*** Northcutt, you just get the feeling sometimes that folks are applying what they learned in school. Solid use of a process, accuracy is fine, research into the source addresses. There are limits to what you can do with a firewall, but Walter seems to be dead on target for getting all you can get out of one, add some screenshots and this could be grown into a tutorial. 81 *

Walter Grech

GIAC Certified Intrusion Analyst Exam
Ten detects with analysis

Detect #1

```
13:42:22.006944 A.B.C.D.137 > A.B.C.255.137: udp 50 (ttl 128, id 13570)
13:42:22.756539 A.B.C.D.137 > A.B.C.255.137: udp 50 (ttl 128, id 13826)
13:42:23.506469 A.B.C.D.137 > A.B.C.255.137: udp 50 (ttl 128, id 14082)
13:44:52.038410 A.B.C.D.138 > A.B.C.255.138: udp 222 (ttl 128, id 14338)
13:47:22.055575 A.B.C.D.138 > A.B.C.255.138: udp 207 (ttl 128, id 14594)
13:47:22.055682 A.B.C.D.137 > A.B.C.255.137: udp 50 (ttl 128, id 14850)
13:47:22.804454 A.B.C.D.137 > A.B.C.255.137: udp 50 (ttl 128, id 15106)
13:47:23.554438 A.B.C.D.137 > A.B.C.255.137: udp 50 (ttl 128, id 15362)
13:49:52.081385 A.B.C.D.138 > A.B.C.255.138: udp 222 (ttl 128, id 15618)
13:52:22.102750 A.B.C.D.138 > A.B.C.255.138: udp 207 (ttl 128, id 15874)
13:52:22.102856 A.B.C.D.137 > A.B.C.255.137: udp 50 (ttl 128, id 16130)
13:52:22.852475 A.B.C.D.137 > A.B.C.255.137: udp 50 (ttl 128, id 16386)
```

Description:
This detect appeard on our internal network (behind firewall using tcpdump) and was logged in continuos cycles over an extended period of time. It was quickly identifyed as an anonolmous condition since the source address (A.B.C.D) was out of our internal IP range.

Active Targeting:
Yes

History/Background/Technique:
The trace is an automated process due to its regularity in time (the ID numbers were also in increments of 256 – i.e.not a busy machine). Since the capture times were only during the workday it was possible user interaction was directly or indirectly generating the traffic. The UDP broadcast originating from netbios-ns (255.137) and netbios-dgm (255.138) ports according to CERT and IBM CERT are some of the top exploited/mapping vulnerabilities. My concern was also increased by the unassigned IP address on our internal network scanning for services. Although the traffic was being dropped at the firewall this required immediate investigation.

Threat: High(Illegal address inside)

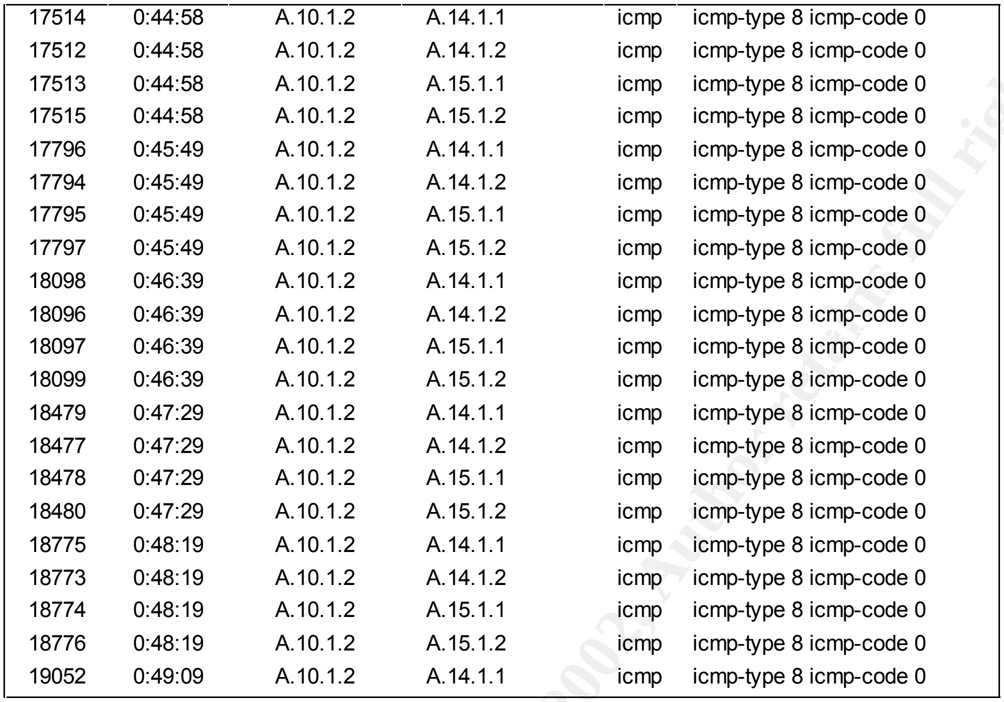
Severity: Moderate

Subsequent Action:
Since the IP address wasn't a valid source existing routing tables couldn't be used to trace back to the source station in a timely manner. Performing a tcpdump –e displayed the Ethernet header;

```
        14:07:23.621162 0:50:da:6a:7a:4d Broadcast ip 92: A.B.C.D.137 > 192.1.2.255.0
```

Working with the internal network team and examining the major internal routers MAC tables, the traffic was traced back to a contractor's workstation (non-company). The contractor had connected a workstation to the internal network (without permission) and was also using a dial-out modem and connecting to an external service. The illegal IP address matched what was bound to the contractor's NIC, it was operating only during the workday with the user dialing out and connecting to a resource database (confirming my earlier assumption). The trace wasn't a UDP map/exploit (false positive) but normal application traffic, the "out-of-range" IP address though led me to the source of the problem.

Detect #2



```
17514   0:44:58   A.10.1.2   A.14.1.1   icmp   icmp-type 8 icmp-code 0
17512   0:44:58   A.10.1.2   A.14.1.2   icmp   icmp-type 8 icmp-code 0
17513   0:44:58   A.10.1.2   A.15.1.1   icmp   icmp-type 8 icmp-code 0
17515   0:44:58   A.10.1.2   A.15.1.2   icmp   icmp-type 8 icmp-code 0
17796   0:45:49   A.10.1.2   A.14.1.1   icmp   icmp-type 8 icmp-code 0
17794   0:45:49   A.10.1.2   A.14.1.2   icmp   icmp-type 8 icmp-code 0
17795   0:45:49   A.10.1.2   A.15.1.1   icmp   icmp-type 8 icmp-code 0
17797   0:45:49   A.10.1.2   A.15.1.2   icmp   icmp-type 8 icmp-code 0
18098   0:46:39   A.10.1.2   A.14.1.1   icmp   icmp-type 8 icmp-code 0
18096   0:46:39   A.10.1.2   A.14.1.2   icmp   icmp-type 8 icmp-code 0
18097   0:46:39   A.10.1.2   A.15.1.1   icmp   icmp-type 8 icmp-code 0
18099   0:46:39   A.10.1.2   A.15.1.2   icmp   icmp-type 8 icmp-code 0
18479   0:47:29   A.10.1.2   A.14.1.1   icmp   icmp-type 8 icmp-code 0
18477   0:47:29   A.10.1.2   A.14.1.2   icmp   icmp-type 8 icmp-code 0
18478   0:47:29   A.10.1.2   A.15.1.1   icmp   icmp-type 8 icmp-code 0
18480   0:47:29   A.10.1.2   A.15.1.2   icmp   icmp-type 8 icmp-code 0
18775   0:48:19   A.10.1.2   A.14.1.1   icmp   icmp-type 8 icmp-code 0
18773   0:48:19   A.10.1.2   A.14.1.2   icmp   icmp-type 8 icmp-code 0
18774   0:48:19   A.10.1.2   A.15.1.1   icmp   icmp-type 8 icmp-code 0
18776   0:48:19   A.10.1.2   A.15.1.2   icmp   icmp-type 8 icmp-code 0
19052   0:49:09   A.10.1.2   A.14.1.1   icmp   icmp-type 8 icmp-code 0
```

Description:
This detect was captured by Checkpoint firewall originating from inside the network. All traffic was dropped due to the source address being out of range to internal addressing. (Also ICMP is dropped at the firewall by default).

Active Targeting:
Yes

History/Background/Technique:
The ICMP trace originates from a single low (.2) IP address to two low (.1 and .2) IP addresses within the same class A address range (A) Typically, low addresses are assigned by network administrators to networking hardware (i.e. routers, hubs, switches..) when networks are first constructed, these are good locations for hackers to investigate/attack. The traffic is running in fairly regular patterns (groups of 4 per second, indicating an automated process) it's also possible the source was spoofed and focusing on the two destination for a "denial of service" attempt.

Threat: High(Could be a compromised machine on the internal net)

Severity: Moderate

Subsequent Action:
Backtracking sensor traces through the main routers to locate to source. Checking for possible Trojans, misconfigured device and/or misrouted external traffic. Check log history for additional similar traffic patterns (although not searching for the same source IP). Performing an nslookup on the source confirms that source IP is an externally registered address.

Detect #3

```
  732  0:00:47    40206  A.B.C.D    E.F.G.H    tcp   71  ftp-data   len 41
 1104  0:01:47    40206  A.B.C.D    E.F.G.H    tcp   71  ftp-data   len 41
 1535  0:02:47    40206  A.B.C.D    E.F.G.H    tcp   71  ftp-data   len 41
 2004  0:03:47    40206  A.B.C.D    E.F.G.H    tcp   71  ftp-data   len 41
 2414  0:04:47    40206  A.B.C.D    E.F.G.H    tcp   71  ftp-data   len 41
 2899  0:05:47    40206  A.B.C.D    E.F.G.H    tcp   71  ftp-data   len 41
 3335  0:06:47    40206  A.B.C.D    E.F.G.H    tcp   71  ftp-data   len 41
 3752  0:07:47    40206  A.B.C.D    E.F.G.H    tcp   71  ftp-data   len 41
 4176  0:08:47    40206  A.B.C.D    E.F.G.H    tcp   71  ftp-data   len 41
 4668  0:09:47    40206  A.B.C.D    E.F.G.H    tcp   71  ftp-data   len 41
 5203  0:10:47    40206  A.B.C.D    E.F.G.H    tcp   71  ftp-data   len 41
 5691  0:11:47    40206  A.B.C.D    E.F.G.H    tcp   71  ftp-data   len 41
 6149  0:12:47    40206  A.B.C.D    E.F.G.H    tcp   71  ftp-data   len 41
 6629  0:13:47    40206  A.B.C.D    E.F.G.H    tcp   71  ftp-data   len 41
 7109  0:14:47    40206  A.B.C.D    E.F.G.H    tcp   71  ftp-data   len 41
 7577  0:15:47    40206  A.B.C.D    E.F.G.H    tcp   71  ftp-data   len 41
 8043  0:16:47    40206  A.B.C.D    E.F.G.H    tcp   71  ftp-data   len 41
16525  0:33:47    40206  A.B.C.D    E.F.G.H    tcp   71  ftp-data   len 41
17038  0:34:47    40206  A.B.C.D    E.F.G.H    tcp   71  ftp-data   len 41
17519  0:35:47    40206  A.B.C.D    E.F.G.H    tcp   71  ftp-data   len 41
17990  0:36:47    40206  A.B.C.D    E.F.G.H    tcp   71  ftp-data   len 41
18497  0:37:47    40206  A.B.C.D    E.F.G.H    tcp   71  ftp-data   len 41
```

Description:

Collected this trace by scanning through the daily logs between our Firewall (E.F.G.H) and an external Internet address (A.B.C.D).

Active Targeting:

Yes

History/Background/Technique:

A quick scan of the entire firewall log file might have overlooked this traffic as normal FTP data port activity mixed in with all the other traces. But the off-hour evenly spaced/sized traffic (exactly 1 min/41 len) looked anonolmous and I chose to investigate further. It resembled a "low and somewhat slow" automated scan pattern that might be involved in a FTP hijack.

Threat: Low(All traffic dropped by Firewall, rule 71 in trace)

Severity: Medium

Subsequent Action:

Performing a nslookup on the source address I found it was assigned to a well known large university (due to the diverse range of needs at colleges and universities, these system are not well known for maintaining high security levels). I then filtered the Firewall logs for any FTP session activity during the time range of this activity with no matching results. I added the source address to my list of sites for tcpdump and Firewall "full" logging tracking. No further attempts have been recorded. Possible explinations are misuse of system by students but more likely a spoofed source address by a non-university user.

Detect #4

```
524762   17:34:22 hme0   drop   chargen   External.IP   Internal.IP   udp    len 112
525038   17:35:04 hme0   drop             External.IP   Internal.IP   icmp   icmp-type 3 icmp-code 3
```

Description:
This snippet appeared in the daily firewall logs during my periodic scans and caught my attention as an anonolmous condition.

Active Targeting:
Yes

History/Background/Methods:
These two packets by themselves don't pose a large threat to "Internal IP" address as they were dropped by the Firewall. If allowed to pass through, a chargen misuse can effectively disable a UNIX server by causing it to spend all its time processing packets that it is sending to itself, or be used in an attack on another machine. What makes this an anonolmous condition is the lack of any additional traffic to/from the "External IP". The time between the two packets is less than a minute which could reflect a manual exploratory scan in an attempt to elicit a response (chargen=character generator or icmp=ping). Examining the normal "stimulus and response" defined in RFC768,791,792 and 793 a hacker can learn a good deal of information.

Threat: Low

Severity: Minimal

Subsequent Action:
Running a filter for the previous and subsequent days log files didn't produce any additional traffic to/from the "External IP". Checked the "Internal IP" server for proper operation, if a chargen attack had been initiated a Kill and restart the inetd daemon would have been required. To remove this vulnerability, editing the /etc/inetd.conf file and disabling the chargen service for inetd would be necessary. In our environment, this service is usually no longer necessary, but sometimes active on internal UNIX hosts.

Detect #5

```
11-Apr-00   18:32:35 qfe0      33846   External.IP   D.M.Z.32      domain-udp
11-Apr-00   18:32:41 qfe0      33849   External.IP   D.M.Z.33      domain-udp
11-Apr-00   18:32:45 qfe0      33850   External.IP   D.M.Z.34      domain-udp
11-Apr-00   18:32:50 qfe0      33853   External.IP   D.M.Z.35      domain-udp
11-Apr-00   18:32:51 qfe0      33855   External.IP   D.M.Z.36      domain-udp
11-Apr-00   18:32:55 qfe0      33856   External.IP   D.M.Z.37      domain-udp
11-Apr-00   18:33:00 qfe0      33858   External.IP   D.M.Z.38      domain-udp
11-Apr-00   18:33:00 qfe0      33860   External.IP   D.M.Z.39      domain-udp
```

Description:

This trace was targeting the DMZ side of a Firewall, the External.IP is scanning an address range for DNS services

Active Targeting:

Yes

History/Background/Methods:

This automated trace (timing is close) originating from the External.IP (not a busy machine, ports are almost sequential) is scanning our DMZ for a DNS server to respond. A Unix server doesn't necessarily have to have the prime role of a DNS server but just have the named deamon running with access allowed to port 53 (domain-udp). Zone transfers will allow a hacker to download specific host information about your systems. DNS servers are one of the primary "investigated" services by hackers.

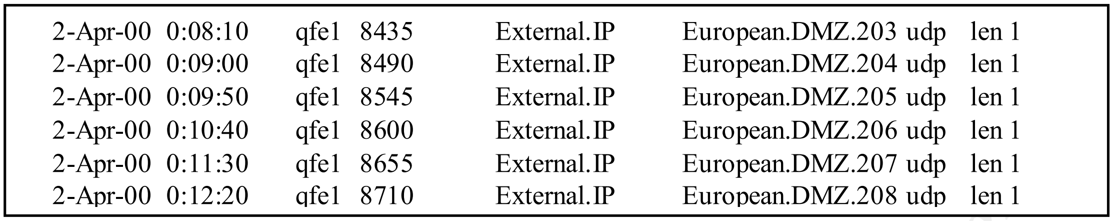
Threat: Medium

Severity: Moderate

Subsequent Action:

Scanned the DMZ logs for past traffic from External.IP, none located. Checked existing DNS server for proper operation (All OK). Verify all other machines on the DMZ are NOT running named/DNS services. Verified the Firewall rulebase to allow domain udp/tcp traffic only to the public DNS server, drop all others (and log).

Detect #6



```
2-Apr-00  0:08:10     qfe1   8435         External.IP          European.DMZ.203 udp    len 1
2-Apr-00  0:09:00     qfe1   8490         External.IP          European.DMZ.204 udp    len 1
2-Apr-00  0:09:50     qfe1   8545         External.IP          European.DMZ.205 udp    len 1
2-Apr-00  0:10:40     qfe1   8600         External.IP          European.DMZ.206 udp    len 1
2-Apr-00  0:11:30     qfe1   8655         External.IP          European.DMZ.207 udp    len 1
2-Apr-00  0:12:20     qfe1   8710         External.IP          European.DMZ.208 udp    len 1
```

Description:
This Firewall trace started late in the evening targeted at a host on our European DMZ.
Active Targeting:
Yes
History/Background/Methods:
This UDP port scan is attempting to scan a single host for available UDP services. The obvious facts are the sequential UDP destination ports and automated timing/source port intervals. A few additional possibilities may be based upon the late evening and the fixed intervals for time and source ports. This could be an "off hour" automated port scan of a few sites blended together causing the fixed intervals and delays, hoping to bypass by intrusion detection devices.
Threat: **Low(Firewall dropped almost all)**
Severity: **Moderate**
Subsequent Action:
Source IP block registered to Asian ISP. Checked few days logs before and after and found no additional traffic from External.IP. However a few "scan type" patterns are emerging. Alerted the European sysadmins and advised to go to a heightened alert. This pattern fits a information gathering probe before an attack. We are continuing to analyze the existing logs, notify the effected ISP and start a tcpdump for anonolmous conditions (See Detect #7).

Detect #7

```
08:48:54.920365 External.IP.36623 > European.DMZ1.184: udp 0 (DF)
08:48:54.921963 External.IP.36623 > European.DMZ1.1513: udp 0 (DF)
08:48:54.922503 External.IP.36623 > European.DMZ1.940: udp 0 (DF)
08:48:54.923062 External.IP.36623 > European.DMZ1.1463: udp 0 (DF)
08:48:54.923880 External.IP.36623 > European.DMZ1.739: udp 0 (DF)
08:48:54.925070 External.IP.36623 > European.DMZ1.573: udp 0 (DF)
08:48:54.925869 External.IP.36623 > European.DMZ1.958: udp 0 (DF)
08:48:54.926855 External.IP.36623 > European.DMZ1.9876: udp 0 (DF)
08:48:54.927488 External.IP.36623 > European.DMZ1.446: udp 0 (DF)
08:48:54.928107 External.IP.36623 > European.DMZ1.425: udp 0 (DF)
08:48:54.929293 External.IP.36623 > European.DMZ1.575: udp 0 (DF)
08:48:54.929946 External.IP.36623 > European.DMZ1.1520: udp 0 (DF)
08:48:54.931137 External.IP.36623 > European.DMZ1.2048: udp 0 (DF)
08:48:54.932236 External.IP.36623 > European.DMZ1.5305: udp 0 (DF)
08:48:54.932870 External.IP.36623 > European.DMZ1.335: udp 0 (DF)
08:48:54.933484 External.IP.36623 > European.DMZ1.330: udp 0 (DF)
08:48:54.934284 External.IP.36623 > European.DMZ1.189: udp 0 (DF)
08:48:54.935289 External.IP.36623 > European.DMZ1.31337: udp 0 (DF)
08:48:54.936109 External.IP.36623 > European.DMZ1.839: udp 0 (DF)
08:48:54.936944 External.IP.36623 > European.DMZ1.443: udp 0 (DF)
08:48:54.937894 External.IP.36623 > European.DMZ1.768: udp 0 (DF)
08:48:54.938694 External.IP.36623 > European.DMZ1.1518: udp 0 (DF)
08:48:54.939548 External.IP.36623 > European.DMZ1.404: udp 0 (DF)
```

Description:
Another UPD port scan (See Detect #6) targeted at a (different) host on our European DMZ
Active Targeting:
Yes
History/Background/Methods:
This UDP port scan is similar to Detect #6 in the sense that it's scanning for available UDP services on a single machine in our European DMZ. This sequence is happening very quickly, during working hours, coming from a fixed source port and targeting random UDP ports. This is scanning software performing a fast search.

**** Rated both Threat and Severity High due to the current heightened state of alert ****

Threat: **High**
Severity: **High**
Subsequent Action:
Found little evidence to tie Detect #6 & 7 together as coming from the same source. Originating IP address resolves to an American ISP cable provider, checking with them for more information , not ruling out the possibility of spoofing. Verified Firewall rules for proper filters and checked European.DMZ1 for proper functionality (All checked out OK). Needless to say we are keeping a sharp eye on this site.

Detect #8

```
09:32:34.770770 External.IP.44622 > A.B.C.D.chargen: S 4170688054:41 70688054(0) win 8760 <mss 1460> (DF)
09:32:34.773348 External.IP.44623 > A.B.C.D.902: S 4170699309:417069 9309(0) win 8760 <mss 1460> (DF)
09:32:34.775346 External.IP.44624 > A.B.C.D.291: S 4170711315:417071 1315(0) win 8760 <mss 1460> (DF)
09:32:34.777487 External.IP.44625 > A.B.C.D.761: S 4170763371:417076 3371(0) win 8760 <mss 1460> (DF)
09:32:34.790514 External.IP.44626 > A.B.C.D.168: S 4170870632:417087 0632(0) win 8760 <mss 1460> (DF)
09:32:34.791666 External.IP.44627 > A.B.C.D.1664: S 4170995572:41709 95572(0) win 8760 <mss 1460> (DF)
09:32:34.794986 External.IP.44628 > A.B.C.D.838: S 4171112898:417111 2898(0) win 8760 <mss 1460> (DF)
09:32:34.801626 External.IP.44629 > A.B.C.D.1011: S 4171119215:41711 19215(0) win 8760 <mss 1460> (DF)
```

Description:
This is a trace of an External.IP sending traffic to one of our public web servers.

Active Targeting:
Yes

History/Background/Methods:
We have seen problem traffic from this address block before and are working with the ISP to reduce and identify it. This is a very fast scan of a single machine originating from sequential source ports. It's targeting random destination ports, probably with a slim hope of avoiding detection. This is a typical example of an NMAP scan. The originator is examining a know public server for available services.

Threat: Low

Severity: Minimal

Subsequent Action:
Since we we're blocking/logging all traffic from this address it was reported to the ISP immediately. The Firewall logs were scanned to assure no traffic was allowed to pass and the public server was reviewed for only essential service ports in operation.

Detect #9

```
10:53:44.631000 A.B.C.D.34795 > E.F.G.H.622: S 842103828:8421038 28(0) win 1024 (DF)
10:53:44.631024 E.F.G.H.622 > A.B.C.D.34795: R 0:0(0) ack 842103 829 win 0 (DF)
10:53:44.632263 A.B.C.D.34795 > E.F.G.H.873: S 842103828:8421038 28(0) win 1024 (DF)
10:53:44.632287 E.F.G.H.873 > A.B.C.D.34795: R 0:0(0) ack 842103 829 win 0 (DF)
10:53:44.633342 A.B.C.D.34795 > E.F.G.H.284: S 842103828:8421038 28(0) win 1024 (DF)
10:53:44.633366 E.F.G.H.284 > A.B.C.D.34795: R 0:0(0) ack 842103 829 win 0 (DF)
10:53:44.634272 A.B.C.D.34795 > E.F.G.H.742: S 842103828:8421038 28(0) win 1024 (DF)
10:53:44.634296 E.F.G.H.742 > A.B.C.D.34795: R 0:0(0) ack 842103 829 win 0 (DF)
10:53:44.635162 A.B.C.D.34795 > E.F.G.H.1407: S 842103828:842103 828(0) win 1024 (DF)
10:53:44.635187 E.F.G.H.1407 > A.B.C.D.34795: R 0:0(0) ack 84210 3829 win 0 (DF)
10:53:44.639229 A.B.C.D.34795 > E.F.G.H.65301: S 842103828:84210 3828(0) win 1024 (DF)
10:53:44.639265 E.F.G.H.65301 > A.B.C.D.34795: R 0:0(0) ack 8421 03829 win 0 (DF)
```

Description:

This trace was taken after a sysadmin reported slow traffic/high usage on a normally quite server.

Active Targeting:

Yes

History/Background/Methods:

Discussing the problem with the sysadmin I found that the server was not normally a high usage machine (E.F.G.H) and no other systems on the (DMZ) network were reporting problems. After re-booting the server and producing the same results, he contacted me. I immediately ran tcpdump and quickly spotted the detect above. The pattern was very quick and regular, A.B.C.D was sending a SYN request to different E.F.G.H ports and E.F.G.H was dutifully replying with either a RST or SYN/ACK. The trace was continuous with the target server reserving memory for each SYN/ACK response. This could constitute as a denial of service attack as the server was holding memory for non-existant connection requests and quickly running out of resources. Or the source is scanning for available service ports on the target machine

Threat: **Medium**

Severity: **High**

Subsequent Action:

The traffic was immediately blocked by adding a new rule to the Firewall. I double-checked the Firewall logs after the creation of the new rule to confirm that the traffic was stopped or rerouted from a new source IP. Contacted the owner of the IP address block and informed them of the offending traffic and emailed them a copy of the trace. The sysadmin was directed to perform a total security service audit of the machine and reassessment of available port services.

Detect #10

```
15:41:51.113057 A.B.C.D.46144 > E.F.G.H.telnet: S 1163185400:1163185400(0) win 8760 <mss 1460> (DF)
15:41:51.113085 E.F.G.H.telnet > A.B.C.D.46144: S 3220663854:3220663854(0) ack 1163185401 win 8760 <mss 1460> (DF)
15:41:51.115098 A.B.C.D.46144 > E.F.G.H.telnet: . ack 1 win 8760 (DF)
……..
15:41:51.120767 A.B.C.D.46144 > E.F.G.H.telnet: P 1:25(24) ack 1 win 8760 (DF)
15:41:51.121844 E.F.G.H.telnet > A.B.C.D.46144: . ack 25 win 8760 (DF)
15:41:51.169685 E.F.G.H.telnet > A.B.C.D.46144: P 1:16(15) ack 25 win 8760 (DF)
15:41:51.171498 A.B.C.D.46144 > E.F.G.H.telnet: . ack 16 win 8760 (DF)
15:41:51.171523 E.F.G.H.telnet > A.B.C.D.46144: P 16:31(15) ack 25 win 8760 (DF)
15:41:51.173285 A.B.C.D.46144 > E.F.G.H.telnet: P 25:31(6) ack 16 win 8760 (DF)
15:41:51.214571 A.B.C.D.46144 > E.F.G.H.telnet: . ack 31 win 8760 (DF)
15:41:51.214595 E.F.G.H.telnet > A.B.C.D.46144: P 31:49(18) ack 31 win 8760 (DF)
15:41:51.223130 A.B.C.D.46144 > E.F.G.H.telnet: P 31:47(16) ack 49 win 8760 (DF)
15:41:51.224021 E.F.G.H.telnet > A.B.C.D.46144: P 49:70(21) ack 47 win 8760 (DF)
……..
15:42:02.084835 E.F.G.H.telnet > A.B.C.D.46144: FP 496:504(8) ack 82 win 8760 (DF)
15:42:02.086650 A.B.C.D.46144 > E.F.G.H.telnet: . ack 505 win 8760 (DF)
15:42:02.090596 A.B.C.D.46144 > E.F.G.H.telnet: F 82:82(0) ack 505 win 8760 (DF)
15:42:02.090624 E.F.G.H.telnet > A.B.C.D.46144: . ack 83 win 8760 (DF)
```

Description:

Since the first nine detects analyzed anonolmous conditions dealing with (possible) malicious intent, I felt it important to include an analysis for a specific type of trace describing normal "expected behavior" from a telnet session and then break it down into three parts. Since a large portion of an Intrusion Analysis job is scanning over code with different tools, it's a developed learning/experience factor that will assist in quiclky separating the "wheat from the chaff".

Active Targeting:

Yes

History/Background/Methods:

First paragraph: To initiate a telnet session a "three-way handshake" must occur. Client D begins by setting the SYN (synchronize) flag in a packet originating from port 46144 destined to server H on port 23 (telnet). This is called an "active open". *The normal expected behavior from the client would be to send very few "active open" requests until a reply is heard from the server, if the client sends a continuos stream of SYN requests, ignoring all replies from the server, it could be considered as a SYN attack.* Also contained in the first packet is the initial TCP sequence number for communication in the client to server direction. *Calculating the sequence numbers is a hacking method of breaking into a machine.* Next, server H sends a SYN/ACK packet from port 23 to client D port 46144 to confirm receipt of the initial sequence number, this called a "passive open". This packet has an initial sequence number for the server to client communication as well as an acknowledgement for the client's initial sequence number. *Unneeded services should be disabled to prevent scanning of ports to discover existing access points for intrusion attempts.* Finally, client D replies with an ACK to server H that it has received the server's initial sequence number and the session is now ready to proceed.

Second Paragraph: The telnet session packets are now pushed (P) between client and server through the two ports reserved (46144 and 23) and confirmation sent back by means of an acknowledgement (ACK) packet. Again, sequence numbers are used to keep the packets in the proper order at the final destination. *After displaying the initial sequence numbers, tcpdump has made reading easier by incrementing by starting at the number 1.*

Third Paragraph: Here we see the server pushing an end of connection request FIN (F) packet to the client, still utilizing the sequence numbers for proper order. *The user has just issued a "logoff" command.* The client then acknowledges this packet (ACK). At this point no more data will be flowing in that direction, a "half close" has taken place. Since this connection is a full duplex connection (*data flows in both directions)*, both channels must be closed to consider this a proper disconnect. Now the client sends an end of connection request FIN packet to the server and the server responds with an acknowledgment (ACK). *It takes four connections to fully terminate a TCP connection.*

Threat: Low
Severity: Minimal
Subsequent Action:
None taken, session performed with expected normal results