Global Information Assurance Certification Paper

Practical Submission for GCIA

*** Northcutt, good job, shows what can be some with a firewall. Nice research on the source hosts. Free form process, but Drew pulls it off. 82 *

Drew Brunson

Detect #1

Here's a nice little host scan, possibly from Germany, of my class C network on port 8080 beginning with a scan to the BSD broadcast address, maybe hoping to get a clue as to my operating systems (I'm wondering if my logger missed a scan to .255?). This is probably probing for an http proxy. There is a random skip of addresses throughout that might be an attempt to evade detection, but missing some of the packets because of other traffic seems more likely because of the way that the src port increments in sequence with the host number. That might be considered one of its signatures since it starts with src port 42601 to host 0 then 42602 to host 1 etc. The sequence numbers look okay, but I sure wouldn't expect the src port numbers to be so clean unless they were crafted.

The first scan took place in a few seconds but then an hour and 17 minutes later the same host comes through with a scan of the entire Internet.

Doing a dig on this address returns nothing but a dig on 194.78.174.0 returns:

174.78.194.in-addr.arpa. 86400 SOA ns1.skynet.be. dnsmaster.skynet.

I would assign this a relatively low priority because 1) the firewall blocked it; 2) it appears to be part of a general scan looking for vulnerable hosts; 3) no information about my internal network was released. The danger here is that because I returned no information about my hosts those numbers could potentially be used in a DOS relying on hosts that are addressable but not responding on the Internet.

Mar 29 00:32:05 fwall 14 deny: TCP from 194.78.174.5.42601 to xxx.xxx.xxx.0.8080 seq 44E87CD4, ack 0x0, win 8760, SYN
Mar 29 00:32:05 fwall 14 deny: TCP from 194.78.174.5.42602 to xxx.xxx.xxx.1.8080 seq 44EA5ACF, ack 0x0, win 8760, SYN
Mar 29 00:32:05 fwall 14 deny: TCP from 194.78.174.5.42603 to xxx.xxx.xxx.2.8080 seq 44EAC46F, ack 0x0, win 8760, SYN
Mar 29 00:32:05 fwall 14 deny: TCP from 194.78.174.5.42604 to xxx.xxx.xxx.3.8080 seq 44EC8E52, ack 0x0, win 8760, SYN
Mar 29 00:32:06 fwall 14 deny: TCP from 194.78.174.5.42605 to xxx.xxx.xxx.4.8080 seq 44EDCF73, ack 0x0, win 8760, SYN
(Truncated)
Mar 29 00:32:20 fwall 14 deny: TCP from 194.78.174.5.42851 to xxx.xxx.xxx.250.8080 seq 45FB3362, ack 0x0, win 8760, SYN
Mar 29 00:32:20 fwall 14 deny: TCP from 194.78.174.5.42852 to xxx.xxx.xxx.251.8080

seq 45FD1652, ack 0x0, win 8760, SYN
Mar 29 00:32:20 fwall 14 deny: TCP from 194.78.174.5.42854 to xxx.xxx.xxx.253.8080
seq 45FFADD4, ack 0x0, win 8760, SYN
Mar 29 00:32:20 fwall 14 deny: TCP from 194.78.174.5.42855 to xxx.xxx.xxx.254.8080
seq 45FFCF04, ack 0x0, win 8760, SYN
Mar 29 01:19:26 fwall 14 deny: TCP from 194.78.174.5.50500 to 255.255.255.255.8080
seq F9F2C84F, ack 0x0, win 8760, SYN

Detect #2

Well this appears to be a down and dirty scan of my network for the Deep Throat trojan from usr11-dialup335.mix1.Irving.cw.net. Since I'm successfully denying the probes, I'm not worried about this. Since the prober doesn't seem to be trying to hide anything I am assuming that the skipped nodes were simply missed. I did let the ISP know this happened. Probably just someone who picked up a script from the Internet.

Mar 30 18:53:12 fwall 23 deny: UDP from 166.62.210.89.60000 to xxx.xxx.xxx.1.2140
Mar 30 18:53:12 fwall 23 deny: UDP from 166.62.210.89.60000 to xxx.xxx.xxx.2.2140
Mar 30 18:53:12 fwall 23 deny: UDP from 166.62.210.89.60000 to xxx.xxx.xxx.3.2140
Mar 30 18:53:12 fwall 23 deny: UDP from 166.62.210.89.60000 to xxx.xxx.xxx.4.2140
(. . . truncated)
Mar 30 18:53:31 fwall 23 deny: UDP from 166.62.210.89.60000 to xxx.xxx.xxx.247.2140
Mar 30 18:53:31 fwall 23 deny: UDP from 166.62.210.89.60000 to xxx.xxx.xxx.248.2140
Mar 30 18:53:32 fwall 23 deny: UDP from 166.62.210.89.60000 to xxx.xxx.xxx.250.2140
Mar 30 18:53:32 fwall 23 deny: UDP from 166.62.210.89.60000 to xxx.xxx.xxx.252.2140
Mar 30 18:53:32 fwall 12 deny: UDP from 166.62.210.89.60000 to xxx.xxx.xxx.255.2140

. Detect #3

Two separate queries to my Class C trying to determine what hosts I might have responding for further probing. These probes might also be looking for registered addresses that do not respond and that can be used to hide the sources in future DDOS attacks. From Korea, they hit both the older BSD .0 broadcast address and also the 255 broadcast address. I have seen several scans recently from here and I'm wondering if a ramp-up is in progress aimed at a massive DDOS. From my network perspective I'm not too worried, but considering what's been going on with the stock market, we

may be prime for a demoralizing attack if major centers can be shut down during trading hours.

Apr 17 00:33:28 fwall 16 deny: icmp from 137.68.110.178 to xxx.xxx.7.0 type Echo Request
Apr 17 00:33:28 fwall 15 deny: icmp from 137.68.110.178 to xxx.xxx.7.255 type Echo Request
Apr 17 01:34:38 fwall 16 deny: icmp from 137.68.110.178 to xxx.xxx.7.0 type Echo Request
Apr 17 01:34:38 fwall 15 deny: icmp from 137.68.110.178 to xxx.xxx.7.255 type Echo Request

178.110.68.137.in-addr.arpa. 36235 PTR sail.kaist.ac.kr.

;; AUTHORITY RECORDS:
68.137.IN-ADDR.ARPA. 511431 NS NS.kaist.ac.kr.
68.137.IN-ADDR.ARPA. 511431 NS NS.KAIST.KR.APAN.NET.

Detect #4

When I first was given the responsibility of setting up our firewall, I had absolutely no guidance on security and was just about the only one in our corporation with any experience working as a UNIX sys admin (although a few years later there are a multitude of "experts"). One thing I did was try to read as much as I could in what was available on how to secure your network. Thank God for people who write security books! Anyway to make a boring story short, one of the recommendations was to limit outside hosts from initiating an ftp data channel connection unless really necessary. And that's what I do. This example shows one of my internal hosts asking for an ftp connection to innocent host. They complete the three way handshake and exchange a little more data on the command port. At the point the innocent host (not being informed apparently that this is to be a PASV connection, or not being able to respond appropriately) tries to initiate the data channel.

After several attempts, my system sends a FIN-ACK to end everything and then, after another attempt from Mr. Innocent, sends a RST. This may not be a detect in the bad guy sense, but it's nice to have verification every once in a while that your rules are being applied the way you think they should be.

Apr 12 23:54:43 fwall 1 permit: TCP from mysubnet.2.62821 to innocent.119.21 seq 39B58200, ack 0x0, win 16616, SYN
Apr 12 23:54:44 fwall 2 permit: TCP from innocent.119.21 to mysubnet.2.62821 seq 307331A8, ack 0x39B58201, win 8760, SYN ACK
Apr 12 23:54:44 fwall 1 permit: TCP from mysubnet.2.62821 to innocent.119.21 seq 39B58201, ack 0x307331A9, win 17520, ACK
Apr 12 23:54:48 fwall 2 permit: TCP from innocent.119.21 to mysubnet.2.62821 seq 307331A9, ack 0x39B58201, win 8760, PUSH ACK , 28 bytes

Apr 12 23:54:48 fwall 1 permit: TCP from mysubnet.2.62821 to innocent.119.21 seq 39B58201, ack 0x307331C5, win 17520, PUSH ACK , 16 bytes
Apr 12 23:54:48 fwall 2 permit: TCP from innocent.119.21 to mysubnet.2.62821 seq 307331C5, ack 0x39B58211, win 8760, ACK
Apr 12 23:54:48 fwall 2 permit: TCP from innocent.119.21 to mysubnet.2.62821 seq 307331C5, ack 0x39B58211, win 8760, PUSH ACK , 68 bytes
Apr 12 23:54:49 fwall 1 permit: TCP from mysubnet.2.62821 to innocent.119.21 seq 39B58211, ack 0x30733209, win 17520, ACK
(Truncated . . .)
Apr 12 23:54:52 fwall 25 deny: TCP from innocent.119.20 to mysubnet.2.62823 seq 308E7068, ack 0x0, win 8760, SYN
Apr 12 23:54:52 fwall 2 permit: TCP from innocent.119.21 to mysubnet.2.62821 seq 30733257, ack 0x39B5825B, win 8760, ACK
Apr 12 23:54:55 fwall 25 deny: TCP from innocent.119.20 to mysubnet.2.62823 seq 308E7068, ack 0x0, win 8760, SYN
Apr 12 23:55:00 fwall 25 deny: TCP from innocent.119.20 to mysubnet.2.62823 seq 308E7068, ack 0x0, win 8760, SYN
Apr 12 23:55:07 fwall 25 deny: TCP from innocent.119.20 to mysubnet.2.62823 seq 308E7068, ack 0x0, win 8760, SYN
Apr 12 23:55:52 fwall 25 deny: TCP from innocent.119.20 to mysubnet.2.62823 seq 308E7068, ack 0x0, win 8760, SYN
Apr 12 23:55:52 fwall 1 permit: TCP from mysubnet.2.62821 to innocent.119.21 seq 39B5825B, ack 0x30733257, win 17520, FIN ACK
Apr 12 23:55:52 fwall 2 permit: TCP from innocent.119.21 to mysubnet.2.62821 seq 30733257, ack 0x39B5825C, win 8760, ACK
Apr 12 23:56:04 fwall 25 deny: TCP from innocent.119.20 to mysubnet.2.62823 seq 308E7068, ack 0x0, win 8760, SYN
Apr 12 23:57:46 fwall 2 permit: TCP from innocent.119.21 to mysubnet.2.62821 seq 30733257, ack 0x39B5825C, win 8760, PUSH ACK , 52 bytes
Apr 12 23:57:47 fwall 1 permit: TCP from mysubnet.2.62821 to innocent.119.21 seq 39B5825C, ack 0x0, win 17520, RST , 22 bytes

Detect #5

This appears to be from a cable modem out of ntl.com (21.20.252.62.in-addr.arpa. 43200 PTR pc21-gui3.cable.ntl.com.) and appears to match the signaature for a scan for the Hack 'a' Tack trojan. These probably crafted set of packets skip in a not quasi- random pattern through most of the scan, but either I missed a bunch at the end or the scripter did something weird. I know these packets didn't get through because I don't have machines there. Although there is obvious intent here, I don't think there's much danger from this probe.

Mar 30 09:10:55 fwall 20 deny: UDP from 62.252.20.21.31790 to xxx.xxx.xxx.1.31789
Mar 30 09:10:56 fwall 20 deny: UDP from 62.252.20.21.31790 to xxx.xxx.xxx.2.31789

Mar 30 09:10:56 fwall 20 deny: UDP from 62.252.20.21.31790 to xxx.xxx.xxx.3.31789
Mar 30 09:10:56 fwall 20 deny: UDP from 62.252.20.21.31790 to xxx.xxx.xxx.4.31789
(. . .)
Mar 30 09:11:59 fwall 20 deny: UDP from 62.252.20.21.31790 to xxx.xxx.xxx.175.31789
Mar 30 09:11:59 fwall 20 deny: UDP from 62.252.20.21.31790 to xxx.xxx.xxx.193.31789
Mar 30 09:12:00 fwall 20 deny: UDP from 62.252.20.21.31790 to xxx.xxx.xxx.203.31789
Mar 30 09:12:00 fwall 20 deny: UDP from 62.252.20.21.31790 to xxx.xxx.xxx.210.31789
Mar 30 09:12:01 fwall 20 deny: UDP from 62.252.20.21.31790 to xxx.xxx.xxx.232.31789
Mar 30 09:12:02 fwall 20 deny: UDP from 62.252.20.21.31790 to xxx.xxx.xxx.251.31789

Detect #6

Here's a nice little udp scan from a dsl connection (r111-28-dsl.sea.lightrealm.net) looking for hosts listening on the netbios name service port. With so many sites installing NT servers with unprotected shares and unprotected always-on PCs, this is probably not a bad bet for finding someone to compromise. Its primary signature is the consistent src port which matches the dst port, but it also makes two probes to port 137 on each dst host. I think there's a strong indication of intent and if I had any dialup users on my network, I would have a strong interest in finding out the status of their shares.

Apr 15 10:54:12 fwall 20 deny: UDP from 216.122.28.111.137 to xxx.xxx.xxx.1.137
Apr 15 10:54:13 fwall 20 deny: UDP from 216.122.28.111.137 to xxx.xxx.xxx.1.137
Apr 15 10:54:27 fwall 20 deny: UDP from 216.122.28.111.137 to xxx.xxx.xxx.2.137
Apr 15 10:54:27 fwall 20 deny: UDP from 216.122.28.111.137 to xxx.xxx.xxx.2.137
Apr 15 10:54:38 fwall 20 deny: UDP from 216.122.28.111.137 to xxx.xxx.xxx.3.137
Apr 15 10:54:56 fwall 20 deny: UDP from 216.122.28.111.137 to xxx.xxx.xxx.4.137
Apr 15 10:55:01 fwall 20 deny: UDP from 216.122.28.111.137 to xxx.xxx.xxx.4.137
(. . .)
Apr 15 11:48:29 fwall 20 deny: UDP from 216.122.28.111.137 to xxx.xxx.xxx.250.137
Apr 15 11:48:29 fwall 20 deny: UDP from 216.122.28.111.137 to xxx.xxx.xxx.250.137
Apr 15 11:48:39 fwall 20 deny: UDP from 216.122.28.111.137 to xxx.xxx.xxx.251.137
Apr 15 11:48:42 fwall 20 deny: UDP from 216.122.28.111.137 to xxx.xxx.xxx.251.137
Apr 15 11:48:51 fwall 20 deny: UDP from 216.122.28.111.137 to xxx.xxx.xxx.252.137

Apr 15 11:48:53 fwall 20 deny: UDP from 216.122.28.111.137 to xxx.xxx.xxx.252.137
Apr 15 11:49:04 fwall 20 deny: UDP from 216.122.28.111.137 to xxx.xxx.xxx.253.137
Apr 15 11:49:17 fwall 20 deny: UDP from 216.122.28.111.137 to xxx.xxx.xxx.254.137

111.28.122.216.in-addr.arpa. 1787 PTR r111-28-dsl.sea.lightrealm.net.

Detect #7

Here's a nice little scan from what appears to be in a Korean net providers address space. It appears to be looking for hosts listening on the nfs mount port. A very quick scan, it seems to increase host numbers, maybe to avoid setting off a NIDS that is looking for sequential host numbers (although I might also have just missed recording them) but then jumps from node 34 to 238 and then continues absolutely sequentially. The sequence numbers are funny since they increment and decrement abnormally, but the src port numbers look ok.

Although I don't think that any packets actually got through the firewall to query hosts (I don't see any RST or SYN/ACK being returned), I can't be absolutely sure and because of that, the source of the scan, and the vulnerability of what it is looking for I would give this a pretty high degree of risk. I will want to make sure no hosts are listening on 635 unless absolutely necessary and that I don't have any users who have created world mountable exports. I would also want to re-examine the trust relationships established by machines who need 635.

Mar 30 23:50:47 fwall 23 deny: TCP from 210.92.35.5.2394 to mynet.2.635 seq D2E6B162, ack 0x0, win 32120, SYN
Mar 30 23:50:47 fwall 23 deny: TCP from 210.92.35.5.2398 to mynet.6.635 seq D2FB11B9, ack 0x0, win 32120, SYN
Mar 30 23:50:47 fwall 23 deny: TCP from 210.92.35.5.2401 to mynet.9.635 seq D36B7ABF, ack 0x0, win 32120, SYN
Mar 30 23:50:47 fwall 23 deny: TCP from 210.92.35.5.2405 to mynet.13.635 seq D3992D53, ack 0x0, win 32120, SYN
Mar 30 23:50:48 fwall 23 deny: TCP from 210.92.35.5.2408 to mynet.16.635 seq
(. . .)
Mar 30 23:50:48 fwall 23 deny: TCP from 210.92.35.5.2426 to mynet.34.635 seq D3996FCB, ack 0x0, win 32120, SYN
Mar 30 23:50:48 fwall 23 deny: TCP from 210.92.35.5.2630 to mynet.238.635 seq D317D938, ack 0x0, win 32120, SYN
Mar 30 23:50:48 fwall 23 deny: TCP from 210.92.35.5.2631 to mynet.239.635 seq D395CEDA, ack 0x0, win 32120, SYN
Mar 30 23:50:48 fwall 23 deny: TCP from 210.92.35.5.2632 to mynet.240.635 seq D384BD0D, ack 0x0, win 32120, SYN

Mar 30 23:50:48 fwall 23 deny: TCP from 210.92.35.5.2633 to mynet.241.635 seq D39EEDFA, ack 0x0, win 32120, SYN
Mar 30 23:50:48 fwall 23 deny: TCP from 210.92.35.5.2634 to mynet.242.635 seq D2C60831, ack 0x0, win 32120, SYN
(. . .)
Mar 30 23:50:49 fwall 23 deny: TCP from 210.92.35.5.2646 to mynet.254.635 seq D3121A2B, ack 0x0, win 32120, SYN

92.210.in-addr.arpa. 43200 SOA ns.krnic.net. domain.krnic.net.

Detect #8

The earliest log I have on this starts on April 12. At first glance, it looks like my address space was mapped for addresses not actually in use and they are being used to hide the real attacker. I normally would expect to see this in response to an unexpected SYN-ACK maybe in connection with a SYN flood. I caught this after seeing the report on GIAC 4-19 and noticed a couple of other victim hosts involved other than those already reported – 176 and 180 – also notice that the attacker apparently crafted 255.255.255.255 as the src on at least two packets. Although we have correlation that at least two separate address spaces are being used in this attack it's not clear exactly what is happening. But, because the packets are coming in very slowly, uncharacteristic of a SYN flood, I am wondering if this is really a low and slow stealth mapping of the involved networks by hosts on 212.108.4. A dig returns no information on the host but gives the SOA as ripe.net. Ripe.net reports that this address space actually is used by Comned Networks B.V.out of Amsterdam. Other DNS queries return no information, but a ping does find the hosts.

Apr 12 22:43:54 fwall 2 permit: TCP from 212.108.4.152.80 to mysubnet.58.28645 seq 0, ack 0x13E1E35B, win 0, RST ACK
Apr 13 06:44:34 fwall 2 permit: TCP from 212.108.4.152.80 to mysubnet.38.64575 seq 0, ack 0x9CCD4542, win 0, RST ACK
Apr 13 06:49:27 fwall 2 permit: TCP from 212.108.4.153.80 to mysubnet.99.64041 seq 0, ack 0xAB3F531E, win 0, RST ACK
(. . .)
Apr 13 09:21:39 fwall 25 deny: TCP from 212.108.4.152.80 to 255.255.255.255.55789 seq 0, ack 0x594DB229, win 0, RST ACK
(. . .)
Apr 14 09:30:20 fwall 2 permit: TCP from 212.108.4.152.80 to mysubnet.14.16445 seq 0, ack 0xB786E558, win 0, RST ACK
Apr 14 12:14:43 fwall 2 permit: TCP from 212.108.4.153.80 to mysubnet.101.42813 seq 0, ack 0xD186870F, win 0, RST ACK
Apr 14 18:14:35 fwall 2 permit: TCP from 212.108.4.153.80 to mysubnet.79.64964 seq 0, ack 0x43F2D1D, win 0, RST ACK
Apr 14 20:52:32 fwall 2 permit: TCP from 212.108.4.152.80 to mysubnet.103.41489 seq 0, ack 0xFE4FFB5E, win 0, RST ACK
Apr 15 01:22:16 fwall 2 permit: TCP from 212.108.4.153.80 to mysubnet.117.59937

seq 0, ack 0x6D5AFD36, win 0, RST ACK
Apr 15 01:43:23 fwall 2 permit: TCP from 212.108.4.152.80 to mysubnet.21.26763
seq 0, ack 0x5EC4867, win 0, RST ACK
(. . .)
Apr 15 18:58:46 fwall 2 permit: TCP from 212.108.4.154.80 to mysubnet.15.23419
seq 0, ack 0xB87F5E6B, win 0, RST ACK
Apr 16 03:29:30 fwall 2 permit: TCP from 212.108.4.154.80 to mysubnet.35.618
seq 0, ack 0x27B15F2F, win 0, RST ACK
Apr 16 03:40:43 fwall 2 permit: TCP from 212.108.4.153.80 to mysubnet.66.25936
seq 0, ack 0x8ABB114C, win 0, RST ACK
(. . .)
Apr 16 17:23:45 fwall 2 permit: TCP from 212.108.4.153.80 to mysubnet.64.6677
seq 0, ack 0x4072C95F, win 0, RST ACK
Apr 16 21:39:02 fwall 2 permit: TCP from 212.108.4.152.80 to mysubnet.79.50083
seq 0, ack 0x98BDF647, win 0, RST ACK
Apr 17 00:35:43 fwall 2 permit: TCP from 212.108.4.154.80 to mysubnet.100.42042
seq 0, ack 0xAA2BEC08, win 0, RST ACK
Apr 17 00:41:01 fwall 2 permit: TCP from 212.108.4.152.80 to mysubnet.46.6952
seq 0, ack 0x15FF531F, win 0, RST ACK
Apr 17 00:43:05 fwall 2 permit: TCP from 212.108.4.154.80 to mysubnet.55.7616
seq 0, ack 0x6C5B01A, win 0, RST ACK
Apr 17 05:59:50 fwall 25 deny: TCP from 212.108.4.176.80 to
255.255.255.255.24662 seq 0, ack 0x23A54A28, win 0, RST ACK
(. . .)
Apr 17 19:56:55 fwall 1 permit: TCP from 212.108.4.180.80 to mysubnet.83.11863
seq 0, ack 0xE4403D6D, win 0, RST ACK
Apr 17 22:04:56 fwall 1 permit: TCP from 212.108.4.152.80 to mysubnet.126.5287
seq 0, ack 0xA5D3990C, win 0, RST ACK
Apr 18 00:33:12 fwall 1 permit: TCP from 212.108.4.176.80 to mysubnet.124.30276
seq 0, ack 0x2B420F1C, win 0, RST ACK
(. . .)
Apr 18 21:49:53 fwall 1 permit: TCP from 212.108.4.152.80 to mysubnet.110.44109
seq 0, ack 0x52310439, win 0, RST ACK
Apr 19 07:49:33 fwall 1 permit: TCP from 212.108.4.178
.80 to mysubnet.68.52107 seq 0, ack 0x65D72F6A, win 0, RST ACK
Apr 19 08:51:41 fwall 1 permit: TCP from 212.108.4.178
.80 to mysubnet.31.16981 seq 0, ack 0xFB559B4F, win 0, RST ACK

Detect #9

Here's a very fast little scripted scan from a dialup user at mindspring.net looking for snmp information. It's primary signature is the src port. It starts with a scan of the broadcast address for my subnet and starts decrementing (with some misses) down to host 25. The misses could have been intentional, but since this occurred shortly after I returned from the SANS conference and I wasn't logging as aggressively as I am today I kind of think that maybe these UDP packets just got lost. I am glad that I don't allow

snmp traffic from the Internet. Because I block snmp traffic, I'm don't have high concern about this scan. Although it went to the 255 broadcast address it, apparently, didn't try the older bsd .0 broadcast address.

Mar 27 12:49:50 fwall 14 deny: UDP from 209.86.5.170.1027 to mysubnet.255.161
Mar 27 12:49:50 fwall 14 deny: UDP from 209.86.5.170.1027 to mysubnet.254.161
Mar 27 12:49:51 fwall 14 deny: UDP from 209.86.5.170.1027 to mysubnet.253.161
Mar 27 12:49:51 fwall 14 deny: UDP from 209.86.5.170.1027 to mysubnet.252.161
Mar 27 12:49:51 fwall 14 deny: UDP from 209.86.5.170.1027 to mysubnet.251.161
Mar 27 12:49:51 fwall 14 deny: UDP from 209.86.5.170.1027 to mysubnet.250.161
Mar 27 12:49:51 fwall 14 deny: UDP from 209.86.5.170.1027 to mysubnet.249.161
(. . .)
Mar 27 12:49:56 fwall 14 deny: UDP from 209.86.5.170.1027 to mysubnet.27.161
Mar 27 12:49:56 fwall 14 deny: UDP from 209.86.5.170.1027 to mysubnet.26.161
Mar 27 12:49:56 fwall 14 deny: UDP from 209.86.5.170.1027 to mysubnet.25.161

170.5.86.209.in-addr.arpa. 86400 PTR user-38lc1da.dialup.mindspring.com.

Detect #10

Hmmmm, someone on my ISP's address space is probing my firewall to see if it will respond on the telnet port. Whoever it is seems to have crafted the packets since the sequence numbers are all identical and the src ports are all the same. In addition the four probes hit pretty quickly for a normal telnet connection attempt. I definitely will be in touch with them since the source does not match their normal pattern for client naming and possibly they have someone inside or a compromised host doing something I would hope they don't approve of. I rank this at a moderate risk since I am denying it even though this box will respond to telnet under certain conditions.

Apr 17 08:50:00 fwall 25 deny: TCP from 209.156.190.95.16666 to fwall.23 seq 862F, ack 0x0, win 8192, SYN
Apr 17 08:50:02 fwall 25 deny: TCP from 209.156.190.95.16666 to fwall.23 seq 862F, ack 0x0, win 8192, SYN
Apr 17 08:50:09 fwall 25 deny: TCP from 209.156.190.95.16666 to fwall.23 seq 862F, ack 0x0, win 8192, SYN
Apr 17 08:50:25 fwall 25 deny: TCP from 209.156.190.95.16666 to fwall.23 seq 862F, ack 0x0, win 8192, SYN

95.190.156.209.in-addr.arpa. 3563 PTR Woodlands95.splitrock.net

Last Updated: July 23rd, 2017

Upcoming Training

SANS San Antonio 2017	San Antonio, TX	Aug 06, 2017 - Aug 11, 2017	Live Event
SANS Boston 2017	Boston, MA	Aug 07, 2017 - Aug 12, 2017	Live Event
SANS Virginia Beach 2017	Virginia Beach, VA	Aug 21, 2017 - Sep 01, 2017	Live Event
SANS Adelaide 2017	Adelaide, Australia	Aug 21, 2017 - Aug 26, 2017	Live Event
SANS Network Security 2017	Las Vegas, NV	Sep 10, 2017 - Sep 17, 2017	Live Event
SANS vLive - SEC503: Intrusion Detection In-Depth	SEC503 - 201709,	Sep 11, 2017 - Oct 18, 2017	vLive
SANS London September 2017	London, United Kingdom	Sep 25, 2017 - Sep 30, 2017	Live Event
SANS Baltimore Fall 2017	Baltimore, MD	Sep 25, 2017 - Sep 30, 2017	Live Event
Baltimore Fall 2017 - SEC503: Intrusion Detection In-Depth	Baltimore, MD	Sep 25, 2017 - Sep 30, 2017	vLive
Community SANS Scottsdale SEC503	Scottsdale, AZ	Oct 02, 2017 - Oct 07, 2017	Community SANS
SANS October Singapore 2017	Singapore, Singapore	Oct 09, 2017 - Oct 28, 2017	Live Event
Community SANS Ottawa SEC503	Ottawa, ON	Oct 16, 2017 - Oct 21, 2017	Community SANS
SANS Berlin 2017	Berlin, Germany	Oct 23, 2017 - Oct 28, 2017	Live Event
San Diego Fall 2017 - SEC503: Intrusion Detection In-Depth	San Diego, CA	Oct 30, 2017 - Nov 04, 2017	vLive
SANS San Diego 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
SANS Paris November 2017	Paris, France	Nov 13, 2017 - Nov 18, 2017	Live Event
Community SANS Pensacola SEC503	Pensacola, FL	Nov 27, 2017 - Dec 02, 2017	Community SANS
SIEM & Tactical Analytics Summit & Training	Scottsdale, AZ	Nov 28, 2017 - Dec 05, 2017	Live Event
SANS Cyber Defense Initiative 2017	Washington, DC	Dec 12, 2017 - Dec 19, 2017	Live Event
SANS Security East 2018	New Orleans, LA	Jan 08, 2018 - Jan 13, 2018	Live Event
SANS Las Vegas 2018	Las Vegas, NV	Jan 28, 2018 - Feb 02, 2018	Live Event
SANS Dallas 2018	Dallas, TX	Feb 19, 2018 - Feb 24, 2018	Live Event
SANS OnDemand	Online	Anytime	Self Paced
SANS SelfStudy	Books & MP3s Only	Anytime	Self Paced