Global Information Assurance Certification Paper

*** Northcutt, some good stuff here, check out three, but I found it really hard to read the formatting and thus the points deduction. Good use of an analysis process. Variety of sources, home field advantage, and it shows. With 10 you need to show how the attacker is mapping with spoofed addresses. 72 ***

name: Charles Mitchell

1111111111111111111111111111111111111111111111111111111111111
111111111111111

Log format:
date/time: machine/facility: rule:
action:interface:protocol:
src ip/port: dst ip/port: length:TOS: IP ID: IP
Flags: TTL: rule#

Mar 4 16:45:46 fw kernel: Packet log: input REJECT eth3
PROTO=17
192.168.1.51:9999 10.0.0.114:53 L=70 S=0x00 I=44736
F=0x0000 T=25 (#100)
Mar 4 16:46:08 fw kernel: Packet log: input REJECT eth3
PROTO=17
192.168.1.51:9999 10.0.0.114:53 L=70 S=0x00 I=44737
F=0x0000 T=25 (#100)
Mar 4 16:48:26 fw kernel: Packet log: input REJECT eth3
PROTO=17
192.168.1.51:9999 10.0.0.114:53 L=70 S=0x00 I=44544
F=0x0000 T=25 (#100)
Mar 4 16:55:40 fw kernel: Packet log: input REJECT eth3
PROTO=17
192.168.1.51:9999 10.0.0.115:53 L=70 S=0x00 I=44992
F=0x0000 T=25 (#100)
Mar 4 16:55:42 fw kernel: Packet log: input REJECT eth3
PROTO=17
192.168.1.51:9999 10.0.0.116:53 L=70 S=0x00 I=45057
F=0x0000 T=25 (#100)
Mar 4 16:55:52 fw kernel: Packet log: input REJECT eth3
PROTO=17
192.168.1.51:9999 10.0.0.115:53 L=70 S=0x00 I=44800
F=0x0000 T=25 (#100)
Mar 4 16:56:11 fw kernel: Packet log: input REJECT eth3
PROTO=17
192.168.1.51:9999 10.0.0.114:53 L=70 S=0x00 I=44544
F=0x0000 T=25 (#100)

```
Mar  4 16:58:40 fw kernel: Packet log: input REJECT eth3
PROTO=17
192.168.1.51:9999 10.0.0.115:53 L=70 S=0x00 I=44800
F=0x0000 T=25 (#100)
Mar  4 17:08:03 fw kernel: Packet log: input REJECT eth3
PROTO=17
192.168.1.51:9999 10.0.0.116:53 L=70 S=0x00 I=45057
F=0x0000 T=25 (#100)
```

This sequence repeats after a ~7:30 pause for one hour 30 minutes.

Notables:
* The IP ID used the same numbers in each repetition of the sequence.
* The packet size was always the same. Payload unknown.
* Each live machine on our network received packets during each repetition.
* The source port was the same in every repetition.
* Approx. 8 packets in each repetition.

Summary: This detect comes from the packet filter logs on our firewall.
Its pattern stands out clearly from simply grepping the logs. An
nslookup on the source IP gives machine51.somewhere.tld. which
seems to indicate a lab type machine probably with little supervision.
A nice place to do network mapping from and ship the results 'home'.

Technique: Poor. Perhaps they are just learning to use their new tool?
One scan would have been interresting, but repeating it for an hour
and a half is foolish.

Intent: Information gathering. They are trying to locate (and possibly
determine version) DNS servers at each IP.

Severity: Low. All the packets were dropped by the firewall. DNS is not
accessible from the outside.

```
Conclusion:  Network mapping that has no consequence on our
network.  No
     action needed.

1111111111111111111111111111111111111111111111111111111111111
111111111111111
2222222222222222222222222222222222222222222222222222222222222
22222222222222222

Log format:
date/time:         machine/facility:     rule:
action:interface:protocol:
src ip/port:      dst ip/port:  length:TOS:  IP ID: IP
Flags: TTL: rule#

Apr  1 20:51:25 fw kernel: Packet log: input REJECT eth3
PROTO=6
192.168.1.93:51 10.0.0.114:110 L=40 S=0x00 I=53798 F=0x0000
T=53 SYN (#73)
Apr  1 20:51:25 fw kernel: Packet log: input REJECT eth3
PROTO=6
192.168.1.93:51 10.0.0.114:2000 L=40 S=0x00 I=3285 F=0x0000
T=53 SYN (#87)
Apr  1 20:51:27 fw kernel: Packet log: input REJECT eth3
PROTO=6
192.168.1.93:51 10.0.0.114:2049 L=40 S=0x00 I=49912
F=0x0000 T=53 SYN (#89)
Apr  1 20:51:35 fw kernel: Packet log: input REJECT eth3
PROTO=6
192.168.1.93:51 10.0.0.114:111 L=40 S=0x00 I=21697 F=0x0000
T=53 SYN (#85)
Apr  1 20:51:44 fw kernel: Packet log: input REJECT eth3
PROTO=6
192.168.1.93:51 10.0.0.114:37 L=40 S=0x00 I=23787 F=0x0000
T=53 SYN (#67)
Apr  1 20:51:54 fw kernel: Packet log: input REJECT eth3
PROTO=6
192.168.1.93:51 10.0.0.114:525 L=40 S=0x00 I=35691 F=0x0000
T=53 SYN (#82)
Apr  1 20:51:56 fw kernel: Packet log: input REJECT eth3
PROTO=6
192.168.1.93:51 10.0.0.114:515 L=40 S=0x00 I=54522 F=0x0000
T=53 SYN (#80)

Notables:
* Source port is the same for each packet.
* All the destination ports are 'interresting'.
```

* The size is the same for each packet.
* all packets arrived in quick succession.
* tcpd and other services didn't see anything indication a SYN stealth scan.
* Test runs of nmap show 40 as the default packet size for a SYN stealth scan.

Summary: This detect is from the log files on our firewall. Testing with
nmap shows identical results for the SYN scan when lack of tcpd logging
is taken into account. Nslookup for the source gives a numbered machine
at a large local university.

Technique: Fair. They thought far enough to use a 'stealth' SYN scan.

Intent: Host service mapping.

Severity: Low. Currently no important services live on machines accessible
to the public. Services that are accessible are completely isolated
from our internal network.

Conclusion: Higher education at work. No further action needed.

22222222222222222222222222222222222222222222222222222222222222222222
33333333333333333333333333333333333333333333333333333333333333333333

Log format:
date/time: machine/facility: rule: action:interface:protocol:
src ip/port: dst ip/port: length:TOS: IP ID: IP Flags: TTL: rule#

Mar 28 21:00:38 fw kernel: Packet log: input REJECT eth3 PROTO=17
192.168.1.80:3 10.0.0.114:111 L=28 S=0x00 I=13888 F=0x0000 T=37 (#86)
Mar 28 21:00:38 fw kernel: Packet log: input REJECT eth3 PROTO=17

```
192.168.1.80:3 10.0.0.115:111 L=28 S=0x00 I=40419 F=0x0000
T=37 (#86)
Mar  28 21:00:39 fw kernel: Packet log: input REJECT eth3
PROTO=17
192.168.1.80:3 10.0.0.116:111 L=28 S=0x00 I=9586 F=0x0000
T=37 (#86)

Notables:
* Each IP is touched exactly once.
* Source port remains constant.
* Timing is very quick.

Summary:  This detect comes from the logs on our firewall.
This scan
     touched every exposed IP on our network exactly once.
Nslookup
     on the source IP shows that this system is a numbered
system from
     the residence halls of a local university.

Technique:  Fair.  This one keeps a low profile by keeping
traffic
     volume low.  The scan is very focussed.


Intent:  Network mapping for a specific service, RPC.

Severity:  Medium.  This attacker has a very specific
target service.
     Any hits are almost certain to be followed by an
exploit.

Conclusion:  There are no machines on our network that are
accessible
     by the public that run any kind of RPC.  The scan does
not effect
     our network,  no further action needed.

33333333333333333333333333333333333333333333333333333333
333333
44444444444444444444444444444444444444444444444444444444
44444444444

Log format:
date/time:         machine/facility: UID : EUID : process
name : PID
```

Dec 21 04:40:40 merlin kernel: Security: return onto stack
running
as UID 0, EUID 0, process named:67

Summary: This is syslog output from Solar Designer's non-
executable
     stack patch to prevent some types of buffer busting.
The source
     is my main home network server which has some ports
visible to
     the public through the firewall. The attack causes
the named
     process to dump core and die, but a watchdog on this
system
     brings it back at most 60 seconds later. Unfortunately
no data
     was collected on the source of the attack.

Technique: Fair. When it didn't work the attacker should
have gone
     away. Instead they tried once more after the watchdog
brought
     named back up.

Intent: To take control of my system.

Severity: High. The attacker knows (at least some of) the
services
     that are running on my network. He also has exploits
prepared
     to attack any weeknesses found. Up to this point he
remained
     undetected by not scanning all ports on my firewall.
Apparently
     only ports open to the public were scanned.

Conclusion: This was very nearly a successful attack.
Thank you Solar.
     Further action was to upgrade to that latest BIND
distribution
     and also to the latest version of all the other
publicly accessible
     services.

4444444444444444444444444444444444444444444444444444444444
44444444444

```
5555555555555555555555555555555555555555555555555555555555555555
55555555

Log format:
source IP:          date/time:
command:args:       result:

192.168.1.28 - - [07/Feb/1999:23:04:22 -0500] "GET /cgi-
bin/phf" 404 -
192.168.1.28 - - [07/Feb/1999:23:04:23 -0500] "GET /cgi-
bin/test-cgi" 200 407
192.168.1.28 - - [07/Feb/1999:23:04:23 -0500] "GET /cgi-
bin/handler" 404 -

Log format:
date/time:                  status:   source IP:
message:

[Sun Feb  7 23:04:22 1999] [error] [client 192.168.1.28]
script not found
or unable to stat: /var/lib/apache/share/cgi-bin/phf
[Sun Feb  7 23:04:23 1999] [error] [client 192.168.1.28]
script not found
or unable to stat: /var/lib/apache/share/cgi-bin/handler

Notables:
* phf and handler failed.
* test-cgi succeeded as shown in the second log.

Summary:  This detect if from the access logs (first block)
and error logs
     (second block) of my home network.  The attack was
detected by
     activity lights on the modem while I was reading a
book.  I was
     curious about who was checking me out, so I looked at
the logs.
     I was, by luck, able to respond to the attack in near
real time.  The
     attacker did gather some information about my system.
The source
     of this attack resolved to a dynamic IP from an ISP
dialup bank.

Technique:  Fair.  This attack would have gone un-noticed
for perhaps weeks
```

had I not been curious. That would have been plenty of time to run
crack had phf worked.

Intent: Host information gathering and ultimately exploit.

Severity: Medium. I was running up to date enough to have phf turned off.
They did, however, successfully gather information from test-cgi.

Conclusion: This attack was partially successful. Further action was that
I spent the next couple hours learning about apache configuration
so that I could turn off all access to cgi-bin and enable access only
for specific and known scripts.

5555555555555555555555555555555555555555555555555555555555555555
6666666666666666666666666666666666666666666666666666666666666666

Log format:
date/time: machine/facility: rule: action:interface:protocol:
src ip/port: dst ip/port: length:TOS: IP ID: IP Flags: TTL: rule#

Apr 4 10:51:34 fw kernel: Packet log: input REJECT eth3 PROTO=6
10.2.7.240:80 10.0.0.114:1863 L=52 S=0x00 I=4939 F=0x4000 T=49 (#11)
Apr 4 10:51:35 fw kernel: Packet log: input REJECT eth3 PROTO=6
10.2.7.240:80 10.0.0.114:1863 L=52 S=0x00 I=5341 F=0x4000 T=49 (#11)
Apr 4 10:51:37 fw kernel: Packet log: input REJECT eth3 PROTO=6
10.2.7.240:80 10.0.0.114:1863 L=52 S=0x00 I=5756 F=0x4000 T=49 (#11)
Apr 4 10:51:41 fw kernel: Packet log: input REJECT eth3 PROTO=6
10.2.7.240:80 10.0.0.114:1863 L=52 S=0x00 I=6710 F=0x4000 T=49 (#11)

```
Apr  4 10:51:49 fw kernel: Packet log: input REJECT eth3
PROTO=6
10.2.7.240:80 10.0.0.114:1863 L=52 S=0x00 I=9439 F=0x4000
T=49 (#11)
Apr  4 10:52:06 fw kernel: Packet log: input REJECT eth3
PROTO=6
10.2.7.240:80 10.0.0.114:1863 L=52 S=0x00 I=15009 F=0x4000
T=49 (#11)
Apr  4 10:52:38 fw kernel: Packet log: input REJECT eth3
PROTO=6
10.2.7.240:80 10.0.0.114:1863 L=52 S=0x00 I=18496 F=0x4000
T=49 (#11)
Apr  4 10:53:44 fw kernel: Packet log: input REJECT eth3
PROTO=6
10.2.7.240:80 10.0.0.114:1863 L=52 S=0x00 I=33762 F=0x4000
T=49 (#11)
Apr  4 10:55:44 fw kernel: Packet log: input REJECT eth3
PROTO=6
10.2.7.240:80 10.0.0.114:1863 L=52 S=0x00 I=61459 F=0x4000
T=49 (#11)
Apr  4 10:57:44 fw kernel: Packet log: input REJECT eth3
PROTO=6
10.2.7.240:80 10.0.0.114:1863 L=52 S=0x00 I=20983 F=0x4000
T=49 (#11)

Apr  4 10:50:00 fw kernel: Packet log: input REJECT eth3
PROTO=6
10.2.7.244:80 10.0.0.114:1834 L=52 S=0x00 I=49491 F=0x4000
T=49 (#11)
Apr  4 10:50:02 fw kernel: Packet log: input REJECT eth3
PROTO=6
10.2.7.244:80 10.0.0.114:1834 L=52 S=0x00 I=49863 F=0x4000
T=49 (#11)
Apr  4 10:50:07 fw kernel: Packet log: input REJECT eth3
PROTO=6
10.2.7.244:80 10.0.0.114:1834 L=52 S=0x00 I=50288 F=0x4000
T=49 (#11)
Apr  4 10:50:16 fw kernel: Packet log: input REJECT eth3
PROTO=6
10.2.7.244:80 10.0.0.114:1834 L=52 S=0x00 I=52618 F=0x4000
T=49 (#11)

Notables:
* The time between packets starts at 1 second and doubles
between each
     pair up to 120 seconds where it stays for about 20
minutes.
```

* The source address is spoofed.
* There were two sequences of this going on in parallel from very similar
    spoofed IPs.
* The TTLs were the same between the two 'sessions'.
* The detect repeats from one of the IPs four hours later.
* Each detect was to a different unknown TCP port.
* The timing doesn't match other 'load balancing' detects.
* The ports involved don't match other 'load balancing' detects.

Summary: This detect is from our firewall logs. Only the IP of the firewall
    was touched. Only ports 1863, 1834, and 1050 are probed. The timing
    indicates a desire to evade port scan detection tools. The spoofed IP
    indicates a desire to camaflague the real activity with chaff. The timing
    and destination ports don't match 'load balancing' detects.

Technique: Fair. The slow timing of the packets is useful, but the spoofed
    source addresses make the detect much easier.

Intent: Informatin gathering or response measurement.

Severity: Medium. Nothing else of use was detected. Perhaps they carried
    out a successful 'stealth' scan of a single or very few real target
    ports.

Conclusion: This attack may very well have succeeded. The target information
    they were seeking is unknown. Further action is continued watching
    now that snort is running on the public segment.

66666666666666666666666666666666666666666666666666666666666

77777777777777777777777777777777777777777777777777777777777

Log format:

```
date/time:         machine/facility:     rule:
action:interface:protocol:
src ip/port:      dst ip/port:  length:TOS:  IP ID: IP
Flags: TTL: rule#

Dec 23 17:08:51 fw kernel: Packet log: input REJECT eth3
PROTO=17
10.0.0.16:138 10.0.0.255:138 L=216 S=0x00 I=49943 F=0x0000
T=128 (#19)
Dec 23 17:09:15 fw kernel: Packet log: input REJECT eth3
PROTO=17
10.0.0.200:138 10.0.0.255:138 L=234 S=0x00 I=36748 F=0x0000
T=128 (#19)
Dec 23 17:09:17 fw kernel: Packet log: input REJECT eth3
PROTO=17
10.0.0.200:137 10.0.0.255:137 L=78 S=0x00 I=37516 F=0x0000
T=128 (#17)
Dec 23 17:09:48 fw kernel: Packet log: input REJECT eth3
PROTO=17
10.0.0.15:138 10.0.0.255:138 L=246 S=0x00 I=4096 F=0x0000
T=128 (#19)
Dec 23 17:10:08 fw kernel: Packet log: input REJECT eth3
PROTO=17
10.0.0.5:138 10.0.0.255:138 L=229 S=0x00 I=27910 F=0x0000
T=128 (#19)
Dec 23 17:10:18 fw kernel: Packet log: input REJECT eth3
PROTO=17
10.0.0.15:138 10.0.0.255:138 L=231 S=0x00 I=19712 F=0x0000
T=128 (#19)
Dec 23 17:10:40 fw kernel: Packet log: input REJECT eth3
PROTO=17
10.0.0.23:138 10.0.0.255:138 L=241 S=0x00 I=44376 F=0x0000
T=128 (#19)

Syslog from a system under test on our network.

Dec  27 11:42:11 tun1 /bsd: ARP information for 10.0.0.254
overwritten from ethernet address 00:D0:B7:0E:AA:9C
Dec  27 11:45:34 tun1 /bsd: ARP information for 10.0.0.254
verwritten from ethernet address 00:D0:B7:06:83:00

Notables:
* There are many source IPs involved.
* Thay match the actual distribution of our private
network.
* The TTLs are all the same and indicate they are probably
not from the
```

internet.
* They are logged because they are on the wrong ethernet interface of
the firewall.
* The packet sizes seem realistic.

Summary: For a few days users on our network were complaining that
internet response was sometimes very slow. During the same period
I was having mysterious session drops when using ssh to connect to
our firewall. tcpdump showed traffic that appeared to come from
our private network on the public interface of the firewall. The
break came when a system being built with OpenBSD was placed on the
internal network. It began producing the conflicting ARP messages
from the second block of logs above. The MAC addresses were for the
public and private interfaces of the firewall.

Technique: Fair. Someone felt that they could bypass the logging features
of our web proxy by attaching a cable from our private hub to the
hub built into our external router.

Intent: To evade traffic controls for our internet connection.

Severity: Medium. The traffic carried over this backdoor link is unknown.
Traffic may have been able to enter our network from the internet.
Service was degraded for all users during the several days this lasted.

Conclusion: The cable was removed. Locking policy for the room containing
the equipment was revisited.

77777777777777777777777777777777777777777777777777777777777

```
8888888888888888888888888888888888888888888888888888
88

Log format:
Rule:
date/time:          source IP/port:          dest IP/port:
Proto: TTL: TOS: IP ID: Length

[**] Traceroute [**]
04/06-22:47:56.083764 192.168.1.130:38459 ->
10.0.0.116:33489
UDP TTL:1 TOS:0x0 ID:23015 Len: 20

04/06-22:48:08.856300 192.168.1.130:38467 ->
10.0.0.115:33489
UDP TTL:1 TOS:0x0 ID:23493 Len: 20

04/06-22:48:25.482514 192.168.1.130:38483 ->
10.0.0.114:33489
UDP TTL:1 TOS:0x0 ID:24138 Len: 20

Notables:
* The trace has been trimmed, it is typical traceroute.
* They ran a traceroute to each IP in our public range.

Summary:  This detect came very soon after I installed
snort on
     our firewall.  Nslookup showed the source of the scan
to
     be a numbered system at a .edu.

Technique:  Poor.  They are mapping the route to each host
to
     determine subnetting and routes within our range.

Intent:  Network mapping.

Severity:  Low.  They can find that all our public servers
are
     behind our firewall.

Conclusion:  Standard network mapping.  No further action
needed.

8888888888888888888888888888888888888888888888888888
88
```

```
999999999999999999999999999999999999999999999999999999999
999999999

Log format:
date/time:         machine/facility:     rule:
action:interface:protocol:
src ip/port:      dst ip/port:  length:TOS:  IP ID: IP
Flags: TTL: rule#

Apr  5 07:05:57 fw kernel: Packet log: input REJECT eth3
PROTO=6
10.1.1.23:80 10.0.0.114:2111 L=40 S=0x00 I=11116 F=0x4000
T=118 (#11)
Apr  5 15:17:27 fw kernel: Packet log: input REJECT eth3
PROTO=6
10.1.1.23:80 10.0.0.114:1612 L=40 S=0x00 I=61508 F=0x4000
T=110 (#11)
Apr  6 09:37:09 fw kernel: Packet log: input REJECT eth3
PROTO=6
10.1.1.23:80 10.0.0.114:3759 L=40 S=0x00 I=21643 F=0x4000
T=110 (#11)

Notables:
* The timing is extremely slow.
* Don't fragment is set.
* The packet size matches nmaps default size for 'stealth'
SYN scan.
* Only the firewall's external IP was touched.

Summary:  Here is another detect from the logs on our
firewall.
     The timing is extremely slow.  Nslookup showed nothing
for this
     system, but traceroute to it ended with routers in
Poland.

Technique:  Good.  This detect shows activity at an
extremely low level.
     Scan detection tools would have missed this.  They
seem to know
     which host is our firewall and now they want to know
about its
     configuration.

Intent:  Careful, undetected host mapping.
```

Severity: High. This attacker seems to understand some of the pitfals
     of their activity and is willing to work very hard to avoid them.

Conclusion: Further action was to install snort in preparation for
     what may come.

99999999999999999999999999999999999999999999999999999999999999999999
10101010101010101010101010101010101010101010101010101010101010101010

Log format:
date/time: machine/facility: rule:
action:interface:protocol:
src ip/port: dst ip/port: length:TOS: IP ID: IP
Flags: TTL: rule#

Mar 30 10:14:15 fw kernel: Packet log: input REJECT eth3
PROTO=17
192.168.1.11:53 10.0.0.114:64726 L=232 S=0x00 I=53130
F=0x0000 T=44 (#107)
Mar 30 10:14:43 fw kernel: Packet log: input REJECT eth3
PROTO=17
192.168.1.11:53 10.0.0.114:64726 L=232 S=0x00 I=54829
F=0x0000 T=44 (#107)
Mar 30 10:15:39 fw kernel: Packet log: input REJECT eth3
PROTO=17
192.168.1.11:53 10.0.0.114:64726 L=232 S=0x00 I=58238
F=0x0000 T=44 (#107)

Mar 30 10:14:19 fw kernel: Packet log: input REJECT eth3
PROTO=17
192.168.5.32:53 10.0.0.114:64726 L=251 S=0x00 I=31367
F=0x0000 T=44 (#107)
Mar 30 10:14:51 fw kernel: Packet log: input REJECT eth3
PROTO=17
192.168.5.32:53 10.0.0.114:64726 L=251 S=0x00 I=37249
F=0x0000 T=44 (#107)
Mar 30 10:15:55 fw kernel: Packet log: input REJECT eth3
PROTO=17
192.168.5.32:53 10.0.0.114:64726 L=251 S=0x00 I=48756
F=0x0000 T=44 (#107)

```
Mar 30 10:14:11 fw kernel: Packet log: input REJECT eth3
PROTO=17
192.168.70.4:53 10.0.0.114:64726 L=232 S=0x00 I=44947
F=0x4000 T=242 (#107)
Mar 30 10:14:35 fw kernel: Packet log: input REJECT eth3
PROTO=17
192.168.70.4:53 10.0.0.114:64726 L=232 S=0x00 I=44949
F=0x4000 T=242 (#107)
Mar 30 10:15:23 fw kernel: Packet log: input REJECT eth3
PROTO=17
192.168.70.4:53 10.0.0.114:64726 L=232 S=0x00 I=44951
F=0x4000 T=242 (#107)

Mar 30 10:14:27 fw kernel: Packet log: input REJECT eth3
PROTO=17
192.168.222.4:53 10.0.0.114:64726 L=232 S=0x00 I=44948
F=0x4000 T=242 (#107)
Mar 30 10:15:07 fw kernel: Packet log: input REJECT eth3
PROTO=17
192.168.222.4:53 10.0.0.114:64726 L=232 S=0x00 I=44950
F=0x4000 T=242 (#107)

Notables:
* 3 machines/ 4 IPs involved in 'scanning' the dest. port
at the same time.
* 2 IPs are from .mil and 2 are from .edu
* One scan has a different length than the others.
* The delay increases rather than staying steady like OS
generated traffic
     would.  Each scan shows the same timing pattern.
* The IP ID numbers make sense.
* The TTLs are about correct or the IPs seen.

Summary:  Here our firewall is receiving packets on a
particular port from
     four sources at once.  The machines the packets come
from all run
     DNS.

Technique:  Fair.  There seems to be a scan of multiple
machines DNS going on
     using our IP.

Intent:  Network mapping with spoofed IPs as chaff.

Severity:  Low.  The .mil systems will be able to
understand this.  The .edu
```

systems are probably trafficed enough by this sort of thing to be OK.

Conclusion: Our IP is being spoofed on the network. Source unknown. No
further action possible.

10101010101010101010101010101010101010101010101010101010101010101010101010

Last Updated: July 26th, 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 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
Baltimore Fall 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
SANS Baltimore Fall 2017	Baltimore, MD	Sep 25, 2017 - Sep 30, 2017	Live Event
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