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

4. Description of attack:

We see three UDP packets, 12 bytes each, with ttl 1 to ports 33453, 33454 and 33455.
We then see two echo requests and replies.
We then see a connection to the host on port 25 with data being transferred.

5. Attack mechanism:

The three UDP packets are actually traceroutes from a unix host. This explains the port numbers and the small ttl.
Further analysis of payload using snort on the tcpdump file reveals that the attacker is attempting to relay mail through the server.

6. Correlations:

Unauthorized mail relay is a well known exploit

http://info.internet.isi.edu/in -notes/rfc/files/rfc2505.txt
http://www.whitehats.com/info/IDS249

7. Evidence of active targeting:

Yes, host at secondary mx record was targeted.

8. Severity:

(Critical + Lethal) – (System + Net Countermeasures) = Severity
(4 + 3) – (3 + 2) = 2

Critical = 4 – mail server
Lethal = 3 – mail relay
System = 3 – older operating system, some patches missing.
Net Countermeasures = 2 – firewall will let smtp through relies on mail server

© SANS Institute 2000 - 2002, Author retains full rights.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

9. Defensive recommendation:

Check for anti-relay and anti-spoofing rules on mail server.

10. Multiple choice test question:

```
15:06:50.657731 host.badguy.net.43675 > MY.NET.3.154.33453: udp 12 [ttl 1]
15:06:50.688011 host.badguy.net.43675 > MY.NET.3.154.33454:   udp 12 [ttl 1]
15:06:50.715540 host.badguy.net.43675 > MY.NET.3.154.33455: udp 12 [ttl 1]
```

The small ttl in the above trace suggests that this may be:

a) a UDP port scan
b) packet craft
c) a traceroute
d) a ping

Answer: c

© SANS Institute 2000 - 2002, Author retains full rights.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

References:

1. Hacking Exposed: Network Secu rity Secrets and Solutions, Second Edition
   Joel Scambray, Stuart Mcclure, Geirge Kurtz
   Osborne/MxGraw-Hill
   p 44, 62, 471

© SANS Institute 2000 - 2002, Author retains full rights.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

Assignment 2 – Describe the State of Intrusion Detection

Web Site Vandalism

Introduction

IDC research for security breaches experi enced by Australian Enterprises in 2000 lists website vandalism as the least experienced against other attacks but the one that which most IT Managers are worried about. (1)

Website vandalism is equivalent to spraying graffiti over a wall. The difference is the potential cost of the damage is much greater. Even though no data may be lost and the website can easily be restored the true damage is to the reputation of the company who's web site is attacked. Along with the huge embarrassment for the company targeted, consumer confidence in the company will be lost and the potential damage may be in the millions.

Attackers delight in defacing websites, especially the well known ones. They accomplish their task swiftly and then leave to gloat. The media loves a good website vandalism story. Make sure that you know you website has been vandalized before you read about it in the papers.

A quick scans of sites such as attrition and project gamma that report on web site defacing reveal web servers running Microsoft IIS followed by Apache as the most targeted. "Hackers defaced a total of about 5000 web sites in the past 12 months. All the sites had firewalls" (2)

So why am I writing this paper? To illustrate that web site vandalism can easily be executed and hopeful ly prevented. This paper will analyze web site vandalism in the context of intrusion detection. It will use some reported incidents as examples. Describe how the attack was executed. Suggest how the attack may have been prevented or suggest how the attack may be have been detected by an IDS or other means.

© SANS Institute 2000 - 2002 Author retains full rights.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

Example 1: Microsoft New Zealand Defaced

OnTuesday, January 23 2001 the Web site for Microsoft New Zealand was defaced by the group Prime Suspectz (2). The home page was replaced by a message that stated "Oh!! What's happened? Another Microsoft was hacked?"

The hackers defaced the New Zealand site by using the Unicode exploit, which relies on vulnerabilities in IIS 4 and 5. The vulnerability allows a malicious user to execute operating system commands outside of the webroot through the web server.

You can detect whether a site is vulnerable to this attack by using the following URL:

http://www.mydomain.com/msadc/..%c0%af../..%c0%af../..%c0%af../winnt/system32/cmd.exe?/c+dir+c: \

This vulnerability can be detected by Snort using:

alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS 80 (msg:"Unicode exploit"; content:"/msadc|C0AF|"; nocase;)

CVE-1999-0874

Patches:

http://www.microsoft.com/windows2000/downloads/critical/q269862/default.asp

© SANS Institute 2000 - 2002, Author retains full rights.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

Example 2: Internet worm squirms into Linux servers (8)

Many high profile sites including NASA were defaced by an Internet worm that targets Redhat servers running version 6.2 or 7.0.

On each Web page the worm hits the main page is replaced with the message: "Hackers loooooooooooove noodles," signed by the "RameN Crew."

“When trying to infect Red Hat 6.2 systems, the worm will use the RPC.statd and wu -FTP flaws. The worm attempts to compromise Red Hat 7.0 systems by swamping the error logging function of the server's printer service with data” (8)

Snort Signature for WU-FTP:

alert TCP $EXTERNAL any -> $INTERNAL 21 (msg: "IDS458/ftp -wuftp260-tf8"; flags: A+; content: "|31C0 31DB 31C9 B046 CD80 31C0 31DB 43 89D941 B03F CD80|";)

http://www.whitehats.com/info/IDS458

Snort Signature for RPC-STATDX:

alert TCP $EXTERNAL any -> $INTERNAL any (msg: "IDS442/rpc -statdx-exploit"; flags: A+; content: "/bin|c74604|/sh";)

http://www.whitehats.com/info/IDS442
CVE-2000-0666

Patches:
wu-ftp:
http://www.redhat.com/support/errata /RHSA-2000-039-02.html
rpc-statdx:
http://www.redhat.com/support/errata/RHSA -2000-043-03.html

© SANS Institute 2000 - 2002, Author retains full rights.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

Example 3: They just did it… Nike.com site defaced by anti-capitalists (9)

(06/22/2000)

This site was running Netscape -Enterprise server but the attack was exec uted by hijacking the site's DNS entry rather than a direct hack at the server.
"The defacement included the message "'Global Justice is Coming - Prepare Now!" and a link to the website of the Australia -based "S11 alliance", an organization that is preparing protests against the ill effects of globalization at the World Economic Forum" (9).

Although it is not known exactly how this attack was executed it seems likely that the attackers managed to change the DNS entry for nike.com. They would have achieved this by fooling Network Solutions (the domain registrar) that they were authorized to make the change. Network Solutions offers three authorization schemes. The weakest is the "Mail From" authentication where all that is checked is the official contact lis ted in the database. Other methods are crypt -pw authentication that uses a secret password and PGP encryption.

© SANS Institute 2000 - 2002, Author retains full rights.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

Example 4: Apache.org *d*efaced (11)

A group of hackers gained root access to the machine running www.apache.org, and changed the main page to show a 'Powered by Microsoft BackOffice' logo instead of the default 'Powered by Apache' logo (the feather).

The vulnerability was caused by the directory structure under the FTP server ftp://ftp.apache.org being mapped to a path that was accessible from the Apache web server. This made if possible to upload PHP files to the FTP server that could then be executed from the web. PHP is a scripting language that is embedded in HTML. Once this access was achieved bindshell, a backdoor program was uploaded and executed. This program gives telnet access through TCP port 65533 but only with “nobody” account access rights. “Nobody” is the account that Apache server normally runs under. The attackers then looked for a vulnerable program that was running as user roo t. MySQL was targeted as the vulnerable program and was used to create a Trojan to capture root’s password.

© SANS Institute 2000 - 2002, Author retains full rights.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

Prevention

The rules for prevention are simple:

- Keep your operating system, web server and firewall patches up to date.
- Monitor your website site with a tool such as Freshwater Software's Sitescope so that you know your site is down before the newspapers do.
- Use strong authentication when registering your domain.
- Make sure that your DNS server is running the latest version of bind and that zone transfers are only allowed to authorized servers.
- Make sure that the ftproot directory cannot be seen from your web server root.

© SANS Institute 2000 - 2002, Author retains full rights.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

References

1. The Years of Living Dangerously: Internet Security in 2001
   IDC Breakfast Briefing Sydney, 22/3/2001
   Natasha David, *Senior Analyst, Enterprise and Internet Software, IDC Australia*
2. http://www.gilian.com/resource.html

Example 1

3. http://www.attrition.org/security/commentary/microsoft0123.html
4. http://www.theregister.co.uk/content/6/16296.html
5. http://www.sans.org/y2k/unico de.htm
6. http://www.microsoft.com/technet/security/bulletin/MS00 -086.asp
7. http://project.honeynet.org/scans/scan12/team.txt

Example 2

8. http://news.cnet.com/news/0 -1003-200-4508359.html

Example 3

9. http://www.securitywatch.com/newsforward/default.asp?AID=3137
10. http://www.securiteam.com/securitynews/Can_you_keep_your_domain_from_bei ng_hijacked_.html

Example 4

11. http://www.securiteam.com/securitynews/How_apache_org_was_defaced.htm
12. http://www.php.net
13. http://packetstorm.securify.com/advisories/suid/003.txt

© SANS Institute 2000 - 2002, Author retains full rights.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

Assignment 3 – Analyze This

Executive Summary

On an unprotected network there is a huge amount of traffic that can be categorized as suspicious, needing further investigation or just plain shouldn't be there. The data provided proves that this network is highl y susceptible to such traffic. In one month's data there were 194039 alerts, 820398 UDP scans, 492408 TCP scans and 61209 OOS (out of spec) packets. Most of this traffic is at the reconnaissance or information gathering stage.

Alerts found which can gain root access, seriously compromise a system or cause denial of service:

- Attempted Sun RPC high port access/ SUNRPC highport access!
- Back Orifice
- External RPC call
- SITE EXEC - Possible wu -ftpd exploit - GIAC000623
- STATDX UDP attack
- DNS udp DoS attack describ ed on unisog
- connect to 515 from outside/ connect to 515 from inside
- SNMP public access

The large number of scans originating from net 10.1 (MY.NET) would indicate that either:

- someone on the internal network is performing scans,
- that a large number of ho sts have been compromised or,
- that addresses are being spoofed.

A large number of the scans may be attributed to staff playing online games or using chat.

The OOS logs indicate that there is a large amount of crafted packets entering the network.

© SANS Institute 2000 - 2002, Author retains full rights.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

Alerts

194039 alerts found

Signature	# Alerts	# Sources	# Destinations
SITE EXEC - Possible wu-ftpd exploit - GIAC000623	1	1	1
STATDX UDP attack	1	1	1
Happy 99 Virus	1	1	1
site exec - Possible wu-ftpd exploit - GIAC000623	2	2	2
Probable NMAP fingerprint attempt	8	5	6
External RPC call	59	15	25
Back Orifice	77	10	71
TCP SMTP Source Port traffic	100	5	88
Broadcast Ping to subnet 70	154	24	1
connect to 515 from inside	159	10	98
SUNRPC highport access!	204	25	19
SMB Name Wildcard	515	93	171
Russia Dynamo - SANS Flash 28-jul-00	546	2	2
NMAP TCP ping!	558	47	156
SNMP public access	591	20	7
Queso fingerprint	710	52	72
Null scan!	826	527	173
Attempted Sun RPC high port access	2053	16	23
WinGate 1080 Attempt	2239	474	572
Watchlist 000222 NET-NCFC	2401	31	19
connect to 515 from outside	4238	10	2877
Tiny Fragments – Possible Hostile Activity	5340	27	13

© SANS Institute 2000 - 2002. Author retains full rights.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

DNS udp DoS attack described on unisog	16146	8	6
SYN-FIN scan!	51192	37	27067
Watchlist 000220 IL-ISDNNET-990517	105918	46	100

Watchlist 000220 IL-ISDNNET-990517/ Watchlist 000220 IL-ISDNNET-990517

This alert is identifying certain addresses from China and Israel. These alerts have been removed from the current snort rulebase and should not be of any concern.

http://zounds.net/practical.html#Anal yze_This

SYN-FIN scan!

This is a SYN/FIN probe probably looking for active TCP ports. A SYN/FIN is sent to elicit the expected response of a RST ACK from an inactive port. Windows NT/95/98 would respond with a FIN/ACK for an inactive port. The intention is to find open ports behind a firewall. The technique of using SF and well known source ports is to try and fool some packet filtering firewalls that don't keep state.

DNS udp DoS attack described on unisog

This refers to a flood of DNS traffic comin g to UDP port 53 looking up an mx record for aol.com. This was discussed on the "unisog" list.

This traffic to be blocked at the router.

6/1/2001
http://www.google.com/search?q=cache:www.theorygroup.com/Archive/Unisog/2001/msg00005.html+unisog+DNS+DoS+a ttack&hl=en

Tiny Fragments - Possible Hostile Activity

This is probably a host port scan that is trying to elude detection by the firewall or IDS by using the technique of tiny fragments. The tiny fragments are splitting the TCP header into two parts

Ref: SANS 3.3/3.4 p192

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

connect to 515 from outside/ connect to 515 from inside

This is looking for any connections to port 515 which is the BSD LPD or print spooler. This is to alert a possible DOS attack. There is a candidate for inclusion in the CVE list that states: "A continuous stream of LPD options, sent to the LPD port (default TCP port 515) on the host running WinCOM, will eventually consume all the memory on that host"

CAN-2000-0839
http://archives.neohapsis.com/archives/bugtraq/2000 -09/0212.html

WinGate 1080 Attempt

This alert identifies any attempted connection to TCP port 1080. This port usually runs socks on Wingate proxy server. The intention may be looking for a vehicle to relay traffic.

http://www.whitehats.com/info/IDS175

Attempted Sun RPC high port access/ SUNRPC highport access!

Any attempted connection to UDP port 32770+

Solaris rpcbind listens on a high numbered UDP port, which may not be filtered since the standard port number is 111.

CVE-1999-0189

Null scan!

Packets with no flag set. Attacker is trying to allude detection while scanning for open ports.

IDS004
http://ftp.cert.org.tw/tools/IDS/Snort/snort/scan -lib

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

Queso fingerprint

Attacker is using a tool called Queso to fingerprint the OS of a system.

http://www.whiteh ats.com/info/IDS29
CVE CAN-1999-0454

SNMP public access

Attacker is looking for hosts that run SNMP with public set as the community string. If successful, this will give the attacker some very useful information including operating system and routing ta bles. If SNMP is set to read/write then there is potential to make changes to a system.

Make sure that SNMP is set to read only not read/write and that the community string is not set to "public"

NMAP TCP ping!

Using TCP instead of icmp to determine if a host is reachable.

http://www.whitehats.com/info/IDS28
CVE CAN-1999-0523

Russia Dynamo - SANS Flash 28-jul-00

It's difficult to know the intent of this traffic. A whois at RIPE reveals the source address is Russian. The source port is 2478 (SecurSigh t Authentication Server (SLL), destination port is 6699 which is unknown.

SMB Name Wildcard

This is information gathering. The SMB Name Wildcard is used to request remote machine netbios name.

http://lists.sourceforge.net/archives/snort -users/2000-August/000636.html
http://www.google.com/search?q=cache:63.224.89.202/lyle/mirrors/www.sans.org/ne wlook/resources/IDFAQ/port_137.htm+SMB+Name+Wildcard&hl=en

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

Broadcast Ping to subnet 70

This is an attempt to map the network by sending a broadcast ping to MY.NET.70.255. Note that Windows hosts don't respond to an ICMP ping.

TCP SMTP Source Port traffic

This is similar to IDS6 and IDS7 except that the packet is on source TCP port 25 (SMTP). The intent is exploit a design weakness in some packet filters that al low a response to enter on a port even though the service is denied.

http://www.whitehats.com/info/IDS6
http://www.whitehats.com/info/IDS7

Back Orifice

This is most likely a probe to UDP port 31337 looking for the existence of the Back orifice Trojan wh ich allows the attacker to take complete control over Windows 9x/NT hosts. If the host has been compromised then the alert would be " `IDS189/trojan-active-back-orifice`"

CAN-1999-0660
http://www.whitehats.com/info/IDS188
http://www.whitehats.com/info/IDS1 89

External RPC call

This alert is detecting attempted connections to TCP/UDP port 111. The Remote Procedure Call essentially allows a program running on a Unix system to execute a command on a remote Unix computer. This is a very high security risk and this port should be blocked at the router and firewall.

http://www.landfield.com/rfcs/rfc1831.html

Probable NMAP fingerprint attempt

NMAP uses a series of nine tests to determine the OS of a host.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

SITE EXEC - Possible wu-ftpd exploit - GIAC000623

Attempting to execute the "site exec" command on the wu -ftp FTP server allows root access via "site exec" command.

CVE-1999-0080

Happy 99 Virus

The Happy 99 Virus signature was detected in a message to address MY.NET.6.47. The virus will execute a firework s display and then replicate itself via email.

http://www.symantec.com/avcenter/venc/data/happy99.worm.html

STATDX UDP attack

This attack is aimed at compromising a Redhat 6.0 linux system running the rpc.statd service. The protocol used can be either T CP or UDP.

http://www.whitehats.com/info/IDS442

CVE-2000-0666

© SANS Institute 2000 - 2002 Author retains full rights.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

Snort Scan Reports

These reports detected numerous scans that showed a high level of repetition among the Snort Scan reports. Some of the more interesting examples are:

UDP scans

Total number of UDP scans 820398.

Top ten source ports :

	Port Usage
Src Port 6112 appears 123134 times	Dstpcd
Src Port 28800 appears 121492 times	Unassigned
Src Port 9753 appears 64306 times	RASADV
Src Port 53 appears 53385 times	Domain Name Server
Src Port 9000 appears 44553 times	CSlistener
Src Port 32780 appears 40693 times	Unassigned
Src Port 666 appears 25612 times	mdqs, doom Id Software
Src Port 0 appears 24991 times	Reserved
Src Port 32781 appears 21505 times	Unassigned
Src Port 7001 appears 19024 times	afs3 - callbacks to cache managers, Freak 88

Top ten destination ports:

	Port Usage
Dst Port 6112 appears 116087 times	Dstpcd
Dst Port 28800 appears 110283 times	Unassigned
Dst Port 7778 appears 64997 times	Interwise
Dst Port 53 appears 64558 times	Domain Name Server
Dst Port 27015 appears 46377 times	Unassigned
Dst Port 9000 appears 31559 times	CSlistener
Dst Port 0 appears 24992 times	Reserved
Dst Port 9004 appears 12854 times	Unassigned
Dst Port 7000 appears 11385 times	afs3 - file server itself, SubSeven
Dst Port 17771 appears 10747 times	Unassigned

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

Top ten source IP addresses:

Src IP 10.1.98.200 appears 64786 times
Src IP 10.1.100.230 appears 62243 times
Src IP 10.1.213.186 appears 54667 times
Src IP 10.1.202.94 appears 42256 times
Src IP 10.1.218.158 appears 37761 times
Src IP 10.1.217.94 appears 33715 times
Src IP 10.1.214.166 appears 27825 times
Src IP 10.1.217.150 appears 20167 times
Src IP 10.1.218.130 appears 18420 times
Src IP 10.1.156.110 appears 18234 times

Top ten destinatio n IP addresses:

Dst IP 207.46.204.86 appears 14838 times
Dst IP 216.15.60.112 appears 10696 times
Dst IP 10.1.98.133 appears 9655 times
Dst IP 203.164.58.41 appears 6459 times
Dst IP 207.46.204.93 appears 5150 times
Dst IP 207.46.204.98 appears 4962 times
Dst IP 207.46.204.95 appears 3923 times
Dst IP 24.22.135.10 appears 3825 times
Dst IP 209.90.4.89 appears 3759 times
Dst IP 194.251.249.182 appears 3753 times

© SANS Institute 2000 - 2002, Author retains full rights.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

Observations:

The large number of scans originating from net 10.1 (MY.NET) would indicate tha t either:

- someone on the internal network is performing scans,
- that a large number of hosts have been compromised or,
- that addresses are being spoofed.

Weird Ports!:

I investigated some of the top ten ports to determine their intention:

- The game Diablo running on BattleNet servers use port 6112.
- Port 28000 udp is used in the popular online game Starsiege Tribes.
- Port 7798 AsterCity games Port 9753 weather update client, example port in iPlanet web server doc.
- Port 666 – Doom
- Port 7000, 7001 – AOL Chat P orts

Some products are very aggressive at trying to reestablish a connection what we may be seeing is connection attempts rather than scans.

Port 0 is a reversed port that should not normally be used. The purpose of these scans would be to check OS reacti ons to traffic on port 0. Checkpoint FW -1 is known to have a bug where traffic to any host on port 0 caused the firewall to reboot.

References:
http://archives.neohapsis.com/archives/incidents/2000 -03/0171.html
http://archives.neohapsis.com/archives/incide nts/2000-08/0256.html
http://mantisquad.republika.pl/serwery.htm
http://www.dotfunk.com/projects/weatherupdate/
http://www.securiteam.com/exploits/CheckPoint_Firewall1_is_vulnerable_to__Port_0__Denial_of_Service_attack.html
http://members.nbci.com/animech at/aolfaq.htm

© SANS Institute 2000 - 2002, Author retains full rights.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

TCP scans

Total number of TCP scans 492408

Type SYN appears 448550 times
Type SYNFIN appears 26037 times
Type NOACK appears 5332 times
Type INVALIDACK appears 3805 times
Type UNKNOWN appears 2552 times
Type FIN appears 2240 times
Type NULL appears 1716 times
Type VECNA appears 1229 times
Type FULLXMAS appears 362 times
Type XMAS appears 246 times
Type SPAU appears 194 times
Type NMAPID appears 145 times

Flags **S***** appears 448028 times
Flags **SF**** appears 25877 times
Flags ***F**** appears 2094 times
Flags ******** appears 1390 times
Flags 21S***** appears 394 times
Flags 2***R*A* appears 286 times
Flags ****R**U appears 239 times
Flags *1S*R*** appears 235 times
Flags *1*FRP*U appears 195 times
Flags 2**F*P** appears 195 times

The intent of setting bogus TCP flag values is to see if the target host discards/keeps the non-existent flags and thereby fingerprint the OS.

© SANS Institute 2000 - 2002, Author retains full rights.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

OOS Logs:

OOS stands for Out of Spec!
These log contain a large variety of packets which show signs of packet craf t.

Some examples:

```
OOSche12.txt:
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
12/15-02:28:38.763000 194.197.170.7:9055  -> 10.1.1.5:9055
TCP TTL:25 TOS:0x0 ID:39426
**SF**** Seq: 0x5EF0F347   Ack: 0x3E0FAA6E   Win: 0x404
00 00 00 00 0 0 00                    ......

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
12/15-02:28:38.903677 194.197.170.7:9055  -> 10.1.1.12:9055
TCP TTL:25 TOS:0x0 ID:39426
**SF**** Seq: 0x5EF0F347   Ack: 0x3E0FAA6E   Win: 0x404
00 00 00 00 00 00                     ......

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
12/15-02:28:39.042036 194.197.170.7:9055  -> 10.1.1.19:9055
TCP TTL:25 TOS:0x0 ID:39426
**SF**** Seq: 0x5EF0F347   Ack: 0x3E0FAA6E   Wi  n: 0x404
00 00 00 00 00 00                     ......

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
```

Anomalies are:

SF – SYN FIN flags set
Source and Destination ports the same
IP ID is static
TCP sequence number the same.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

```
OOScheck.txt:
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
11/28-09:58:39.852782 195.132.81.119:1679   -> 10.1.212.38:4336
TCP TTL:113 TOS:0x0 ID:18320  DF
*1SFR*** Seq: 0x19A   Ack: 0xBA020A57   Win: 0x5010
TCP Options => EOL EOL EO L EOL EOL EOL SackOK NOP SackOK NOP EOL Opt 49
Opt 49 Opt 49 Opt 49 Opt 49 Opt 49 Opt 49 Opt 49 Opt 49 Opt 49 Opt 49 Opt 49 Opt 49 Opt
49 Opt 49 Opt 49 Opt 49 Opt 49 Opt 49 Opt 49 Opt 49 Opt 49 Opt 49 Opt 49 Opt 49
Opt 49 Opt 49 Opt 49 Opt 49

=+=+=+=+=+=+ =+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
```

An example of multiple TCP options being set. The aim of this packet is an insertion attack.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

Analysis Process

For all files I first substituted MY.NET with 10.1 using sed. eg:

```
#!/bin/sh
for i in `ls Snort*.txt`
do
      cat $i |sed s/MY.NET/10.1/g > $i.cnv
      mv $i.cnv $i
done
```

Alerts

SnortSnarf was used to analyze the alert files. I used the following script:

```
#!/bin/sh
alert_dir=/home/httpd/html/snort/alert
html_dir=/home/httpd/html/s narf/alert
snort_dir=http://XXX.XXX.XXX.XXX/snort/alert
bin=/usr/local/bin
rulesfile=/usr/local/bin/snort/snort.conf
rulesdir=/usr/local/bin/snort

$bin/snortsnarf.pl -d $html_dir -ldir "$snort_dir" -homenet 10.1/16 \
-rulesfile $rulesfile -rulesdir $rules dir $alert_dir/SnortA*.txt
```

Scans

I started to analysis the scans with SnortSnarf but gave up after several attempts. SnortSnarf either filled up my disk or exhausted all swap space. I even tried adding another 256MB of swap to my initial 128MB of swap and 128MB of RAM but still no success.

UDP Scans

I first used grep to extract all the UDP scans from the files and create one large UDP scan file eg:

```
grep UDP SnortS*.txt > UDP.out
```


© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

Then I used awk to create another file with a field separator of : to be u sed for further analysis. I also sorted the file by field eight which is the src port. eg:

```
# prep1.sh
# =======
# Set field separator to be either : or space
BEGIN {FS=":|[ \t]+"}
# Put colons between all fields
# and then sort on the 8th field (src port)
{printf( "%s:%s:%s:%d:%d:%d:%s:%d:%s:%s:%d:%s  \n",
      $1,$2,$3,$4,$5,$6,$7,$8,$9,$10,$11,$12) | "sort  -t: +7"
}

# awk –f prep1.awk UDP.out > UDP_sort.out
```

I then used a series of awk programs to count the source and destination ports by frequency and the source and destination IP addresses by frequency. eg:

```
# c1.awk
# Count Src Ports
# Set Field Separator = :
BEGIN {FS=":"}

# Do a count of each port and assign to array count
{
count[$8]++
}

# Print it out
END { for (port in count)
            printf ("Src Port %d appears %d times \n", port, count[port])
      }

# awk –f c1.awk UDP_sort.out > src_port.out
```

© SANS Institute 2000 - 2002, Author retains full rights.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

```
# c2.awk
# Count Dst Ports
# Set Field Separator = :
BEGIN {FS=":"}

# Do a count of each port and assign to array count
{
count[$11]++
}

# Print it out
END { for (port in count)
        printf ("Dst Port %d appears %d times \n", port, count[port])
    }

# awk –f c2.awk UDP_sort.out > dst_port.out


# c3.awk
# Count Src IP addresses
# Set Field Separator = :
BEGIN {FS=":"}

# Do a count of each port and assign to array count
{
count[$7]++
}

# Print it out
END { for (ip in count)
        printf ("Src IP %s appears %d times \n", ip, count[ip])
    }


# awk –f c3.awk UDP_sort.out > src_ip.out
```

© SANS Institute 2000 - 2002, Author retains full rights.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

```
# c4.awk
# Count Dst IP addresses
# Set Field Separator = :
BEGIN {FS=":"}

# Do a count of each port and assign to array count
{
count[$10]++
}

# Print it out
END { for (ip in count)
        printf ("Dst IP %s appears %d times \n", ip, count[ip])
    }

# awk –f c4.awk UDP_sort.out > dst_ip.out
```

Finally, I sorted each output file to find the top ten:

```
# sort.sh
#!/bin/sh
# sort on the number of times it appears
sort -t" " +4rn $1

# ./sort.sh dst_ip.out > sort_dst_ip.out | head sort_dst_ip.out
```

TCP Scans

I used a series of greps to extract all the TCP scans into one file:

```
grep "\->" SnortS*.txt > TCP1.out &
grep -v "UDP" TCP1.out > TCP.out
```

Then I used awk to create another file with a field separator of : to be used for further analysis. eg:

```
# prep_tcp.awk
# Set field separator to be either : or space
BEGIN {FS=":|[ \t]+"}
# Put colons between all fields
{printf( "%s:%s:%s:%d:%d:%d:%s:%d:%s:%s:%d:%s:%s  \n",
    $1,$2,$3,$4,$5,$6,$7,$8,$9,$10,$11,$12,$13)
}

# awk -f prep_tcp.awk TCP.out > TCP_1.out
```


© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

Next determine the type of TCP packets being sent. eg:

```
# c_type.awk
# Count Type of Packet
# Set Field Separator = :
BEGIN {FS=":"}

# Do a count of each type and assign to array count
{
count[$12]++
}

# Print it out
END { for (type in count)
        printf ("Type %s appears %d times \n", type, count[type])
    }

awk -f c_type.awk TCP_1.out > type.out
```

Finally, sort the output:

```
# sort.sh
#!/bin/sh
# sort on the number of times it appears
sort -t" " +3rn $1

# ./sort.sh type.out > sort_type.out
```

Next determine the flags being sent:

```
# Count Flags
# Set Field Separator = :
BEGIN {FS=":"}

# Do a count of each flags and assign to array count
{
count[$13]++
}

# Print it out
END { for (flags in count)
        printf ("Type %s appears %d times \n", flags, count[flags])
    }

# awk -f c_flags.awk TCP_1.out > flags.out
```

© SANS Institute 2000 - 2002, Author retains full rights.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

Finally, sort the output:

```
# sort.sh
#!/bin/sh
# sort on the number of times it appears
sort -t" " +3rn $1

# ./sort.sh flags.out > sort_flags.out
```

© SANS Institute 2000 - 2002, Author retains full rights.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.