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
"Intrusion Detection In-Depth (Security 503)"
at http://www.giac.org/registration/gcia

Practical for SANS DC2000 Intrusion Detection

Merik Karman

Assignment One – Network Detects

Number One – Vulnerable CGI programs and application extensions. (old attacks never die)

Supporting RealSecure data

Event: HTTP_Shells

Priority	Date	From	From Port	To	To Port
Information					
High	3/08/00 5:22:42	24.192.0.14	35712	X.X.121.3	80
URL	/c%67%69-bi%6E/%70er%6C.exe (***This decodes to /cgi-bin/perl.exe)***				
High	3/08/00 5:22:45	24.192.0.14	35778	X.X.121.3	80
URL	/cgi-b%69%6E/ma%6E.sh (***This decodes to /cgi-bin/man.sh)***				
High	3/08/00 6:06:38	24.192.0.14	50156	Y.Y.86.130	80
URL	/cg%69-bin/pe%72l.e%78e (***Still /cgi-bin/perl.exe just different encoding)***				
High	3/08/00 6:06:39	24.192.0.14	50206	Y.Y.86.130	80
URL	/c%67%69-bin/man.sh (***Ditto /cgi-bin/man.sh)***				
High	3/08/00 6:08:30	24.192.0.14	54265	Z.Z.112.48	80
URL	/cg%69-bin/pe%72l.e%78e (***Back to the same encoding)***				
High	3/08/00 6:08:31	24.192.0.14	54317	Z.Z.112.48	80
URL	/c%67%69-bin/man.sh (***Ditto)***				

The data above is extracted from a database of network even ts recorded by the RealSecure system. This detect shows the a **priority** of "High", followed by a **date and time stamp** , the **source address** of "24192.0.14", the **source port** of "35712", the **destination address** of a government web server and **destination port** of "80". The following line tells gives us information about the specific URL.

Event: HTTP_DotDot

Priority	Date	From	From Port	To	To Port
Information					
High	3/08/00 5:22:58	24.192.0.14	36156	X.X.121.3	80
URL	/../../../../				
High	3/08/00 5:22:58	24.192.0.14	36159	X.X.121.3	80
URL	/../../../../config.sys				
High	3/08/00 5:22:58	24.192.0.14	36166	X.X.121.3	80
URL	/../../../../etc/hosts				
High	3/08/00 5:22:59	24.192.0.14	36182	X.X.121.3	80
URL	/../../../../../../../boot.ini				
High	3/08/00 5:22:59	24.192.0.14	36184	X.X.121.3	80
URL	/../../../../../winnt/repair/sam._				
(snip similar traffic to multiple web servers)					
High	3/08/00 6:08:42	24.192.0.14	54734	Z.Z.112.48	80
URL	/../../../../				
High	3/08/00 6:08:42	24.192.0.14	54740	Z.Z.112.48	80
URL	/../../../../config.sys				
High	3/08/00 6:08:42	24.192.0.14	54742	Z.Z.112.48	80
URL	/../../../../etc/hosts				
High	3/08/00 6:08:42	24.192.0.14	54755	Z.Z.112.48	80

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

URL /../../../../../../../boot.ini
High 3/08/00 6:08:42 24.192.0.14 54758 Z.Z.112.48 80
URL /../../../../../winnt/repair/sam._

(snip similar traffic from this source trying multiple cgi and active content vulnerabilities)

Supporting snort data

```
[**] IDS250/web -coldfusion -openfile [**]
08/03 -05:53:37.776789 0:E0:1E:BC:7A:49 -> 8:0:20:78:F4:49 typ e:0x800 len:0xC5
24.192.0.14:23022 -> X.X.147.130:80 TCP TTL:54 TOS:0x0 ID:56443 DF
*****PA* Seq: 0x2E9B0295 Ack: 0x24DC70CD Win: 0x2238
47 45 54 20 2F 63 66 64 6F 63 73 2F 65 78 70 65 GET /cfdocs/expe
6C 76 61 6C 2F 6F 70 65 6E 66 69 6C 65 2E 63 66 lval/openfile.cf
<snip>
6F 76 2E 61 75 0D 0A 56 69 61 3A 20 31 2E 31 20 ov.au..Via: 1.1
6E 65 74 61 70 70 20 28 4E 65 74 43 61 63 68 65 netapp (NetCache
20 34 2E 30 52 34 44 31 29 0D 0A 43 6F 6E 6E 65 4.0R4D1)..Conne
63 74 69 6F 6E 3A 20 4B 65 65 70 2 D 41 6C 69 76 ction: Keep -Aliv
65 0D 0A 0D 0A e....

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

[**] IDS219/web -cgi-perl-exe [**]
08/03 -13:34:28.532777 0:E0:1E:BC:7A:49 -> 8:0:20:78:F4:49 type:0x8 00 len:0xBE
24.192.0.14:31486 -> 192.55.Y.Y:80 TCP TTL:54 TOS:0x0 ID:46752 DF
*****PA* Seq: 0xE9816F07 Ack: 0x8E50DB73 Win: 0x2238
TCP Options => NOP NOP TS: 548528 354364771
47 45 54 20 2F 63 67 69 2D 62 69 6E 2F 70 65 72 GET /cgi -bin/per
6C 2E 65 78 65 20 48 54 54 50 2F 31 2E 30 0D 0A l.exe HTTP/1.0..
<snip>
76 2E 61 75 0D 0A 56 69 61 3A 20 31 2E 31 20 6E v.au..Via: 1.1 n
65 74 61 70 70 20 28 4E 65 74 43 61 63 68 65 20 etapp (NetCache
34 2E 30 52 34 44 31 29 0D 0A 43 6F 6E 6E 65 63 4.0R4D1)..C onnec
74 69 6F 6E 3A 20 4B 65 65 70 2D 41 6C 69 76 65 tion: Keep -Alive
0D 0A 0D 0A ....

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

Supporting tcpdump data

```
02:13:53.624227 CPE-24-192-0-14.vic.bigpond.net.au.8843 > www.X.gov.au.www: S 1632619934:1632619934(0) win 20480 (DF)
02:13:53.627839 www.X.gov.au.www > CPE-24-192-0-14.vic.bigpond.net.au.8843: S 268770007:268770007(0) ack 1632619935 win 8760 (DF)
02:13:53.641288 CPE-24-192-0-14.vic.bigpond.net.au.8843 > www.X.gov.au.www: . ack 268770008 win 8760 (DF)
02:13:53.642948 CPE-24-192-0-14.vic.bigpond.net.au.8843 > www.X.gov.au.www: P 1632619935:1632620514(579) ack 268770008 win 8760 (DF)
02:13:53.646361 www.X.gov.au.www > CPE-24-192-0-14.vic.bigpond.net.au.8843: P 268770008:268770801(793) ack 1632620514 win 8181 (DF)
02:13:53.646431 www.X.gov.au.www > CPE-24-192-0-14.vic.bigpond.net.au.8843: F 268770801:268770801(0) ack 1632620514 win 8181 (DF)
02:13:53.662543 CPE-24-192-0-14.vic.bigpond.net.au.8843 > www.X.gov.au.www: . ack 268770802 win 8760 (DF)
02:14:08.551422 CPE-24-192-0-14.vic.bigpond.net.au.8843 > www.X.gov.au.www: F 1632620514:1632620514(0) ack 268770802 win 8760 (DF)
```


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

02:14:08.552610 www.X.gov.au.www > CPE-24-192-0-14.vic.bigpond.net.au.8843: . ack 1632620515 win 8181 (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 production network.

2 Detect was generated by

2.1 RealSecure

The RealSecure network sensor was running on the external network segment of my production network.

2.2 tcpdump

A shadow sensor logging tcpdump data on the same segment gathered the supporting data.

2.3 Snort

As a backup to our primary IDS we run the snort IDS system as well. This system uses the arachNIDS database obtained from www.whitehats.com .

3 Probability that the source address was spoofed

Low.

The attacker is trying to gather information about vulnerable programs on a web server therefore he wants the response from each of the servers. He could spoof the source and ensure that he had access to a packet sniffer somewhere on the packet return path. This would be however unlikely.

4 Description of the attack

Our attacker is obviously fishing for CGI programs. He is trying some IDS evation techniques by encoded ASCII values instead of the real characters. N otice the evasion techniques vary between probes to different web servers. Either a smart automated evasion technique or someone typing by hand, although the lack of backspace characters would lead me to believe the former. This in fact looks like someone using a scanner like whisker although I have not been able to determine whether whisker can randomise its encoding. Anyway the point being that this looks like an a very “behind the times” script kiddy who has just discovered packetstorm and is testing for very old vulnerabilities. We should however not write off the possibility that someone is still running this old code.

A good reference for URL encoding of ASCII is at *http://i-technica .com/whitestuff/urlencodechart.html*

5 Attack Mechanism

Without the benefit of logs prior to this event I am unable to demonstrate the scanning that almost certainly preceded this attack. This scanning could have either come from the same address or from a preceding information -gathering scan from a different source. Once the attacker has found his target web servers he has run an automated cgi scanning tool like whisker against the list.

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

6 Correlation

The logs from RealSecure are supported by both tcpdump and snort output.

BugTRAQ May 1998 "The May 1998 issue of SysAdmin Magazine contains an article, "Web-Enabled Man Pages", which includes source code for very nice cgi script named man.sh to feed man pages to a web browser. The hypertext links to other man pages are an especially attractive feature. Unfortunately, this script is vulnerable to attack. Essentially, anyone who can execute the cgi thru their web browser can run any system commands with the user id of the web server and obtain the output from them in a web page. "

http://www.insecure.org/sploits/pcwebserver.perl-cgi.html describes the issues with leaving perl.exe in the cgi-bin directory. This is probably what the attacker is looking for.

Information on the whisker CGI scanner can be found at Rain Forrest Puppy's site at http://www.wiretrip.net/rfp

7 Evidence of active targeting

This is a deliberate scan of specific web servers. I unfortunately do not have any logs predating the attack to analyse the techniques used to find the web servers. I can only assume that those logs would reveal a port scan across my subnets looking for accessible web servers. With only one external IDS point within my network it is impossible to determine if this scanning was carried out on a wider audience.

8 Severity

The standard formula for determining severity is as follows:

(Criticality + Lethality) - (System countermeasures + Network countermeasures)

Each element is scored from 1 to 5, 1 being low, 5 being high.

Criticality = 3. While these servers are public web servers, they are key Government web sites with a high degree of political embarrassment potential.

Lethality = 3 . The scan is in its elf not damaging.

System countermeasures = 2 . The systems are not all up to date. With no additional host based countermeasures.

Network countermeasures = 3. There is a strong filtering firewall between these servers and the Internet. While this will not prevent HTTP attacks it will prevent other attacks and gives us a high degree of logging and visibility.

Severity = (3 + 3) - (2 + 3) = 1.

9 Defensive recommendations

Each of the web servers targeted should be immediately audited to ensure current patch levels and hot-fixes. These should be brought up to date immediately and an internal vulnerability scan performed with tools like Nesus or Whisker.

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

10 Multiple choice test question

Which of the following is a well known IDS evasion tactic?

a) Using random hig h TCP source ports
b) Using ASCII values within a URL in place of the real character
c) Clipping the transmit wire on the ethernet card
d) None of the above.

Answer: (**b**)

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

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

Number Two – Buffer Overflow (Or don't jump to conclusions)

Supporting RealSecure data

Event: Generic_Intel_Overflow

Priority	Date	From	From Port	To	To Port
Information					
High	8/08/00 7:18:16	207.126.127.68	49224	X.X.86.20	25
ARCH	Intel	OS	Linux		
High	8/08/00 7:18:16	207.126.127.68	49224	X.X.86.20	25
ARCH	Intel	OS	Linux		
High	11/08/00 6 :12:09	207.126.127.68	44710	X.X.86.20	25
ARCH	Intel	OS	Linux		
High	11/08/00 6:12:10	207.126.127.68	44710	X.X.86.20	25
ARCH	Intel	OS	Linux		

The data above is extracted from a database of network events recorded by the RealSecure system. This detect shows the a **priority** of "High", followed by a **date and time stamp** , the **source address** of "207.126.127.68", the **source port** of "49224" or "44710", the **destination address** of my external DNS/SMTP server and **destination port** of "25". The following line tells us that the machine **architecture** is "Intel" and the **OS** is "Linux".

Supporting snort data

```
[**] IDS181/shellcode -x86-nops [**]
08/11-06:12:56.615723 0:E0:1E:BC:7A:49   -> 8:0:20:78:F4:49 type:0x800 len:0x5EA
207.126.127.68:44710  -> X.X.86.20:25 TCP TTL:240 TOS:  0x0 ID:10088  DF
*****PA* Seq: 0x76F605C3   Ack: 0x6058C241   Win: 0x2238
0A 53 75 62 6A 65 63 74 3A 20 20 20 20 20 20 52  .Subject:      R
65 3A 20 4E 65 77 20 6F 72 20 6F 6C 64 20 46 54  e: New or old FT
50 20 65 78 70 6C 6F 69 74 3F 0D 0A 54 6F 3A 20  P     exploit?..To:
49 4E 43 49 44 45 4E 54 53 40 53 45 43 55 52 49  INCIDENTS@SECURI
54 59 46 4F 43 55 53 2E 43 4F 4D 0D 0A 49 6E 2D  TYFOCUS.COM..In    -
52 65 70 6C 79 2D 54 6F 3A 20 20 3C 6D 33 38 7A  Reply    -To:  <m38z
75 39 39 65 6B 33 2E 66 73 66 40 63 65 72 6   5 73  u99ek3.fsf@ceres
2E 75 6E 69 74 2E 6C 69 75 2E 73 65 3E 0D 0A 0D  .unit.liu.se>...
0A 48 69 20 21 0D 0A 49 20 62 65 6C 69 65 76 65  .Hi !..I believe
20 69 74 27 73 20 74 68 65 20 22 6E 65 77 22 20   it's the "new"
77 75 66 74 70 64 32 2E 36 2E 30 28   31 29 20 65  wuftpd2.6.0(1) e
78 70 6C 6F 69 74 2C 20 49 20 68 61 64 20 74 68  xploit, I had th
<snip>
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 31 C0 31 DB 31   C9 B0 0D 0A 46 CD 80 31 C0  ..1.1.1....F..1.
31 DB 43 89 D9 41  B0 3F CD 80 EB 6B 5E 31  C0 31  1.C..A.?...k^1.1
C9 8D 5E 01 88 46 04 66  B9 FF FF 01 B0 27 CD 80  ..^..F.f....'..
31 C0 8D 5E 01 B0 3D CD 80 31 C0 31 DB 8D 5E 08  1..^..=..1.1..^.
89 0D 0A 43 0 2 31 C9 FE C9 31 C0 8D 5E 08 B0 0C  ...C.1...1..^...
CD 80 FE C9  75 F3 31 C0 88 46 09 8D 5E 08 B0 3D  ....u.1..F..^..=
CD 80 FE 0E B0 30 FE C8 88 46 04 31 C0 88 46 07  .....0...F.1..F.
89 76 08 89 46 0C 89 F3 8D 4E 08 8D 56 0C B0 0D  .v..F....N..V...
0A 0B  CD 80 31 C0 31  DB B0 01 CD 80  E8 90 FF FF  ....1.1.........
FF FF FF FF 30 62 69 6E 30 73 68 31 2E 2E 31 31  ....0bin0sh1..11
3C 2F 46 4F 4E 54 3E 0D 0A 3C 46 4F 4E 54 20 43  </FONT>..<FONT C
4F 4C 4F 52 3D 22 23 32 32 32 32 35 35 22 3E 3E  OLOR="#222255"    >>
20 50 41 53 53 20 0D 0A 90 90 90 90 90 90 90 90   PASS ..........
<snip>
```


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

11 Source of trace

My production network

12 Detect was generated by

12.1 RealSecure

The RealSecure network sensor was running on the external network segment of my production network.

12.2 Snort

As a backup to our primary IDS we run the snort IDS system as well. This system uses the arachNIDS database obtained from www.whitehats.com .

13 Probability that the source address was spoofed

Very Low

As we discover very quickly with this detect, it is not in fact an attack.

14 Description of the attack

When the alert for this one first came up on the console it caused quite a stir. An "Intel Overflow" it said, and against or primary DNS and SMTP external server. Unfortunately the RealSecure logs did not elaborate on the data within the offending packet. On closer inspection of the snort logs I discovered that it was indeed an overflow attack but not one against us. It was being discussed on a mailing list and some kind sole had posted the entire dump of the attack to the incidents mailing list at securityfocus.

15 Attack Mechanism

There was in fact no attack. The key lesson here is not to jump to conclusions and ensure that your IDS is set-up to log the suspicious packet. The ISS RealSecure sensor is quite capable of logging the offending packet but was not set-up to do so.

16 Correlation

The snort data was the key to solving this case.

17 Evidence of active targeting

There is no active targeting on this network, just on someone else's across the world ☺

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

18 Severity

The standard formula for determining severity is as follows:

(Criticality + Lethality) - (System countermeasures + Network countermeasures)

Each element is scored from 1 to 5, 1 being low, 5 being high.

Criticality = 4. This is our key external name server and mail relay..

Lethality = 1 . Its not an attack.

System countermeasures = 4 . The system is up to date and hardened.

Network countermeasures = 3. There is a strong filtering firewall between this server and the Internet.

Severity = (4 + 1) - (4 + 3) = -2.

19 Defensive recommendations

Any outside, Internet visible server should be kept up to date and hardened.

20 Multiple choice test question

Given the following trace, what is the most likely event?

```
08/11-06:12:56.615723 0:E0:1E:BC:7A:49   -> 8:0:20:78:F4:49 type:0x800 len:0x5EA
207.126.127.68:44710  -> X.X.86.20:25 TCP TTL:240 TOS:0x0 ID:10088  DF
*****PA* Seq: 0x76F605C3   Ack: 0x6058C241   Win: 0x2238
```

a) DNS lookup
b) Incoming mail to X.X.86.20
c) Port scan looki ng for SMTP servers
d) Netbios name query

Answer: (**b**)

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

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

Number Three – Port Scanning (A month in the life)

Supporting RealSecure data

Event: IPHalfScan

Priority	Date	From	From Port	To	To Port	Information
High	10/08/00 1:59:20	24.2.238.67	109	X.X.121.14	109	
High	10/08/00 1:59:20	24.2.238.67	109	X.X.121.15	109	
High	10/08/00 1:59:20	24.2.238.67	109	X.X.121.16	109	
High	10/08/00 1:59:20	24.2.238.67	109	X.X.121.17	109	
<snip> This goes on forever						
High	9/08/00 18:52:29	209.21.176.35	21	192.55.112.81	21	
High	9/08/00 18:52:29	209.21.176.35	21	192.55.112.82	21	
High	9/08/00 18:52:29	209.21.176.35	21	192.55.112.83	21	
High	9/08/00 18:52:29	209.21.176.35	21	192.55.112.84	21	
<snip> This goes on for about 25 subnets.						

The data above is extracted from a database of network events recorded by the RealSecure system. This detect shows the a **priority** of "High", followed by a **date and time stamp** , the **source address** of "24.2.238.67", the **source port** of "109", the **destination address** of "X.X.121.n" and **destination port** of "109".

Supporting tcpdump data

```
02:00:03.332688 cx149856-a.phnx1.az.home.com.109 > X.X.121.14.109: SF 1814829785:1814829785(0) win 1028
02:00:03.352137 cx149856-a.phnx1.az.home.com.109 > X.X.121.15.109: SF 1814829785:1814829785(0) win 1028
02:00:03.371470 cx149856-a.phnx1.az.home.com.109 > X.X.121.16.109: SF 1814829785:1814829785(0) win 1028
02:00:03.392437 cx149856-a.phnx1.az.home.com.109 > X.X.121.17.109: SF 1814829785:1814829785(0) win 1028
<snip>
18:53:09.255885 wks35.adero.com.21 > X.X.gov.au.21: SF 115687160:115687160(0) win 1028
18:53:09.280296 wks35.adero.com.21 > X.X.gov.au.21: SF 115687160:115687160(0) win 1028
18:53:09.291431 wks35.adero.com.21 > X.X.gov.au.21: SF 115687160:115687160(0) win 1028
18:53:09.322275 wks35.adero.com.21 > X.X.gov.au.21: SF 115687160:115687160(0) win 1028
<snip>
```

1 Source of trace

My production network

2 Detect was generated by

2.1 RealSecure

The RealSecure network sensor was running on the external network segment of my production network.

2.2 tcpdump

A shadow sensor logging tcpd ump data on the same segment gathered the supporting data.

3 Probability that the source address was spoofed

Very Low

The scanner needs to gather the results of this probe so it is important to receive the reply packets.

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

4 Description of the attack

The scanning we see in the above traces are typical of daily activity through the perimeter of my network. This happens on a daily basis and in abundance. In this case the scanner is searching for POP2 and FTP servers. These port scans typically target one servi ce and "walk" many subnets to find potential attack targets.

5 Attack Mechanism

The scanner is using a Syn Fin scan. Further evidence of "unnatural " packets are the sequence numbers all being the same and the source port the same as the service being scan ned for.

6 Correlation

The tcpdump data gives more of an insight into the scanning techniques.

7 Evidence of active targeting

There is no active targeting on this network, in fact a scatter gun approach is being used.

8 Severity

The standard formula for det ermining severity is as follows:

(Criticality + Lethality) - (System countermeasures + Network countermeasures)

Each element is scored from 1 to 5, 1 being low, 5 being high.

Criticality = 2. Information gathering only.

Lethality = 3 . Its not an attack .

System countermeasures = 4 . The system is up to date and hardened.

Network countermeasures = 3. There is a strong filtering firewall between most servers and the Internet.

Severity = (2 + 3) - (4 + 3) = -2.

9 Defensive recommendations

Any outside, I nternet visible server should be kept up to date and hardened.

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

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

10 Multiple choice test question

Given the following trace, what is the most likely event?

```
02:00:03.332688 cx149856-a.phnx1.az.home.com.109 > X.X.121.14.109: SF 1814829785:1814829785(0) win 1028
02:00:03.352137 cx149856-a.phnx1.az.home.com.109 > X.X.121.15.109: SF 1814829785:1814829785(0) win 1028
02:00:03.371470 cx149856-a.phnx1.az.home.com.109 > X.X.121.16.109: SF 1814829785:1814829785(0) win 1028
```

a) Attack
b) Active response from an IDS
c) Reconnaissance
d) Electronic mail traffic

Answer: (**c**)

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

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

Number Four – Echo Denial of Service

Supporting RealSecure data

Event: Echo_Denial_of_Service

Priority	Date	From	From Port	To	To Port	Information
High	19/07/00 5:28:50	195.162.36.13	106	X.X.112.255	7	
High	19/07/00 6:28:48	195.162.36.13	386	X.X.112.255	7	
High	19/07/00 8:01:41	195.162.36.13	508	X.X.112.255	7	
High	19/07/00 8:22:01	195.162.36.13	212	X.X.112.255	7	
High	19/07/00 8:40:40	195.162.36.13	143	X.X.112.255	7	
High	19/07/00 8:56:30	195.162.36.13	108	X.X.112.255	7	
High	19/07/00 9:31:30	195.162.36.13	80	X.X.112.255	7	
High	19/07/00 9:34:04	195.162.36.13	153	X.X.112.255	7	
High	19/07/00 9:43:35	195.162.36.13	46	X.X.112.255	7	
High	19/07/00 9:47:12	195.162.36.13	487	X.X.112.255	7	
High	19/07/00 10:06:38	195.162.36.13	117	X.X.112.255	7	
High	19/07/00 10:52:50	195.162.36.13	359	X.X.112.255	7	
High	19/07/00 11:42:46	195.162.36.13	449	X.X.112.255	7	
High	19/07/00 12:29:36	195.162.36.13	394	X.X.112.255	7	
High	19/07/00 13:33:09	195.162.36.13	262	X.X.112.255	7	
High	19/07/00 14:57:10	195.162.36.13	90	X.X.112.255	7	
High	19/07/00 18:17:33	195.162.36.13	348	X.X.112.255	7	

The data above is extracted from a database of network events recorded by the RealSecure system. This detect shows the a **priority** of "High", followed by a **date and time stamp** , the **source address** of "195.162.36.13", the **source port** of "various", the **destination address** of "X.X.112.255" and **destination port** of "7".

Supporting firewall logs

```
32415;19Jul2000; 6:36:46;203.32 .86.14;log;accept;;le0;inbound;icmp;195.162.36.13
;X.X.112.255;50;8;0;;;;;;;;;;;;;;
```

The data above is extracted from a firewall one log. The log shows a entry number followed by date, time, origin of the log, type of log entry, the action taken, the alert action taken, interface name, interface direction, protocol, source, destination, rule number, icmp type, icmp code.

```
32456;19Jul2000; 6:37:02;203.32.86.14;log;accept;;le0;inbound;udp;195.162.36.13;
X.X.112.255;109;;;echo -udp;9463;1052;;;;;;;;;;;
```

The data above is extracted from a firewall one log. The log shows a entry number followed by date, time, origin of the log, type of log entry, the action taken, the alert action taken, interface name, interface direction, protocol, source, destination, src port, dest port.

```
32490;19Jul2000; 6:37:17;203.32.86.14;log;accept;;le0;inbound;icmp;195.162.36.13
;X.X.112.255;50;8;0;;;;;;;;;;;;;;
32624;19Jul2000; 6:38:19;203.32.86.14;log;accept;;le0;inbound;icmp;195.162.36.13
;X.X.112.255;50;8;0;;;;;;;;;;;;;;
32657;19Jul2000; 6:38:58;203.32.86.14;log;accept;;le0;inbound;icmp;195.162.36.13
;X.X.112.255;50;8;0;;;;;;;;;;;;;;
32658;19Jul2000; 6:38:58;203.32.86.14;log;accept;;le0;inbound;udp;195.162.36.13;
X.X.112.255;109;;;echo -udp;51220;1052;;;;;;;;;;;
32712;19Jul2000; 6: 39:29;203.32.86.14;log;accept;;le0;inbound;icmp;195.162.36.13
;X.X.112.255;50;8;0;;;;;;;;;;;;;;
32730;19Jul2000; 6:39:52;203.32.86.14;log;accept;;le0;inbound;udp;195.162.36.13;
X.X.112.255;109;;;echo -udp;51980;1052;;;;;;;;;;;
<snip>
```


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

1 Source of trace

My production network

2 Detect was generated by

2.1 RealSecure

The RealSecure network sensor was running on the external network segment of my production network.

2.2 Firewall One

The primary external firewall is Checkpoint FirewallOne.

3 Probability that the source address was spoofed

Very High

It appears that this is a denial of service attack on X.X.112.0/24 network. As the resultant traffic is likely to be high it would be logical for the attacker to spoof the source so as to avoid any return traffic.

4 Description of the attack

This is a denial of service attack utilising both ICMP and UDP traffic directed to the broadcast address of the subnet.

5 Attack Mechanism

When I first noticed this attack the IDS had only picked up the UDP -Echo port traffic. Upon closer inspection the quantity of traffic through the firewall was very large and it included both UDP traffic directed to the echo port and ICMP echo requests, both directed at the broadcast address of the subnet in question.

6 Correlation

The Firewall logs gave a better picture of what was happening.

7 Evidence of active targeting

This subnet is one of the most open behind our firewalls. No tcpdump data is available to determine whether a probe preceded the attack. I believe that this “open” subnet was actively targeted.

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

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

8 Severity

The standard formula for determining severity is as follows:

(Criticality + Lethality) - (System countermeasures + Network countermeasures)

Each element is scored from 1 to 5, 1 being low, 5 being high.

Criticality = 4. This is a ke y Government subnet.

Lethality = 3 . Denial of service attacks likely to have performance impact.

System countermeasures = 2 . Some unpatched servers and unhardened systems.

Network countermeasures = 1. The filter rules are quite open to this sub -net.

Severity = (4 + 3) - (2 + 1) = 4.

9 Defensive recommendations

This subnet should be closed up to other than required traffic. All ICMP and small servers traffic should be blocked.

10 Multiple choice test question

Which of the following are typical signature s of DOS attacks

a) Traffic to port 7
b) ICMP type 8 to A.B.C.255
c) Chargen traffic
d) All the above

Answer: (**d**)

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

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

Number Five – Probe for an active Trojan

Supporting Pix data

Aug 10 12:40:45 [my.firewall.ip] %PIX -7-106011:
Deny self route tcp src outside:
209.179.4.249/4116 dst outside:ext.net.230.101/27374
Aug 10 12:40:45 [my.firewall.ip] %PIX -7-106011:
Deny self route tcp src outside:
209.179.4.249/4117 dst outside:ext.net.230.102/27374
Aug 10 12:40:45 [my.firewall.ip] %PIX -7-106011:
Deny self ro ute tcp src outside:
209.179.4.249/4118 dst outside:ext.net.230.103/27374
Aug 10 12:40:45 [my.firewall.ip] %PIX -7-106011:
Deny self route tcp src outside:
209.179.4.249/4119 dst outside:ext.net.230.104/27374
Aug 10 12:40:45 [my.firewall.ip] %PIX -7-106011:
Deny self route tcp src outside:
209.179.4.249/4120 dst outside:ext.net.230.105/27374

1 Source of trace

http://www.sans.org/y2k/081300.htm

2 Detect was generated by

2.1 Pix Firewall

This firewall is installed the border of ext.net.

3 Probability that the source address was spoofed

Low

The scanner is looking for a particular Trojan and needs to get the replies.

4 Description of the attack

This is a simple scan with the scanner looking for the SubSeven 2.1 Trojan.

5 Attack Mechanism

The scanner is se quentially polling the subnet looking for response on port 27374.

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

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

6 Correlation

For extra information on SubSeven and its variants see http://www.sans.org/y2k/subseven.htm

7 Evidence of active targeting

There is no active targeting on this network, just a meandering scan for the Trojan.

8 Severity

The standard formula for determining severity is as follows:

(Criticality + Lethality) - (System countermeasures + Network countermeasures)

Each element is sc ored from 1 to 5, 1 being low, 5 being high.

Criticality = 3. This is our internal network.

Lethality = 1 . Its not an attack.

System countermeasures = 2 . The system is up to date and all incoming mail is scanned for Trojans.

Network countermeasures = 3 . There is a strong filtering firewall between this subnet and the Internet.

Severity = (3 + 1) - (2 + 3) = -1.

9 Defensive recommendations

Keep the filters in place on the firewall and ensure virus signatures on electronic mail scanners are kept up to date.

10 Multiple choice test question

Which popular Trojan uses port 27374

a) Back Orifice
b) SubSeven
c) SubSeven 2.1
d) SMTP

Answer: (**c**)

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

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

Assignment Two – Evaluate an Attack

The following evaluation is of the nmap scanning and network vulnerability assessment tool. This tool is available from the following URL:

http://www.insecure.org/nmap/

The machine called "chunky" is a mandrake Linux server with two PIII 550e processors and 512 Mb Ram. (Hence the name) and we are scanning a Solaris server called ram on the same subnet.

First of all we try a UDP scan on a port that we know is not configured.

```
root@chunky:/data01/temp/SANS/scan# nmap -sU ram -p 124

<The –sU is a UDP scan with ram being the target and –p specifiying the port to be scanned>

Starting nmap V. 2.30BETA17 by fyodor@insecure.org ( www.insecure.org/nmap/ )
No ports open for host ram.X.com.au (10.12.1.14)
Nmap run completed -- 1 IP address (1 host up) scanned in 1 second

10:09:50.920914 chunky > ram.X.com.au: icmp: echo request
<First of all we send an ICMP echo request to ram>
10:09:50.920963 chunky.50902 > ram.X.com.au.www: . ack 806802502 win 3072
<Then an ACK to port 80 on ram>
10:09:50.921240 ram.X.com.au > chunky: icmp: echo reply (DF)
<ram responds with an ICMP echo reply>
10:09:50.921450 ram.X.com.au.www > chunky.50902: R 806802502:806802502(0) win 0 (DF)
<ram responds with a RESET>
10:09:51.223987 chunky.50882 > ram.X.com.au.124: udp 0
<chunky sends a zero byte UDP packet to port 124 on ram>
10:09:51.224443 ram.X.com.au > chunky: icmp: ram.X.com.au udp port 124 unreachable (DF)
<ram responds with an ICMP port unreachable>
```

All this seems pretty self explanatory with the exception of the ACK sent to port 80 on ram. On further investigation in the nmap man pages, it seems that the two methods nmap uses to determine whether the machine is alive or not is to ping the machine and to send an ACK to port 80. If the machine responds with a RESET then it is alive. A machine will respond with a RESET wheth er the port 80 is open or not as it has no established connection with the scanning machine. So we are sure that ram is alive so we send a 0byte UDP packet to port 124 on ram. Ram then responds with an ICMP port unreachable which tells us that ram does not have port 124 open.

Now we try a UDP scan on a port that we know is open.

```
root@chunky:/data01/temp/SANS/scan# nmap -sU ram -p 123

Starting nmap V. 2.30BETA17 by fyodor@insecure.org ( www.insecure.org/nmap/ )
Interesting ports on ram.X.com.au (10.12.1.14):
Port       State      Service
123/udp    open       ntp

Nmap run completed -- 1 IP address (1 host up) scanned in 1 second

10:11:16.632728 chunky > ram.X.com.au: icmp: echo request
<First of all we send an ICMP echo request to ram>
10:11:16.632849 chunky.58585 > ram.X.com.au.www: . ack 237238120 win 2048
<Then an ACK to port 80 on ram>
10:11:16.633056 ram.X.com.au > chunky: icmp: echo reply (DF)
<ram responds with an ICMP echo reply>
10:11:16.633248 ram.X.com.au.www > chunky.58585: R 237238120:237238120(0) win 0 (DF)
<ram responds with a RESET>
10:11:16.935357 chunky.58565 > ram.X.com.au.ntp: [len=0] [|ntp]
<chunky sends a zero byte UDP packet to port 123 on ram>
10:11:17.251585 chunky.58566 > ram.X.com.au.ntp: [len=0] [|ntp]
<chunky sends another zero byte UDP packet to port 123 on ram>
```


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

The trace above is very similar with the exception of two points. The fact that chunky sends the 0 byte UDP packet twice and the lack of ICMP response. The reason nmap sends the UDP packet twice is that UDP is a lossy connectionless protocol and the packet may have been lost in transit, so it sends the packet twice. The fact that ram does not respond with an ICMP port unreachable message leads us to assume that the port is open.

Then we try a TCP scan on a port that we know is not configured.

```
root@chunky:/data01/temp/SANS/scan# nmap -sS ram -p 21

<The –sS is a SYN or half open scan with ram being the target and –p specifiying the port to be
scanned>

Starting nmap V. 2.30BETA17 by fyodor@insecure.org ( www.insecure.org/nmap/ )
No ports open for host ram.X.com.au (10.12.1.14)
Nmap run completed -- 1 IP address (1 host up) scanned in 0 seconds

10:16:44.035658 chunky > ram.X.com.au: icmp: echo request
<First of all we send an ICMP echo request to ram>
10:16:44.035791 chunky.56026 > ram.X.com.au.www: . ack 719836970 win 4096
<Then an ACK to port 80 on ram>
10:16:44.036044 ram.X.com.au > chunky: icmp: echo reply (DF)
<ram responds with an ICMP echo reply>
10:16:44.036272 ram.X.com.au.www > chunky.56026: R 719836970:719836970(0) win 0 (DF)
<ram responds with a RESET>
10:16:44.338550 chunky.56006 > ram.X.com.au.ftp: S 49319495:49319495(0) win 4096
<we send a SYN to ram on port 21>
10:16:44.339097 ram.X.com.au.ftp > chunky.56006: R 0:0(0) ack 49319496 win 0 (DF)
<ram responds with a RESET>
```

The preamble to this scan is exactly the same as the previous traces. Once nmap has determined that the machine is alive it continues by sending a SYN to port 21 (ftp) on ram. Because ram is not listening on this port it responds with a RESET.

Then we try a TCP scan on a port that we know is open.

```
root@chunky:/data01/temp/SANS/scan# nmap -sS ram -p 22

<The –sS is a SYN or half open scan with ram being the target and –p specifiying the port to be
scanned>

Starting nmap V. 2.30BETA17 by fyodor@insecure.org ( www.insecure.org/nmap/ )
Interesting ports on ram.X.com.au (10.12.1.14):
Port       State       Service
22/tcp     open        ssh

Nmap run completed -- 1 IP address (1 host up) scanned in 0 seconds

10:15:23.003889 chunky > ram.X.com.au: icmp: echo request
<First of all we send an ICMP echo request to ram>
10:15:23.003939 chunky.56726 > ram.X.com.au.www: . ack 10853558 win 2048
<Then an ACK to port 80 on ram>
10:15:23.004260 ram.X.com.au > chunky: icmp: echo reply (DF)
<ram responds with an ICMP echo reply>
10:15:23.004491 ram.X.com.au.www > chunky.56726: R 10853558:10853558(0) win 0 (DF)
<ram responds with a RESET>
10:15:23.306480 chunky.56706 > ram.X.com.au.ssh: S 2569074901:2569074901(0) win 2048
<we send a SYN to ram on port 22>
10:15:23.307048 ram.X.com.au.ssh > chunky.56706: S 4071831402:4071831402(0) ack 2569074902 win 9112
<mss 536> (DF)
<ram responds with a SYN with an ACK of 2569074902 which is one more than the sequence number of the
previous packet>
10:15:23.307096 chunky.56706 > ram.X.com.au.ssh: R 2569074902:2569074902(0) win 0 (DF)
<chunky then responds with a RESET>
```

The difference with this is the fact that port 22 (ssh) is open on ram. On seeing the SYN arrive from chunky ram attempts the next phase of the three way handshake and returns a SYN ACK. Nmap is now satisfied that the port is open and sends a RESET to abruptly close the connection attempt.

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

Assignment Three – "Analyse This"

The following snort output has been collected from an IDS point on a network somewhere. I have no reference point as to the topology of the network or access to the rules used by the snort system.

Given these constraints I can offer the following observations and recommendations.

There was generally a lot of scanning for ports and hosts going on using a variety of scanning methods including null scans, SYN FIN scans and port scans.

```
05/28-01:42:15.020125  [**] Attempted Sun RPC high port access [**] 205.188.153.100:4000 ->
MY.NET.217.2:32771
05/28-01:43:14.949657  [**] Attempted Sun RPC high port access [**] 205.188.153.100:4000 ->
MY.NET.217.2:32771
05/28-01:44:14.872532  [**] Attempted Sun RPC high port access [**] 205.188.153.100:4000 ->
MY.NET.217.2:32771
05/28-01:45:14.827623  [**] Attempted Sun RPC high port access [**] 205.188.153.100:4000 ->
MY.NET.217.2:32771
05/28-01:48:14.665188  [**] Attempted Sun RPC high port access [**] 205.188.153.100:4000 ->
MY.NET.217.2:32771
05/28-01:49:14.595563  [**] Attempted Sun RPC high port access [**] 205.188.153.100:4000 ->
MY.NET.217.2:32771
05/28-01:50:14.620278  [**] Attempted Sun RPC high port access [**] 205.188.153.100:4000 ->
MY.NET.217.2:32771
05/28-01:52:14.418356  [**] Attempted Sun RPC high port access [**] 205.188.153.100:4000 ->
MY.NET.217.2:32771
05/28-01:53:14.374269  [**] Attempted Sun RPC high port access [**] 205.188.153.100:4000 ->
MY.NET.217.2:32771
```

While this may be RPC traffic it is more likely to be related to ICQ. The key is to evaluate the source port as well. I would recommend that ICQ is not a good thing to let into your network anyway.

```
05/28-13:08:24.127009  [**] External RPC call [**] 216.148.73.6:2666 -> MY.NET.100.130:111
05/28-13:08:24.127009  [**] External RPC call [**] 216.148.73.6:2666 -> MY.NET.100.130:111
```

This shows portmapper access from external address space. Surely a bad thing.

```
05/28-14:30:50.876461  [**] SUNRPC highport access! [**] MY.NET.253.12:43746 -> MY.NET.16.0:32771
05/28-14:30:51.185774  [**] SUNRPC highport access! [**] MY.NET.253.12:43747 -> MY.NET.16.0:32771
05/28-14:31:04.905230  [**] SUNRPC highport access! [**] MY.NET.253.12:43749 -> MY.NET.16.0:32771
05/28-14:31:05.245775  [**] SUNRPC highport access! [**] MY.NET.253.12:43750 -> MY.NET.16.0:32771
```

There seems to be all together too much RPC style traffic from external addresses to MY.NET. The firewall (if any) should be configured to block all this traffic. In fact a more restrictive firewall policy should be enforced on this gateway. Something along the lines of "All that is not explicitly allowed is denied".

```
05/16-22:32:56.988094  [**] Attempted Sun RPC high port access [**] 216.80.63.9:7777 ->
MY.NET.98.150:32771
05/16-22:51:08.989718  [**] Attempted Sun RPC high port access [**] 63.90.234.50:7777 ->
MY.NET.98.150:32771
05/16-22:51:12.892742  [**] Attempted Sun RPC high port access [**] 63.90.234.50:7777 ->
MY.NET.98.150:32771
05/16-22:51:16.103139  [**] Attempted Sun RPC high port access [**] 63.90.234.50:7777 ->
MY.NET.98.150:32771
05/16-22:51:16.337144  [**] Attempted Sun RPC high port access [**] 63.90.234.50:7777 ->
MY.NET.98.150:32771
05/16-22:51:16.726851  [**] Attempted Sun RPC high port access [**] 63.90.234.50:7777 ->
MY.NET.98.150:32771
05/16-22:51:17.626777  [**] Attempted Sun RPC high port access [**] 63.90.234.50:7777 ->
MY.NET.98.150:32771
05/16-22:51:18.497029  [**] Attempted Sun RPC high port access [**] 63.90.234.50:7777 ->
MY.NET.98.150:32771
05/16-22:51:20.231520  [**] Attempted Sun RPC high port access [**] 63.90.234.50:7777 ->
MY.NET.98.150:32771
05/16-22:51:21.072034  [**] Attempted Sun RPC high port access [**] 63.90.234.50:7777 ->
MY.NET.98.150:32771
05/16-22:51:22.427600  [**] Attempted Sun RPC high port access [**] 63.90.234.50:7777 ->
MY.NET.98.150:32771
```


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

```
05/16-22:51:23.759209  [**] Attempted Sun RPC high port access [**] 63.90.234.50:7777 ->
MY.NET.98.150:32771
```

And still more, this continues with a number of external sources. Some access to portmapper and other appear to be accessing typical high -port RPC services.

```
05/25-11:21:12.841511  [**] GIAC 000218 VA-CIRT port 34555 [**] 216.64.2.218:25 -> MY.NET.253.24:34555
05/25-11:21:13.082075  [**] GIAC 000218 VA-CIRT port 34555 [**] 216.64.2.218:25 -> MY.NET.253.24:34555
05/25-11:21:21.098854  [**] GIAC 000218 VA-CIRT port 34555 [**] 216.64.2.218:25 -> MY.NET.253.24:34555
05/25-11:21:25.442981  [**] GIAC 000218 VA-CIRT port 34555 [**] 216.64.2.218:25 -> MY.NET.253.24:34555
05/25-11:21:29.016239  [**] GIAC 000218 VA-CIRT port 34555 [**] 216.64.2.218:25 -> MY.NET.253.24:34555


05/25-01:47:17.966506  [**] GIAC 000218 VA-CIRT port 34555 [**] 209.38.76.60:113 -> MY.NET.6.34:34555
05/25-01:47:18.100379  [**] GIAC 000218 VA-CIRT port 34555 [**] 209.38.76.60:113 -> MY.NET.6.34:34555
05/25-01:47:18.100440  [**] GIAC 000218 VA-CIRT port 34555 [**] 209.38.76.60:113 -> MY.NET.6.34:34555
05/25-01:47:18.241045  [**] GIAC 000218 VA-CIRT port 34555 [**] 209.38.76.60:113 -> MY.NET.6.34:34555


05/25-04:00:48.153520  [**] GIAC 000218 VA-CIRT port 34555 [**] 130.114.200.6:53 -> MY.NET.1.8:34555
05/25-04:00:48.248262  [**] GIAC 000218 VA-CIRT port 34555 [**] 130.114.200.6:53 -> MY.NET.1.8:34555
05/25-04:00:48.266840  [**] GIAC 000218 VA-CIRT port 34555 [**] 130.114.200.6:53 -> MY.NET.1.8:34555
```

```
06/13-00:48:43.581467  [**] GIAC 000218 VA-CIRT port 35555 [**] 216.181.91.92:32788 -> MY.NET.100.37:35555
06/13-00:48:58.604373  [**] GIAC 000218 VA-CIRT port 35555 [**] 216.181.91.92:32788 -> MY.NET.100.37:35555
06/13-00:48:58.606810  [**] GIAC 000218 VA-CIRT port 35555 [**] 216.181.91.92:32788 -> MY.NET.100.37:35555
```


```
06/13-00:48:43.581467  [**] GIAC 000218 VA-CIRT port 35555 [**] 216.181.91.92:32788 -> MY.NET.100.37:35555
06/13-00:48:58.604373  [**] GIAC 000218 VA-CIRT port 35555 [**] 216.181.91.92:32788 -> MY.NET.100.37:35555
06/13-00:48:58.606810  [**] GIAC 000218 VA-CIRT port 35555 [**] 216.181.91.92:32788 -> MY.NET.100.37:35555
```


This traffic above could indicate an active “trinoo” server. The traffic in the first “VA -CIRT” detect above would have to be examined more carefully. My first reaction would be that this is normal traffic given the source po rts like 25 (smtp), 113 (ident) and 53 (dns). With the following traces I would be more suspicious given the high source port.

```
06/18-01:10:01.394862  [**] Tiny Fragments - Possible Hostile Activity [**] 63.236.34.174 -> MY.NET.1.8
06/18-01:10:01.394944  [**] Tiny Fragments - Possible Hostile Activity [**] 63.236.34.174 -> MY.NET.1.8
06/18-01:10:01.395084  [**] Tiny Fragments - Possible Hostile Activity [**] 63.236.34.174 -> MY.NET.1.8


05/23-15:24:38.099890  [**] Tiny Fragments - Possible Hostile Activity [**] 206.193.209.254 ->
MY.NET.219.58
05/23-15:24:42.932586  [**] Tiny Fragments - Possible Hostile Activity [**] 206.193.209.254 ->
MY.NET.219.58
05/23-15:24:43.245630  [**] Tiny Fragments - Possible Hostile Activity [**] 206.193.209.254 ->
MY.NET.219.58
05/23-15:24:44.262475  [**] Tiny Fragments - Possible Hostile Activity [**] 206.193.209.254 ->
MY.NET.219.58
05/23-15:24:46.175629  [**] Tiny Fragments - Possible Hostile Activity [**] 206.193.209.254 ->
MY.NET.219.58
05/23-15:24:47.995867  [**] Tiny Fragments - Possible Hostile Activity [**] 206.193.209.254 ->
MY.NET.219.58
05/23-15:24:48.590209  [**] Tiny Fragments - Possible Hostile Activity [**] 206.193.209.254 ->
MY.NET.219.58
```

Tiny fragment can be a sign of evasion techniques. By splitting the packet down into tiny pieces it can evade some firewalls by having the header split over multiple packets. Without the benefit of tcpdump traces or extra data it is very hard to determine what is going on here.

```
06/13-10:26:37.292191  [**] Happy 99 Virus [**] 207.172.132.67:1038 -> MY.NET.253.52:25
05/25-09:53:44.364111  [**] Happy 99 Virus [**] 207.172.145.30:1294 -> MY.NET.253.51:25
05/25-09:53:44.364111  [**] Happy 99 Virus [**] 207.172.145.30:1294 -> MY.NET.253.51:25
```

I would recommend that all inbound elect ronic mail transmissions be checked for virus or other malicious content. This signature shows at least the possibility of infected files inbound.

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

```
05/28-13:18:29.397835  [**] SMB Name Wildcard [**] MY.NET.101.160:137 -> MY.NET.101.192:137
05/28-13:32:55.095543  [**] SMB Name Wildcard [**] MY.NET.101.160:137 -> MY.NET.101.192:137
05/28-13:32:56.586288  [**] SMB Name Wildcard [**] MY.NET.101.160:137 -> MY.NET.101.192:137
05/28-13:32:58.088429  [**] SMB Name Wildcard [**] MY.NET.101.160:137 -> MY.NET.101.192:137
05/28-16:06:54.050136  [**] SMB Name Wildcard [**] MY.NET.101.160:137 -> MY.NET.101.192:137
05/28-16:06:55.837485  [**] SMB Name Wildcard [**] MY.NET.101.160:137 -> MY.NET.101.192:137
```

It is unwise to allow 137 and friends into your network. This tra ffic would certainly be subject to more thorough investigation.

```
06/01-09:39:54.804923  [**] SMB Name Wildcard [**] 192.168.7.2:137 -> MY.NET.14.1:137
06/01-09:41:20.947779  [**] SMB Name Wildcard [**] 192.168.7.2:137 -> MY.NET.14.1:137
06/01-09:41:22.446465  [**] SMB Name Wildcard [**] 192.168.7.2:137 -> MY.NET.14.1:137
```

```
05/22-13:11:40.249460  [**] SMB Name Wildcard [**] 63.208.207.71:137 -> MY.NET.100.130:137
05/22-13:13:11.242134  [**] SMB Name Wildcard [**] 63.208.207.71:137 -> MY.NET.100.130:137
05/22-13:14:12.190534  [**] SMB Name Wildcard [**] 63.208.207.71:137 -> MY.NET.100.130:137
05/22-13:18:38.578417  [**] SMB Name Wildcard [**] 63.208.207.71:137 -> MY.NET.100.130:137
05/22-13:18:40.081692  [**] SMB Name Wildcard [**] 63.208.207.71:137 -> MY.NET.100.130:137
05/22-13:18:43.203497  [**] SMB Name Wildcard [**] 63.208.207.71:137 -> MY.NET.100.130:137
05/22-13:18:44.678391  [**] SMB Name Wildcard [**] 63.208.207.71:137 -> MY.NET.100.130:137
05/22-13:24:05.067158  [**] SMB Name Wildcard [**] 63.208.207.71:137 -> MY.NET.100.130:137
```

This is an example of SMB access to the internal networks. Could be a result of scanning for open windows devices or just harmless traffic emanating from any windows machine accessing your network.

```
05/26-08:01:15.103192  [**] SNMP public access [**] MY.NET.97.59:1070 -> MY.NET.101.192:161
05/26-08:01:17.296998  [**] SNMP public access [**] MY.NET.97.59:1071 -> MY.NET.101.192:161
05/26-08:01:17.523612  [**] SNMP public access [**] MY.NET.97.59:1072 -> MY.NET.101.192:161
05/26-08:01:17.714759  [**] SNMP public access [**] MY.NET.97.59:1073 -> MY.NET.101.192:161
05/26-08:01:19.910111  [**] SNMP public access [**] MY.NET.97.59:1074 -> MY.NET.101.192:161
```

It appears that some network administrator has been lazy and left a network device with the default SNMP "PUBLIC" string. It is not wise to leave this default value in place.

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

Last Updated: April 28th, 2017

Upcoming Training

Click Here to {Get CERTIFIED!}

SANS Security West 2017	San Diego, CA	May 09, 2017 - May 18, 2017	Live Event
SANS Zurich 2017	Zurich, Switzerland	May 15, 2017 - May 20, 2017	Live Event
Mentor Session - SEC503	Oceanside, CA	May 29, 2017 - Jun 29, 2017	Mentor
Security Operations Center Summit & Training	Washington, DC	Jun 05, 2017 - Jun 12, 2017	Live Event
SANS Houston 2017	Houston, TX	Jun 05, 2017 - Jun 10, 2017	Live Event
SANS Columbia, MD 2017	Columbia, MD	Jun 26, 2017 - Jul 01, 2017	Live Event
SANS London July 2017	London, United Kingdom	Jul 03, 2017 - Jul 08, 2017	Live Event
SANSFIRE 2017	Washington, DC	Jul 22, 2017 - Jul 29, 2017	Live Event
SANSFIRE 2017 - SEC503: Intrusion Detection In-Depth	Washington, DC	Jul 24, 2017 - Jul 29, 2017	vLive
SANS San Antonio Fall 2017	San Antonio, TX	Aug 06, 2017 - Aug 11, 2017	Live Event
SANS Boston 2017	Boston, MA	Aug 07, 2017 - Aug 12, 2017	Live Event
SANS Adelaide 2017	Adelaide, Australia	Aug 21, 2017 - Aug 26, 2017	Live Event
SANS Virginia Beach 2017	Virginia Beach, VA	Aug 21, 2017 - Sep 01, 2017	Live Event
SANS Network Security 2017	Las Vegas, NV	Sep 10, 2017 - Sep 17, 2017	Live Event
SANS vLive - SEC503: Intrusion Detection In-Depth	SEC503 - 201709,	Sep 11, 2017 - Oct 18, 2017	vLive
SANS Baltimore Fall 2017	Baltimore, MD	Sep 25, 2017 - Sep 30, 2017	Live Event
Baltimore September 2017 - SEC503: Intrusion Detection In-Depth	Baltimore, MD	Sep 25, 2017 - Sep 30, 2017	vLive
SANS London September 2017	London, United Kingdom	Sep 25, 2017 - Sep 30, 2017	Live Event
Community SANS Scottsdale SEC503	Scottsdale, AZ	Oct 02, 2017 - Oct 07, 2017	Community SANS
Community SANS Ottawa SEC503	Ottawa, ON	Oct 16, 2017 - Oct 21, 2017	Community SANS
SANS October Singapore 2017	Singapore, Singapore	Oct 16, 2017 - Oct 28, 2017	Live Event
SANS San Diego Fall 2017	San Diego, CA	Oct 30, 2017 - Nov 04, 2017	Live Event
SANS Seattle 2017	Seattle, WA	Oct 30, 2017 - Nov 04, 2017	Live Event
San Diego Fall 2017 - SEC503: Intrusion Detection In-Depth	San Diego, CA	Oct 30, 2017 - Nov 04, 2017	vLive
SANS OnDemand	Online	Anytime	Self Paced
SANS SelfStudy	Books & MP3s Only	Anytime	Self Paced