Global Information Assurance Certification Paper

GCIA Practical Exam
Rinaldo Ribeiro

1/17/2005

*** Northcutt, I should probably recuse myself, I know Rinaldo, he was an IDNET winner for one thing. He must of put a tremendous effort into the writeup since English is a second language for him so I am giving full credit for clarity. Good research into the source addresses. Analysis shows solid depth of knowledge, this is another example of the look and feel of a process, but the intuitive leaps of someone in the trade. Took some easy ones though. 88 **

GCIA Practical Exam

Rinaldo Ribeiro

Background:
All detects were retrieved from a network with two OpenBSD boxes: one running IPF - firewall and the other running Snort as our IDS.
This site was installed at november/99 and all its logs were used whenever the "history" was necessary.
There are only servers (web servers mainly) behind the firewall and all the expected traffic was not "user generated" leaving our network.
Only the destination network address has been sanitized.
Whenever possible, I used both traces (FW and IDS) to illustrate what were the "attacker" intents.

Severity is calculated using:

Severity = (Criticality + Lethality) – (system countermeasures + network countermeasures)

GCIA Practical Exam

Rinaldo Ribeiro

1/17/2005

GCIA Practical Exam

Rinaldo Ribeiro

1/17/2005

Detect #01

Detect Information: This trace was captured on the firewall.

```
Apr 10 02:07:51 kundun ipmon[1594]: 02:07:50.594491    ep1 @0:80 b 209.67.42.160,2900 -> x.y.z..33,53 PR tcp len 20 104 -S IN
Apr 10 02:07:51 kundun ipmon[1594]: 02:07:50.595656    ep1 @0:80 b 209.67.42.160,2901 -> x.y.z..33,53 PR tcp len 20 104 -S IN
Apr 10 02:07:51 kundun ipmon[1594]: 02:07:50.595771    ep1 @0:80 b 209.67.42.160,2902 -> x.y.z..33,53 PR tcp len 20 104 -S IN

Apr 10 03:02:28 kundun ipmon[1594]: 03:02:28.320739    ep1 @0:80 b 209.67.42.160,3800 -> x.y.z..12,53 PR tcp len 20 104 -S IN
Apr 10 03:02:28 kundun ipmon[1594]: 03:02:28.320848    ep1 @0:80 b 209.67.42.160,3801 -> x.y.z..12,53 PR tcp len 20 104 -S IN
Apr 10 03:02:28 kundun ipmon[1594]: 03:02:28.320952    ep1 @0:80 b 209.67.42.160,3802 -> x.y.z..12,53 PR tcp len 20 104 -S IN

Apr 10 03:13:13 kundun ipmon[1594]: 03:13:13.266251    ep1 @0:80 b 209.67.42.148,3000 -> x.y.z..12,53 PR tcp len 20 104 -S IN
Apr 10 03:13:13 kundun ipmon[1594]: 03:13:13.266858    ep1 @0:80 b 209.67.42.148,3001 -> x.y.z..12,53 PR tcp len 20 104 -S IN
Apr 10 03:13:13 kundun ipmon[1594]: 03:13:13.267027    ep1 @0:80 b 209.67.42.148,3002 -> x.y.z..12,53 PR tcp len 20 104 -S IN

Apr 10 05:24:49 kundun ipmon[1594]: 05:24:48.811312    ep1 @0:80 b 209.67.42.185,2900 -> x.y.z..12,53 PR tcp len 20 104 -S IN
Apr 10 05:24:49 kundun ipmon[1594]: 05:24:48.812251    ep1 @0:80 b 209.67.42.185,2901 -> x.y.z..12,53 PR tcp len 20 104 -S IN
Apr 10 05:24:49 kundun ipmon[1594]: 05:24:48.812369    ep1 @0:80 b 209.67.42.185,2902 -> x.y.z..12,53 PR tcp len 20 104 -S IN
```

Existence: Packets from net 209.67.42 are being sent to our servers' TCP 53 port.

History: Our firewall has been blocking packets sent from this network to port TCP 53 since the end of last year. It has been identified also packets sent to high udp ports, seeming to be originated by a "unix like" traceroute program.

GCIA Practical Exam

Rinaldo Ribeiro

1/17/2005

Intent: The intent is collect RTT – "round trip time" information about how long it takes traffic to move between "client" and "server".

Technique: Three SYN packets are sent always in less than one second by a automated tool to TCP 53 port. The source port always increments by one and it starts between 2000 and 4000.

Analysis: This traffic is generated by a "global load-balancing system" called 3DNS, manufactured by F5 labs. In this case, everytime ours servers tried to resolve www.starmedia.com, some systems determine what is the closest server to my site sending packets as seen in this detect. It was considered by me, some time ago, as a "zone transfer" attempt and it could be done by another analyst if we had only a single packet. Some research was done to figure out all the picture:

1) who is 209.67.42.160?

   By accessing www.registro.br, maintained by FAPESP, the organization responsible for the domains in Brazil, we could find that the network 209.67.42 was registered in some starmedia.com's name servers.

2) getting in touch!

   The adminstrator of this domain was notified by e-mail and few days later I received a huge and complete answer about the "probe". It said "the traffic you're seeing can safely be ignored" and "if it's a real problem for you, I can hard-code your netblocks to our Brazil datacenter"... "that'll stop the probes".

3) looking at our logs again!

After this was much easier to find another "probe", done by another "RTT collect" tool and usually trying to access our server by a "unix like" traceroute before sending "TCP 53" packets.

GCIA Practical Exam

Rinaldo Ribeiro

1/17/2005

Severity:

Criticality	5	Web servers
Lethality	1	Just getting the RTT.
System Countermeasures	5	All operating systems are running the latest patches and don't allow "zone transfers" from a name server that isn't a secondary.
Network Countermeasures	4	Firewall blocks this traffic.
Severity Score	-3	Severity = (Criticality + Lethality) – (system countermeasures + net countermeasures

Detect #02

Detect Information: This trace was captured on the firewall.

```
Apr 11 08:59:10 kundun ipmon[9669]: 08:59:10.153493        ep1 @0:82 b 200.215.4.161,3130 -> x.y.z.11,7 PR udp len 20 82  IN
Apr 11 08:59:11 kundun ipmon[9669]: 08:59:10.891487 5x        ep1 @0:82 b 200.215.4.161,3130 -> x.y.z.11,7 PR udp len 20 102  IN
Apr 11 08:59:12 kundun ipmon[9669]: 08:59:11.577640 4x        ep1 @0:82 b 200.215.4.161,3130 -> x.y.z.11,7 PR udp len 20 100  IN
Apr 11 08:59:13 kundun ipmon[9669]: 08:59:12.511945 6x        ep1 @0:82 b 200.215.4.161,3130 -> x.y.z.11,7 PR udp len 20 92  IN
Apr 11 08:59:14 kundun ipmon[9669]: 08:59:13.517143 2x        ep1 @0:82 b 200.215.4.161,3130 -> x.y.z.11,7 PR udp len 20 97  IN
Apr 11 08:59:15 kundun ipmon[9669]: 08:59:14.669449 3x        ep1 @0:82 b 200.215.4.161,3130 -> x.y.z.11,7 PR udp len 20 94  IN
Apr 11 08:59:16 kundun ipmon[9669]: 08:59:15.698395 3x        ep1 @0:82 b 200.215.4.161,3130 -> x.y.z.11,7 PR udp len 20 93  IN
Apr 11 08:59:17 kundun ipmon[9669]: 08:59:16.641983 2x        ep1 @0:82 b 200.215.4.161,3130 -> x.y.z.11,7 PR udp len 20 106  IN
```

Existence: UDP packets are being sent to echo port of our main web server.

GCIA Practical Exam

Rinaldo Ribeiro

1/17/2005

History: We have been seeing this packets claiming to be sent by 200.215.4 network since the end of last year, when I sent some traces to GIAC. No other hostile traffic has been identified from this network in our firewall or IDS.

Intent: Collect information to populate a web proxy cache.

Technique: UDP packets are sent to port 7 (echo) with a fixed source port -3130 - to our main web server. The header lenght is always 20 bytes and the payload is variable. Its done by an automated tool that sends even 6 packets per second.

Analysis: Some research was necessary to comprehend this traffic. This packets are from ICP protocol – Internet Cache Protocol – and they were sent by a proxy server, trying to populate its database. The source port used –3130 – is the default port and recommended on the RFC 2186/2187. Because this protocol deals with web access, it must be fast and that's the reason of 6 packets in less than one second. The RFC says that if a proxy server using ICP can't find a record in its own database or in its neighbor's, its possible to access directly the origin server. And that what is represented in this detect. The fixed 20 bytes of header and the variable payload is the "signature" of this protocol. Notice that each packet has a different payload. Again, it was necessary to get in touch with a system admin by e-mail to notify them and ask for stopping the "probe".

Some parts of the RFC that prove what I am saying:

RFC 2186

ICP is a message format used for communicating between Web caches.
ICP is implemented on top of UDP
An ICP query/reply exchange needs to occur quickly, typically within a second or two. *(6 packets per second is fast, uh?*)
The ICP message format consists of a 20-octet fixed header plus a **variable sized payload** (*bingo*!).

GCIA Practical Exam

Rinaldo Ribeiro

1/17/2005

RFC2187

ICP may be a "net win" in some situations, and a *"**net loss**"* in others. We recognize that certain practices described in this document are suboptimal. (*maybe that's my case.. :* -)

When a cache does not hold a requested object, it may ask via ICP whether any of its neighbor caches has the object. If any of the neighbors does have the requested object (i.e., a "neighbor hit"), then the cache will request it from them. If none of the neighbors has the object (a "neighbor miss"), then the cache must forward the request either to a parent, or **directly to the origin server.**

Severity:

Criticality	5	Our main web server.
Lethality	2	Udp packets to echo port.
System Countermeasures	5	All the servers behind the firewall don't listen to this port and they have the latest patches.
Network Countermeasures	4	Firewall blocks this traffic.
Severity Score	-2	Severity = (Criticality + Lethality) – (system countermeasures + net countermeasures

Detect #03

Detect Information: This trace was captured on the firewall and on the IDS.

```
Apr 10 23:24:01 kundun ipmon[1594]: 23:24:01.230185       ep1 @0:82 b 171.209.161.198,31338 -> x.y.z.1,31337 PR udp len 20 47  IN
Apr 10 23:24:01 kundun ipmon[1594]: 23:24:01.956161       ep1 @0:82 b 171.209.161.198,31338 -> x.y.z.4,31337 PR udp len 20 47  IN
```

GCIA Practical Exam

Rinaldo Ribeiro

1/17/2005

```
Apr 10 23:24:01 kundun ipmon[1594]: 23:24:01.956191       ep1 @0:82 b 171.209.161.198,31338 -> x.y.z.5,31337 PR udp len 20 47  IN
Apr 10 23:24:01 kundun ipmon[1594]: 23:24:01.956788       ep1 @0:82 b 171.209.161.198,31338 -> x.y.z.2,31337 PR udp len 20 47  IN
Apr 10 23:24:01 kundun ipmon[1594]: 23:24:01.956816       ep1 @0:82 b 171.209.161.198,31338 -> x.y.z.3,31337 PR udp len 20 47  IN
Apr 10 23:24:02 kundun ipmon[1594]: 23:24:02.925654       ep1 @0:82 b 171.209.161.198,31338 -> x.y.z.6,31337 PR udp len 20 47  IN
Apr 10 23:24:03 kundun ipmon[1594]: 23:24:03.332320       ep1 @0:82 b 171.209.161.198,31338 -> x.y.z.10,31337 PR udp len 20 47  IN
Apr 10 23:24:03 kundun ipmon[1594]: 23:24:03.332349       ep1 @0:82 b 171.209.161.198,31338 -> x.y.z.12,31337 PR udp len 20 47  IN
Apr 10 23:24:03 kundun ipmon[1594]: 23:24:03.332375       ep1 @0:82 b 171.209.161.198,31338 -> x.y.z.8,31337 PR udp len 20 47  IN
...
```

```
Apr 10 23:24:01 IDS snort: Back Orifice: 171.209.161.198:31338 -> x.y.z.1:31337
Apr 10 23:24:01 IDS snort: Back Orifice: 171.209.161.198:31338 -> x.y.z.4:31337
Apr 10 23:24:01 IDS snort: Back Orifice: 171.209.161.198:31338 -> x.y.z.5:31337
Apr 10 23:24:01 IDS snort: Back Orifice: 171.209.161.198:31338 -> x.y.z.2:31337
Apr 10 23:24:01 IDS snort: Back Orifice: 171.209.161.198:31338 -> x.y.z.3:31337
Apr 10 23:24:02 IDS snort: Back Orifice: 171.209.161.198:31338 -> x.y.z.6:31337
Apr 10 23:24:03 IDS snort: Back Orifice: 171.209.161.198:31338 -> x.y.z.10:31337
Apr 10 23:24:03 IDS snort: Back Orifice: 171.209.161.198:31338 -> x.y.z.12:31337
Apr 10 23:24:03 IDS snort: Back Orifice: 171.209.161.198:31338 -> x.y.z.8:31337
```

Existence: Someone claiming to be from 171.209.161.198 (ABD1A1C6.ipt.aol.com) is visiting us.

History: No history is available. No previous traffic was identified on our firewall or IDS from this host or from its network.

Intent: Host scan looking for a trojan horse – back orifice.

Technique: A automated tool is sending udp packets to port 31337 with a fixed source port – 31338.. Sometimes 5 packets are sent in less then one second. The packets are not arriving out or order, they have been sent out of order.

Analysis: This is a classic example of a back orifice client looking for a server in our address space. These packets are probably not spoofed. With a fixed source port (31338) this was definitely generated by an automated tool.

Severity:

Criticality	3	All my address space, not a specific target.
Lethality	4	Back Orifice is a lethal tool.
System Countermeasures	5	All the servers behind the firewall don't listen to this port and they have the latest patches.
Network Countermeasures	4	Firewall blocks this traffic.
Severity Score	-2	Severity = (Criticality + Lethality) – (system countermeasures + net countermeasures

Detect #04

Detect Information: This trace was captured on the firewall.

```
Apr  9 15:18:13 kundun ipmon[1594]: 15:18:13.409212        ep1 @0:80 b 141.30.122.20,40239 -> x.y.z.1,81 PR tcp len 20 44 -S IN
...
Apr  9 15:18:13 kundun ipmon[1594]: 15:18:13.428789        ep1 @0:80 b 141.30.122.20,40250 -> x.y.z.12,81 PR tcp len 20 44 -S IN
Apr  9 15:18:13 kundun ipmon[1594]: 15:18:13.459482        ep1 @0:80 b 141.30.122.20,40239 -> x.y.z.1,81 PR tcp len 20 40 -R IN
...
Apr  9 15:18:13 kundun ipmon[1594]: 15:18:13.477595        ep1 @0:80 b 141.30.122.20,40250 -> x.y.z.12,81 PR tcp len 20 40 -R IN
Apr  9 15:18:13 kundun ipmon[1594]: 15:18:13.478668        ep1 @0:80 b 141.30.122.20,40251 -> x.y.z.13,81 PR tcp len 20 44 -S IN
Apr  9 15:18:13 kundun ipmon[1594]: 15:18:13.480074        ep1 @0:80 b 141.30.122.20,40252 -> x.y.z.14,81 PR tcp len 20 44 -S IN
Apr  9 15:18:13 kundun ipmon[1594]: 15:18:13.482111        ep1 @0:80 b 141.30.122.20,40253 -> x.y.z.15,81 PR tcp len 20 44 -S IN
...
Apr  9 15:18:13 kundun ipmon[1594]: 15:18:13.513918        ep1 @0:80 b 141.30.122.20,40251 -> x.y.z.13,81 PR tcp len 20 40 -R IN
```

GCIA Practical Exam

Rinaldo Ribeiro

1/17/2005

```
Apr  9 15:18:13 kundun ipmon[1594]: 15:18:13.514346        ep1 @0:80 b 141.30.122.20,40254 -> x.y.z.16,81 PR tcp len 20 40 -R IN
Apr  9 15:18:13 kundun ipmon[1594]: 15:18:13.514375        ep1 @0:80 b 141.30.122.20,40255 -> x.y.z.17,81 PR tcp len 20 40 -R IN
...
Apr  9 15:18:23 kundun ipmon[1594]: 15:18:23.275571        ep1 @0:44 b 141.30.122.20,48153 -> k.l.m.254,81 PR tcp len 20 44 -S IN
Apr  9 15:18:23 kundun ipmon[1594]: 15:18:23.301630        ep1 @0:44 b 141.30.122.20,48153 -> k.l.m.254,81 PR tcp len 20 40 -R IN
```

Existence: Someone claiming to be from 141.30.122.20 (eiet20.et.tu-dresden.de) is visiting us.

History: No history is available. No previous traffic was identified on our firewall or IDS from this host or from its network.

Intent: There are at least two possibilities: 1) trying to find hosts running a service on 81 TCP, maybe a trojan horse 2) a recon probe. If someone was just looking for a "TCP 81 service", why send crafted RESET packets? So my guess is the second option.

Technique: A automated tool is doing a very very fast host scan on TCP 81. Only 12 hosts are probed on the first "phase", sending 12 SYN followed by 12 RESET packets. After this, all the address space is probed. After trying to open a connection sending a SYN packet, approximately half second later a RESET packet is sent using the same source port used in the SYN packet.

Analysis: The first observation is that about the speed. The last two packets were targeting another network, and they arrive only 10 seconds after the last packet in our net x.y.z. It shows how fast the scan is. The "attacker" doesn't know my address space but he can target at least two entire class C network in less than 10 seconds. Its really fast.
We see RESET after SYN packets using the same source port. Its looking for something at the probable SYN/ACK response, like the TCP sequence number for example, or its maybe trying to make a recon scan , looking for "unreachable" or "admin-prohibited" messages from a router/firewall, to figure out how are my networks. But it seems that some oracle web servers run at 81 TCP by default, maybe he is looking for one to exploit.
Definitely its done by an automated tool that crafts the packets.
Every connection is closed sending a RESET after half second.
This could be done (specially the "first phase") easily by a tool like hping2.

GCIA Practical Exam

Rinaldo Ribeiro

1/17/2005

Severity:

Criticality	3	All my address space, not a specific target.
Lethality	4	Fast scan can be dangerous and sometimes interpreted as a DoS attack.
System Countermeasures	5	All the servers behind the firewall don't listen to this port and they have the latest patches.
Network Countermeasures	4	Firewall blocks this traffic.
Severity Score	-2	Severity = (Criticality + Lethality) – (system countermeasures + net countermeasures

Detect #05

Detect Information: This trace was captured on the firewall.

```
Apr 10 21:40:37 kundun ipmon[1594]: 21:40:36.918363 ep1 @0:81 b 200.231.55.10 -> x.y.z.11 PR tcp len 20 (64) frag 44@456 IN
Apr 10 21:40:43 kundun ipmon[1594]: 21:40:42.953649 ep1 @0:81 b 200.231.55.10 -> x.y.z.11 PR tcp len 20 (64) frag 44@456 IN
Apr 10 21:40:55 kundun ipmon[1594]: 21:40:54.951133 ep1 @0:81 b 200.231.55.10 -> x.y.z.11 PR tcp len 20 (64) frag 44@456 IN
Apr 10 21:41:19 kundun ipmon[1594]: 21:41:18.917114 ep1 @0:81 b 200.231.55.10 -> x.y.z.11 PR tcp len 20 (64) frag 44@456 IN
Apr 10 21:45:42 kundun ipmon[1594]: 21:45:42.713330 ep1 @0:81 b 200.231.55.10 -> x.y.z.11 PR tcp len 20 (64) frag 44@456 IN
Apr 10 21:49:42 kundun ipmon[1594]: 21:49:42.528964 ep1 @0:81 b 200.231.55.10 -> x.y.z.11 PR tcp len 20 (64) frag 44@456 IN
Apr 10 21:51:42 kundun ipmon[1594]: 21:51:42.539338 ep1 @0:81 b 200.231.55.10 -> x.y.z.11 PR tcp len 20 (64) frag 44@456 IN
Apr 10 21:53:42 kundun ipmon[1594]: 21:53:42.570069 ep1 @0:81 b 200.231.55.10 -> x.y.z.11 PR tcp len 20 (64) frag 44@456 IN
Apr 10 21:55:42 kundun ipmon[1594]: 21:55:42.454746 ep1 @0:81 b 200.231.55.10 -> x.y.z.11 PR tcp len 20 (64) frag 44@456 IN
```

GCIA Practical Exam

Rinaldo Ribeiro

1/17/2005

```
Apr 10 21:57:42 kundun ipmon[1594]: 21:57:42.439387   ep1 @0:81 b 200.231.55.10 -> x.y.z.11 PR tcp len 20 (64) frag 44@456 IN
Apr 10 21:59:42 kundun ipmon[1594]: 21:59:42.597223   ep1 @0:81 b 200.231.55.10 -> x.y.z.11 PR tcp len 20 (64) frag 44@456 IN
Apr 10 22:01:42 kundun ipmon[1594]: 22:01:42.424332   ep1 @0:81 b 200.231.55.10 -> x.y.z.11 PR tcp len 20 (64) frag 44@456 IN
Apr 10 22:03:42 kundun ipmon[1594]: 22:03:42.436523   ep1 @0:81 b 200.231.55.10 -> x.y.z.11 PR tcp len 20 (64) frag 44@456 IN
```

Existence: Many fragments from 200.231.55.10 (sjclp05.valley-bbs.com.br) sent to our main web server.

History: Many fragmented packets were blocked and logged in our firewall since March/99 coming from the same network – (200.231.55.0/32).

Intent: A normal connection trying to access our main web server.

Technique: Repeated fragmented packets were sent to our main web server. The firewall is blocking some data causing timeout to the application, that is re-transmitting the packets.

Analysis: Since these packets are not the first fragment we can't see much about the header and possible flags. We can't see also if this is the last fragment, with the DF flag set. All packets were sent to our main web server and they have the same offset and data. They are not overlapping previous data... The IP filter firewall has an feature that we are not using: "keep frags". It means that fragments are not welcome...

So its totally possible that these packets are coming from a normal connection to our web server and probably have the DF set, as a last normal fragment. In fact in this case, the second and last packet... My guess is a router in the middle with a misconfigured MTU, making disnecessary fragmentation.
Notice that the fragments are coming from the client to the server and to be part of a normal connection we must check if it was possible to make a request that could be fragmented. Using tcpdump (tcpdump –n –l –x –s 1500 –i ep1 'tcp[13] &

0x08 != 0 and src net ournet.com and port = 80' | tcpdecode.pl') I realized that the requests to our main web server (x.y.z.11) has approximately 550 bytes because of large URLs and cookies.

The time between the packets after the fifth one is 2 minutes exactly. This could be sent by the application, resending the packet after a timeout.

A e-mail message was sent to valley-bbs' system administrator alerting the possible mistake but no answer yet.

* After sending this e-mail message, the valley-bbs' sys admin answered telling that they are really having problems with their "radio link" and many "bizarre" fragmentation problems have been observed. They are working hard to solve this problem. ;)

Severity:

Criticality	5	My main web server.
Lethality	1	Normal connections!
System Countermeasures	4	All the servers behind the firewall have the latest patches.
Network Countermeasures	4	Firewall blocks this traffic.
Severity Score	-2	Severity = (Criticality + Lethality) – (system countermeasures + net countermeasures

Detect #06

Detect Information: This trace was captured on the firewall.

Apr 11 04:27:52 kundun ipmon[1594]: 04:27:51.909708 ep1 @0:81 b 210.102.139.1,53 -> x.y.z.20,111 PR tcp len 20 40 -S IN

GCIA Practical Exam

Rinaldo Ribeiro

1/17/2005

```
Apr 11 05:06:14 kundun ipmon[1594]: 05:06:14.072829       ep1 @0:81 b 208.243.231.9,53 -> x.y.z.1,111 PR tcp len 20 40 -S IN
Apr 11 09:28:00 kundun ipmon[9669]: 09:28:00.528732       ep1 @0:81 b 210.102.139.1,53 -> x.y.z.21,111 PR tcp len 20 40 -S IN
Apr 11 09:58:05 kundun ipmon[9669]: 09:58:05.894213       ep1 @0:81 b 208.243.231.9,53 -> x.y.z.2,111 PR tcp len 20 40 -S IN
Apr 11 14:22:39 kundun ipmon[9669]: 14:22:38.728581       ep1 @0:81 b 210.102.139.1,53 -> x.y.z.22,111 PR tcp len 20 40 -S IN
Apr 11 14:58:59 kundun ipmon[9669]: 14:58:59.260734       ep1 @0:81 b 208.243.231.9,53 -> x.y.z.3,111 PR tcp len 20 40 -S IN
Apr 11 19:06:11 kundun ipmon[9669]: 19:06:11.024305       ep1 @0:81 b 210.102.139.1,53 -> x.y.z.23,111 PR tcp len 20 40 -S IN
Apr 11 19:52:43 kundun ipmon[9669]: 19:52:42.482054       ep1 @0:81 b 208.243.231.9,53 -> x.y.z.4,111 PR tcp len 20 40 -S IN
...
```

Existence: Probes to portmapper are coming from two different hosts from different networks.

History: Our firewall was blocking tcp packets to portmapper firstly only from 210.102.139.1 (April, 9th) and then both hosts started sending this packets together.

Intent: Find a server with portmapper enabled as part of a recon probe to find out which remote procedures call services are available and then maybe try to exploit someone.

Technique:. Each packet is sent sometimes in 4 hour intervals. One starts scanning host from .1 and up and the other from .20 . The source port used is TCP 53 in both cases.

Analysis: This is a slow and slow coordinated scan made by two different hosts. Its trying to elude us by sending packets slowly. While one starts scanning from .1 host the other one starts from .20. Probably the hosts used in this scan were "0wn3d" by the same person who is controlling the probe and dealing with the answers.

Severity:

Criticality	3	All my address space, not a specific target.
Lethality	4	Coordinated scans preparing a future attack.
System Countermeasures	4	All the servers have the latest patches.
Network Countermeasures	4	Firewall blocks this traffic.
Severity Score	-1	Severity = (Criticality + Lethality) – (system countermeasures + net countermeasures

Detect #07

Detect Information: This trace was captured on the firewall and on the IDS.

```
Apr  9 02:26:46 kundun ipmon[1594]: 02:26:46.337414        ep1 @0:80 b 200.203.184.163,0 -> x.y.z.25,1592 PR tcp len 20 40 -FUP IN
Apr 10 10:29:01 kundun ipmon[1594]: 10:29:00.213698      ep1 @0:80 b 150.162.110.68,0 -> x.y.z.11,1321 PR tcp len 20 40 -RSUP IN
Apr 11 07:51:06 kundun ipmon[9669]: 07:51:06.420108       ep1 @0:81 b 200.213.105.162,0 -> x.y.z.25,1080 PR tcp len 20 40 -RSF IN
Apr 11 10:28:50 kundun ipmon[9669]: 10:28:49.903834         ep1 @0:81 b 200.137.194.49,0 -> x.y.z.11,1157 PR tcp len 20 48 - IN
Apr 11 12:55:31 kundun ipmon[9669]: 12:55:30.326333      ep1 @0:81 b 200.194.103.59,0 -> x.y.z.11,1159 PR tcp len 20 40 -ARSFP IN
Apr 11 21:02:51 kundun ipmon[9669]: 21:02:50.922102         ep1 @0:81 b 200.205.95.5,0 -> x.y.z.11,3894 PR tcp len 20 40 -SFU IN
Apr 11 21:25:12 kundun ipmon[9669]: 21:25:11.782818         ep1 @0:81 b 200.205.95.5,0 -> x.y.z.11,4317 PR tcp len 20 40 -SF IN
Apr 12 08:08:05 kundun ipmon[9669]: 08:08:05.344596        ep1 @0:81 b 200.145.78.101,0 -> x.y.z.11,1113 PR tcp len 20 478 -RS IN
Apr 12 08:11:41 kundun ipmon[9669]: 08:11:41.124480        ep1 @0:81 b 200.145.78.101,0 -> x.y.z.11,1155 PR tcp len 20 543 -RFP IN
```

Apr 11 10:28:05 **IDS** snort: NULL Scan: 200.137.194.49:0 -> x.y.z.11:1157

Existence: Many different hosts are sending packets with source port 0 (zero) and bogus TCP flags to our web servers.

History: No history available.

GCIA Practical Exam

Rinaldo Ribeiro

1/17/2005

Intent: These packets are part of a recon probe, trying a OS identification or a recon scan.

Technique: Crafted packets are being sent with bogus TCP flags for days to our web servers. The source port used is zero and the destination port is random. There is no relation between the time stamps.

Analysis: By sending packets with bogus flags someone is trying to make a recon scan. Since all the packets are sent to unknown services, except socks – 1080, the main goal is probably make a recon scan taking advantage of a packet filter that sometimes can send back a "unreachable" message reveling information about the network and about the implemented filters. Or maybe the attacker is trying to evade defenses...

Certainly these packets are not from a normal connection, since there is no reason to a packet come with a source port equal to zero and weird flag combinations like SYN and FIN together on the same packet.

Severity:

Criticality	4	Web servers.
Lethality	2	Recon probe.
System Countermeasures	5	All the servers behind the firewall don't listen to these ports and they have the latest patches.
Network Countermeasures	4	Firewall blocks this traffic.
Severity Score	-3	Severity = (Criticality + Lethality) – (system countermeasures + net countermeasures

Detect #08

GCIA Practical Exam

Rinaldo Ribeiro

1/17/2005

Detect Information: This trace was captured on the firewall and on the IDS.

```
Apr  9 02:16:37 kundun ipmon[1594]: 02:16:37.067357        ep1 @0:80 b 200.192.100.170,1999 -> x.y.z.1,5742 PR tcp len 20 44 -S IN
Apr  9 02:16:37 kundun ipmon[1594]: 02:16:37.136755        ep1 @0:80 b 200.192.100.170,2000 -> x.y.z.2,5742 PR tcp len 20 44 -S IN
Apr  9 02:16:37 kundun ipmon[1594]: 02:16:37.243485        ep1 @0:80 b 200.192.100.170,2001 -> x.y.z.3,5742 PR tcp len 20 44 -S IN
Apr  9 02:16:37 kundun ipmon[1594]: 02:16:37.328398        ep1 @0:80 b 200.192.100.170,2002 -> x.y.z.4,5742 PR tcp len 20 44 -S IN
Apr  9 02:16:37 kundun ipmon[1594]: 02:16:37.405075        ep1 @0:80 b 200.192.100.170,2003 -> x.y.z.5,5742 PR tcp len 20 44 -S IN
...
Apr  9 02:17:34 kundun ipmon[1594]: 02:17:34.628558        ep1 @0:80 b 200.192.100.170,2258 -> x.y.z.1,2583 PR tcp len 20 44 -S IN
Apr  9 02:17:35 kundun ipmon[1594]: 02:17:34.833286        ep1 @0:80 b 200.192.100.170,2259 -> x.y.z.2,2583 PR tcp len 20 44 -S IN
Apr  9 02:17:35 kundun ipmon[1594]: 02:17:34.967248        ep1 @0:80 b 200.192.100.170,2260 -> x.y.z.3,2583 PR tcp len 20 44 -S IN
Apr  9 02:17:35 kundun ipmon[1594]: 02:17:35.048405        ep1 @0:80 b 200.192.100.170,2261 -> x.y.z.4,2583 PR tcp len 20 44 -S IN
Apr  9 02:17:35 kundun ipmon[1594]: 02:17:35.123960        ep1 @0:80 b 200.192.100.170,2262 -> x.y.z.5,2583 PR tcp len 20 44 -S IN
...

Apr  9 02:16:37 IDS snort: BACKDOOR ATTEMPT-WinCrash: 200.192.100.170:1999 -> x.y.z.1:5742
Apr  9 02:16:37 IDS snort: BACKDOOR ATTEMPT-WinCrash: 200.192.100.170:2000 -> x.y.z.2:5742
Apr  9 02:16:37 IDS snort: BACKDOOR ATTEMPT-WinCrash: 200.192.100.170:2001 -> x.y.z.3:5742
...
Apr  9 02:17:34 IDS snort: BACKDOOR ATTEMPT-WinCrash2: 200.192.100.170:2258 -> x.y.z.1:2583
Apr  9 02:17:35 IDS snort: BACKDOOR ATTEMPT-WinCrash2: 200.192.100.170:2259 -> x.y.z.2:2583
Apr  9 02:17:35 IDS snort: BACKDOOR ATTEMPT-WinCrash2: 200.192.100.170:2260 -> x.y.z.3:2583
...
```

Existence: Someone claiming to be from 200.192.100.170 (modem140-pgo.convoy.com.br) is looking for opened 5742 and 2583 TCP ports.

History: No previous attempt is available from this host in our firewall or IDS logs.

GCIA Practical Exam

Rinaldo Ribeiro

1/17/2005

Intent: Find a machine infected with trojan horses, WinCrash1 or WinCrash2.

Technique: Packets are being sent to our address space looking for a infected machine running WinCrash. The source port is always incremented by one and the destination port is the default port of WinCrash1 and WinCrash2.

Analysis: A automated tool is making a fast and noisy scan in our address space looking for WinCrash1 and WinCrash2. After looking for WinCrash1 a new scan was made trying to find WinCrash2. Since the "attacker" is probably not worried about being discovered there are 3 options: 1) The machine that is originating these packets was "owned" by some "hacker" 2) All the packets were spoofed 3) This guy is really a newbie or a real lamer.

Severity:

Criticality	3	All my address space, not a specific target.
Lethality	4	WinCrash is a lethal tool.
System Countermeasures	4	All the servers behind the firewall don't listen to these ports
Network Countermeasures	4	Firewall blocks this traffic.
Severity Score	-1	Severity = (Criticality + Lethality) – (system countermeasures + net countermeasures

GCIA Practical Exam

Rinaldo Ribeiro

1/17/2005

Detect #09

Detect Information: This trace was captured on the IDS.

```
Apr 11 15:42:09 IDS snort: SNMP public access: 206.77.145.97:2200 -> x.y.z.0:161
Apr 11 16:25:03 IDS snort: SNMP public access: 206.77.145.97:2200 -> x.y.z.1:161
Apr 11 17:13:38 IDS snort: SNMP public access: 206.77.145.97:2200 -> x.y.z.2:161
Apr 11 18:00:31 IDS snort: SNMP public access: 206.77.145.97:2200 -> x.y.z.3:161
Apr 11 18:47:57 IDS snort: SNMP public access: 206.77.145.97:2200 -> x.y.z.4:161
Apr 11 19:34:56 IDS snort: SNMP public access: 206.77.145.97:2200 -> x.y.z.5:161
Apr 11 20:22:01 IDS snort: SNMP public access: 206.77.145.97:2200 -> x.y.z.6:161
Apr 11 21:08:52 IDS snort: SNMP public access: 206.77.145.97:2200 -> x.y.z.7:161
Apr 11 21:54:27 IDS snort: SNMP public access: 206.77.145.97:2200 -> x.y.z.8:161
Apr 11 22:41:08 IDS snort: SNMP public access: 206.77.145.97:2200 -> x.y.z.9:161
...
```

Existence: Someone claiming to be from 206.77.145.97(m20677145097.austin.cc.tx.us) is visiting us.

History: No previous attempt is available from this host in our firewall or IDS logs.

Intent: Access an snmp agent using the default community string “public”.

Technique: Using the same source port, 2200, many packets are being sent to our address space looking for a snmp agent. There is approximately 47 minutes between each time stamp.

Analysis: It starts sending a packet to x.y.z.0, the old BSD’s style of broadcast address, maybe waiting for a response from all alive hosts and perhaps all the subsequent packets are sent because no answer is given to the first packet. Certainly an automated tool is sending these packets expecting to find a device using the default snmp community

"public". If any access is possible using at least a read only community many useful information can be obtained to a future "attack". Since this traffic is coming from austin.cc.tx.us and we not accept access to port udp 161 the possibility of a misconfigured SNMP manager is not being considered.

Severity:

Criticality	3	All my address space, not a specific target.
Lethality	3	"Read only" snmp access attempt.
System Countermeasures	4	All the servers behind the firewall don't listen to these ports
Network Countermeasures	4	Firewall blocks this traffic.
Severity Score	-2	Severity = (Criticality + Lethality) – (system countermeasures + net countermeasures

Detect #10

Detect Information: This trace was captured on the IDS.

Mar 12 16:23:29 IDS snort: FrontPage Author PWD Scan: 200.245.108.150:3232 -> x.y.z.11:80
Mar 12 16:23:44 IDS snort: FrontPage-shtml.dll: 200.245.108.150:3249 -> x.y.z.11:80
Mar 12 16:23:46 IDS snort: FrontPage-shtml.exe: 200.245.108.150:3252 -> x.y.z.11:80
Mar 12 16:24:08 IDS snort: FrontPage Service PWD Scan: 200.245.108.150:3269 -> x.y.z.11:80
Mar 12 16:24:10 IDS snort: FrontPage User PWD Scan: 200.245.108.150:3271 -> x.y.z.11:80
Mar 12 16:24:14 IDS snort: FrontPage Admin PWD Scan: 200.245.108.150:3275 -> x.y.z.11:80

Existence: Someone claiming to be from 200.245.108.150 (ita11149.itanet.com.br) is sending hostile traffic to our main web server.

GCIA Practical Exam

Rinaldo Ribeiro

1/17/2005

History: No previous attempt is available from this host in our firewall or IDS logs.

Intent: Find FrontPage server extensions installed in our main web server and access it trough “FP client”.

Technique:. Someone is trying to access some default files in a server with FrontPage extensions installed. All attempts are made to our main web server and looking at the source ports we could say that isn’t a busy machine.

Analysis:. Accessing a server with FP extensions installed, someone could at least modify the main web page. When someone tries to access a server with FP extensions a request is make to “/_vti_bin/shtml.dll”, what was reported by the IDS. All the others logs were generated because someone is trying to access default files like “service.pwd” , looking for a server to exploit.
The firewall is not “context based” and all this packets are interpreted as a legitimated and normal connection to the web server. The IDS alerts us. This was probably made by an automated tool or script from a ppp dial-up connection, since there is sometimes 2 seconds between 2 attempts, that is too fast for a manual process and to slow for an automated one.

Severity:

GCIA Practical Exam

Rinaldo Ribeiro

1/17/2005

Criticality	5	Main web server
Lethality	4	FrontPage extensions exploit.
System Countermeasures	4	All the servers don't have FP extensions installed.
Network Countermeasures	3	This traffic is not blocked by the firewall that isn't a "context based" tool but at least an IDS can identify the attempt.
Severity Score	2	Severity = (Criticality + Lethality) – (system countermeasures + net countermeasures

Last Updated: October 18th, 2017

Upcoming Training

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 Munich December 2017	Munich, Germany	Dec 04, 2017 - Dec 09, 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
Community SANS Nashville SEC401^	Nashville, TN	Jan 08, 2018 - Jan 13, 2018	Community SANS
Las Vegas 2018 - SEC503: Intrusion Detection In-Depth	Las Vegas, NV	Jan 28, 2018 - Feb 02, 2018	vLive
SANS Las Vegas 2018	Las Vegas, NV	Jan 28, 2018 - Feb 02, 2018	Live Event
SANS London February 2018	London, United Kingdom	Feb 05, 2018 - Feb 10, 2018	Live Event
SANS Dallas 2018	Dallas, TX	Feb 19, 2018 - Feb 24, 2018	Live Event
SANS Northern VA Spring - Tysons 2018	McLean, VA	Mar 17, 2018 - Mar 24, 2018	Live Event
SANS Secure Canberra 2018	Canberra, Australia	Mar 19, 2018 - Mar 24, 2018	Live Event
SANS 2018	Orlando, FL	Apr 03, 2018 - Apr 10, 2018	Live Event
SANS Baltimore Spring 2018	Baltimore, MD	Apr 21, 2018 - Apr 28, 2018	Live Event
SANS Security West 2018	San Diego, CA	May 11, 2018 - May 18, 2018	Live Event
Community SANS Columbia SEC503	Columbia, MD	Aug 13, 2018 - Aug 18, 2018	Community SANS
SANS OnDemand	Online	Anytime	Self Paced
SANS SelfStudy	Books & MP3s Only	Anytime	Self Paced