Global Information Assurance Certification Paper

Intrusion Detection In Depth

GCIA Practical Assignment

Version 3.3

Alfred Koo

August 14, 2004

TABLE OF CONTENTS

Abstract:

This paper consisted of three parts. The first part is a study on the CISCO Intrusion Detection Technology and the principles behind it. In addition, we took a closer examination on the existing setup of the CISCO IDS in our office. We highlighted the setup environment, the potential problems and also researched the remedial solutions.

The second part consisted of analysis on three network detects. The three network detects are Shellcode NOOP X86, Proxy Scan, Bad traffic tcp port 0. For each detect, we provided the description of the attack, the attack mechanism, correlations on the attack. Finally, based upon on our analysis, we calculated the severity of the attack and provided a defensive mechanism solution.

The third part consisted of a detailed and comprehensive network analysis. We download three different types of log (Alert, scan and out of specification) from the internet for a consecutive five days period. Through a series of manipulation, we produced analysis on the top ten activities pertaining to alert and alert activities. Furthermore, analysis were also conducted on identifying the following:

- Alert traffic top talkers and ports
- Top ten source ports for alerts
- Top ten destination ports for alert traffic
- Scan traffic top talkers and ports
- Top ten source ports by scanning traffic
- Top ten addresses where the scans are going
- Top ten destination ports for scan traffic
- Out of specification traffic
- Top ten IP addresses for scan and alert traffic:

Finally, we presented a link graph to further identifying the relationship for the source and destination addresses. In addition, defense mechanism is also provided for reference.

ASSIGNMENT 1: State of Intrusion Detection Technology

CISCO Secure Intrusion Detection Technology

Intrusion attacks on the world's network infrastructure are becoming an increasingly serious problem. For the past twelve months, we have seen dozens of hacks perpetrated against high profile targets, including newspaper, telephone companies, Internet startups and even government agencies. The attack are perpetrated for a wide variety of reasons, including fraud, espionage, sabotage, or mere curiosity. Majority of the time, these attacks can cross-global network and national boundaries, resulting in the misuse of unauthorized systems, system break-ins, equipment theft, interception of network traffic or even reconfiguration of the victim systems to enable future access.

In order to tackle with these security threats, a lot of companies or various vendors have invented security products in the market such as firewalls, access-control lists, and encryption device and authentication models.

These products, although providing a certain measure of security, contain certain limitations that may allow attackers to get pass. To keep the story short, we can summarize the pros and cons for each product in the following table:

	Features	Problem
Firewall	- Customer has to define comprehensive policy with authorized traffic flows and services that will enforce by the firewall	- It requires that the customer know the risks involved before authorizing services to pass through the firewall. - Firewall typically are used as access control devices on a network and could impact the throughput on a communication link.
Encryption	- Provide point-to-point confidentiality of data. Every packet between the end points is encrypted or encrypted by session	- The point to point encryption does not protect the hard disk and web site's data. Attacker can find vulnerability in the web server that would allow web access.
Authentication	- A widely used schemes by requiring a user to log on a machine on the network with an appropriate password.	- Cannot protect against users creating weak or easily guessed password. Some attacks can even bypass the authentication safeguard and install services on a system that give them future uncontested access.
Access-Control List	- Set of rules that router and firewalls use to permit or deny certain traffic, they are widely used in router or firewall interfaces	- No one has a clear picture of ongoing policy violations, except the system administrator instruct the router to log any instances in which traffic is denied.

In order to tackle those weakness, Intrusion detection technology is widely used as a complimentary tool alongside with the traditional security products. This paper provides a technical overview on the CISCO Secure Intrusion Detection System including the components, capabilities, architecture as well as the applications on the actual use.

Initially, we will provide a general introduction for each of the component in CISCO Secure Intrusion Detection System. Afterwards, we will provide detailed examination on each component together with the roles and responsibilities for each component in the networking environment. Finally, we will provide some cases studies during the implementation of the CISCO Secure IDS in our office together with some work around solutions.

Introduction to the components of the CISCO Secure Intrusion Detection System

The CISCO Secure IDS is a real-time Intrusion Detection Security system that includes the following components:

- CISCO Secure IDS Sensor
- CISCO Secure IDS Director
- CISCO Secure Integrated Software
- Access Control lists

CISCO Secure Intrusion Detection Sensor

The sensor is a network appliance that that is installed and maintained on a network. It provides real-time intrusion detection monitoring of network packets and syslog data from CISCO routers, it also captures and analyzes network packets (on both single and multiple packets. The IDS Sensor will capture network packets with one of its interface, and then it will reassembles and compares the data against a rule set that is predefined by the user. One of the strength of the CISCO IDS Sensor is the capability for scanning nearly every packet on a network segment. The IDS Sensor is actually examined the patterns of misuse by checking the data portion or the header portion of the network packets. The misuse patterns can be as simple as an attempt to access a specific port on a particular host (**Atomic attack**), or as complex as sequences of operations distributed across multiple hosts over an arbitrary period of time (**Composite attack**).

As we have previously mentioned about the examination on the data portion and header portion of the packets. According to CISCO, Content-based attacks are derived from data portion while context-based attacks are derived from the header portion. The following table provides a highlight on each attack:

	Pattern	
Attack	**Atomic**	**Composite**
Context(header)	Ping of death	Port sweep
	Finger	SYN attack
		TCP hijacking
Content(data)	MS IE attack	Telnet attack
	E-mails attacks	

- Our company selected the CISCO IDS router and firewall sensor with network 4250 sensor (for Ethernet segment).

When we designed our IDS sensor system in our Company's network, we divided the detection capability into the following three categories:

- Names attacks – This is the attack that have specific names or common identities, e.g. Smurf attack, PHF or Land attack.
- General Category attack – This attack keeps on appearing in new variations but with same basic methodology, e.g. Impossible IP packet or IP fragmentation.
- Extraordinary attacks – Extremely complicated or multi-faceted attacks such as TCP hijacking or E-mail spam.

One interesting thing we observed from the IDS Sensor is the false positive alarms, due to the scale and size of our network, we need to ping some of the devices from time to time on a particular interface address. However, due to the sensitivity of the CISCO IDS Sensor, we have a lot of alarms on the Director. Interesting thing is we did some fine-tuning on the sensor to ignore the specific alarm from that particular address, but the CISCO IDS still could recognize another alarms from that address.

CISCO Secure Intrusion Detection Director

The Director monitors and manages IDS Sensors, collects and analyzes network security data, downloads new attack signatures, and facilitates user operation. One distinct feature of CISCO Detection Director is without the on board reporting. In order to provide better critical attack information, the Director will export all the data to the third party database, and reporting system. As a result, no reporting is provided by the Director.

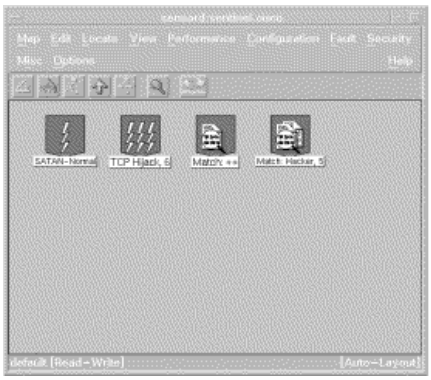
The CISCO Secure Intrusion Detection Director software running on HP OpenView application-level login mechanism, and is protected by standard UNIX password mechanisms. It provides a centralized graphical interface for the management of security across the network.

Typical functions are:

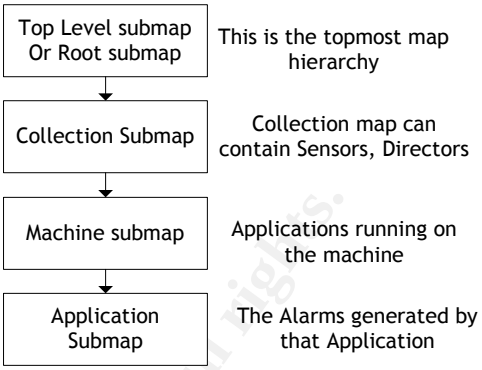
- Access to network security database (NSDB) – The director uses a map-based, color icon (**GREEN** is normal, **YELLOW** is Marginal, **RED** is Critical) method of displaying alarms. The user can actually drill down and pinpoint the exact alarm and analyze its threat potential.

- Remote monitoring of sensors – The Director displays real-time security information sent to it by the Sensors. The information is presented via icons drawn on one or more network security maps. The Director arranges icons into hierarchical security maps based on the Network Node Management (NNM) user interface of HP Openview. There are four levels for the submap, users can double-click on an icon to view the next lower "Submap" in the hierarchy.

- For simplicity, we only indicated the bottom layer of the submap hierarchy which contains Alarm and Error icons.



Bottom layer of the submap



Four level submap diagram

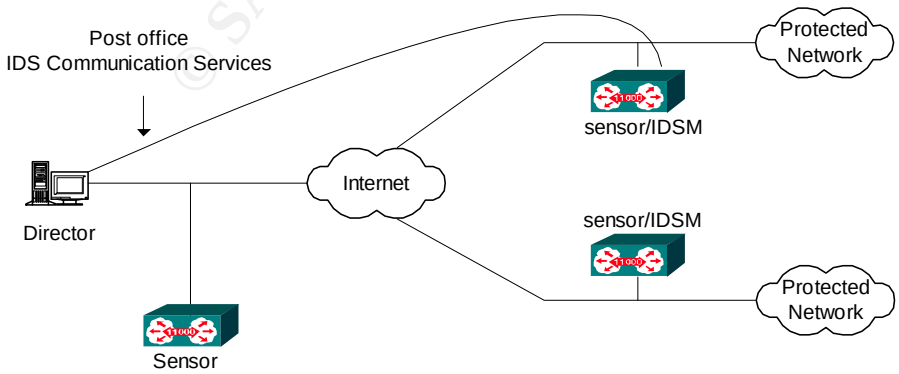
- Send event information to a trouble-ticketing system, pager or e-mail to alert security personnel when security events occur.

For our company's IDS Director, it is installed on the following platform:

- Sun Enterprise Sunfire[tm] V120 server
- 650Mhz Ultra Sparc Iii processor with 512K Level 2 cache
- 1G Memory and 60G swapping hard drive
- Sun Solaris 8 as the basic operating system
- HP Openview NNM V6.2

Post office

Post office is a communication architecture that provides a better, fast, reliable and efficient communications between CISCO IDS sensors and IDS Directors. The Post office also facilitates the transfer of files, such as configurations and log files, between CISCO ISD nodes. The following diagram indicated the relationship between the IDS Director and the IDS Sensor. One of the corporate standard in our office is to keep the monitoring Director machine from 9:00 am to 6:00 pm, and then transfer the control onto a central Director every evening which is located in Sydney, Australia.



CISCO Secure Integrated Software

The CISCO Integrated software is installed together with the CISCO ISD Sensor and acted as an in-line detection sensor to monitor the intranet, extranet and remote locations for any network violations.

When the packets or packet match a signature, the Integrated software will perform the following:

- Generate an alarm - Enable the sensor generate alarms, and route to one or more remote Director systems.
- Log the alarm event – All sensor log data is written to a flat files, which are either event logs or IP session logs.
- Drop - Dropping the offending packet
- Reset - Resetting the TCP connection. The software will enable the Sensor to reset individual TCP connections after an attack to eliminate the threat. Communication between all other connections continues, thereby decreasing the likelihood of denial-of-service problem that occurs

The following Secure Integrated software is employed by our company:

- CISCO Secure Integrated software H323 and RISP (Real Time Streaming Protocol) Inspection.

Access Control Lists (ACL)

The access control lists permit or deny passage of data packets through physical interface ports. Each programmed ACL contains permit and deny conditions that apply to IP addresses or types of traffic.

Actual implementation of the CISCO IDS

Due to the size and complexity of our company's network, we have tried to implement the IDS sensor in our own network environment, and the following aspects are taken into consideration:

- Sizing of the network nodes and the corresponding network traffic
- The location of the critical resources such as File server, and hosts on the network
- The interconnection between company network and other internet and intranet network

Based on the above-mentioned traits, we have placed our sensors in the following locations, and tried to figure the most appropriate way to protect our network and assets.

Location 1:
Placing the sensor in front of the firewall. Our primary goal is to monitor all incoming and outgoing network traffic.

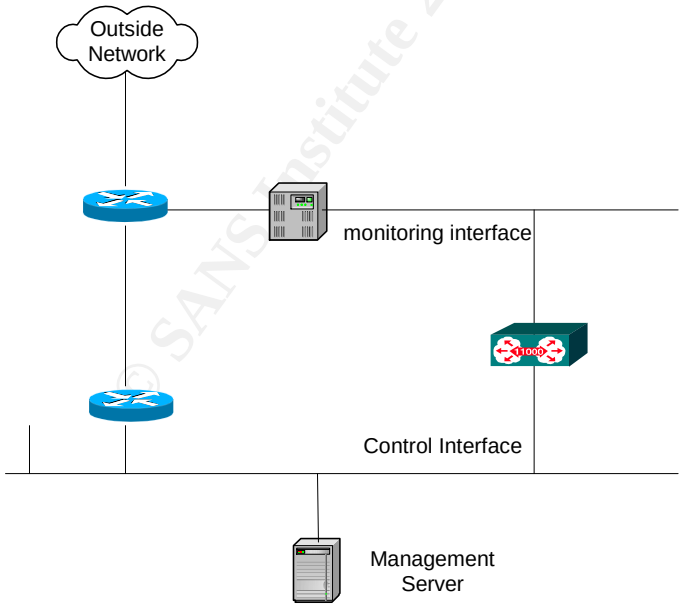
Problems:
The sensor will not normally detect the traffic that is internal to the network, as this traffic is behind the firewall. An inside attacker will take the advantage of the vulnerabilities in network services. The external sensor will not detect it probably.

Location 2:
Placing the sensor behind the firewall

Problems:
This will shield the sensor from any policy violations that the firewall rejects. For instances, if the firewall is configured to deny any passage of ping sweeps, then the sensor will not detect this type of activity or generate any alarms.

Solutions:
After having some researches on the CISCO IDS, we have decided to take advantage of the sensor's two interfaces:

We have placed the first interface (Control Interface) to communicate with the CISCO Director or router through the firewall, and the second sensor interface (Monitoring interface) directly in front of the firewall with promiscuous mode without an IP address.
The whole setup is exemplified as follows:



Further deployment

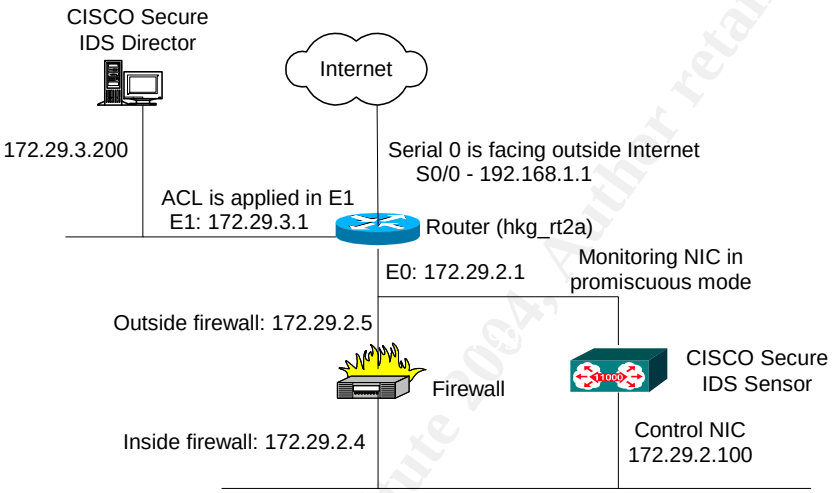
After we have investigated the location of the sensor, we have decided to further deploy our global standard solutions with a router, firewall, together with a sensor in such a way that the Sensor can dynamically update the router's ACLs to spurn attackers.

Equipment used in our setup

The following equipment are going to use in this deployment, they are indicated as follows:

- CISCO 3640 router running CISCO IOS software 12.0.12
- CISCO PIX 515S Software
- CISCO Secure Intrusion Detection Director and Sensor (detailed specification mentioned above)

The whole network setup is highlighted as follows:



Step 1: Initializing Director and Sensor

For integrity purposes, we have made up the IP addresses and the organization Name

Configuration	IDS Director	IDS Sensor
IP address	172.29.3.200	172.29.2.100
Host ID	200	100
Host Name	Director	Sensor
Organization ID	300	300
Organization Name	abccorp	Abccorp

Step 2: Installation of IDS Director and Sensor

Complete the installation of the Sun Solaris OS, HP Openview and the CISCO IDS Director (we will skip the step by step configuration of the UNIX)

Step 3: Verified the configuration of Director and Sensor

Verify the configuration in Director's /usr/nr/etc/hosts file

$ more /user/nr/etc/hosts (to display the contents in the hosts file)

200.300 localhost
200.300 director.abcorp
100.300 sensor.abcorp

Verify the configuration in Director's /usr/nr/etc/routes file

$ more /usr/nr/etc/routes (to display the contents in the routes file)

sensor.abcorp 1 172.29.2.100 45000 1

Verify the configuration in Sensor's /usr/nr/etc/hosts file

 # more /usr/nr/etc/hosts (to display the contents in the hosts file)

100.300 localhost
100.300 sensor.abcorp
200.300 director.abcorp

Verify the configuration in Sensor's /usr/nr/etc/routes file

$ more /usr/nr/etc/routes (to display the contents in the routes file)

director.abcorp 1 172.29.3.200 45000 1

Step 4: Configure the firewall

We have to configure the PIX Firewall to allow the following traffic to pass through :

- Enable the telnet traffic from the sensor's control interface (172.29.2.100) to the router (172.29.2.1)
- CISCO Secure IDS Communication (UDP Port 45000) between the sensor and the director (interface 172.29.3.200)

Step 5: Configuring the router

```
Using 1023 out of 29688 bytes
!
version 12.0.12
service timestamps debug uptime
service timestamps log uptime
service password-encryption
!
hostname hkg_rt2a
!
logging console
enable password cisco
!
```

```
memory-size iomem 10
!
interface Ethernet 0
ip address 172.29.2.1 255.255.255.128
!
interface Ethernet 1
access-group 1 out
ip address 172.29.3.1 255.255.255.128
!
interface serial0/0
ip address 192.168.1.1 255.255.0.0
!
access-list 1 permit 172.29.0.0 0.0.255.255
```

Note: An ACL (access control list) to deny all traffic outside the 172.29.0.0 network, this is to protect the CISCO Secure Intrusion Detection Director from any possible attacker outside. The ACL is applied to the out direction of the Ethernet 1 interface of the router. (Changes are highlighted in RED)

Step 6: Sensor Setup

We are going to set up the Sensor to dynamically reconfigure the filters and access control lists on the router to shun an attacker. Shunning is refer to the ability of the sensor to use a network device to deny entry to a specify network host. We have been very careful in setting up the Shunning rules in the Sensor.

There are about twenty steps to step up the Sensor's device management, we are going to skip the step by step illustration of the due to the tedious step. The final entries in the Sensor's /usr/nr/etc/managed.conf file are as follows:

```
$ more /usr/nr/etc/managed.conf

FilenameOfError                 ……/var/errors.managed

Netdvice        172.29.2.1    DefaultCISCO              password      enable
```

Note: For this scenario, the Sensor is never shunning the below addresses.

```
NeverShunAddress 172.29.3.200 255.255.255.128      #Director
NeverShunAddress 172.29.2.100 255.255.255.128      #Sensor
NeverShunAddress 172.29.2.5 255.255.255.128        #Firewall—outer interface
NeverShunAddress 172.29.2.4 255.255.255.128        #Firewall—inner interface
NeverShunAddress 172.29.2.1 255.255.255.128        #router
```

Note: For this scenario, the Sensor communicates with the 172.29.2.1 interface, and dynamically shuns on the 192.168.1.1 interface which is facing outside the Internet.

ShunInterfaceCisco 192.168.1.1 serial0/0 in
ShunAclCISCO 200
MaxShunEntries 150

Note: The Sensor will use this ACL on the router to dynamically shun attackers. The Sensor will also use the ACL 199 as the second ACL to write to whenever changes need to be made to the original ACL.

FilenameOfError ../var/errors.managed
FilenameOfConfig ../etc/managed.conf
EventLevelOfErrors 1
EventLevelOfCommandLogs 1
EnableACLLogging 0

Now the CISCO Intrusion Detection Sensor is now ready to initiate shunning by sending a dynamic ACL to the router's serial0 interface.

Step 7 : Director Setup

The above setup is now ready to initiate shunning by writing a dynamic ACL to the serial 0 interface of the router. What next is to decide which signatures will trigger a shun response. This type of automated response by the Sensor should only be configured for attack signatures with a low probability of false positive detection. In case of any suspicious activity that does not trigger the automatic shunning, we can now use the Director's function to shun manually.

For our setup, we need to configure the Security menu under the Director interface menu, and manually set the General Signatures to Shun the following signatures:

- 3250 TCP Hijacking
- 3500 rlogin – froot
- 3600 IOS DoS
- 4001 UDP Port scan
- 4053 Back Orifice
- 6001 Normal SATAN Probe
- 6002 Heavy SATAN Probe

During the installation the IDS Sensor, CISCO has already identified some of the signatures which are considered to have a very low incidence of false positives; In other words, these signatures fire only on actual attack.

We have used the following filter to extract the signatures from the packetd.conf file under the /usr/nr/etc/wgc/templates directory.

Grep SigOfGeneral /usr/nr/etc/wgc/templates/packetd.conf | awk '(if (($4 - /5/) && ($5 - /5/) && ($6 - /5/) && (&7 - /5/)) {print $0})'

The following signatures are identified:

- # IP options – Loose source route
- # IP options – Strict source route
- # Impossible IP packet
- # UDP Port scan
- # IP fragment overlap
- # ICMP network sweep w/Timestamp
- # ICMP network sweep w/Address Mask
- # ICMP Ping of Death
- # IOS DoS
- # TCP Frag SYN port sweep
- # Back Orifice
- # Heavy SATAN Probe
- # RPC Port registration
- # RPC Mounted Port sweep
- # Ident Buffer Overflow

……<Skipped>

Observations

We have placed the above mentioned set up in a testing period for three days, and the both the data are stored in the */usr/nr/var* directory for both the Sensor and the Director, and the files are named as log.YYYYMMDDHHMM.

For our environment, we have set our active logs to a maximum size of 300KB, or after 240 minutes have elapsed, whichever comes first. Archived log files are stored in /usr/nr/var/new directory.

There are five levels of log for CISCO intrusive detection. Level 1 is more severe and will forward to the Sensor, and Level 2 to 4 are forwarded to the log file on the Director. We will give out more explanation later on.

Sample Event Log

The Sample event log file are highlighted as follow, due to the size of the log, we have only extracted the first thirty lines for reference only:

4,1028109,11/10/2002,22:07:00,11/22/2002,10008,100,200,OUT,IN,5,3600,51304,ICMP,31.215.210.2,172.29.2.5,1034,21,0.0.0.0,hac
ked in,75736 ? E0D0A,
4,1028110,11/10/2002,22:07:01,11/22/2002,10008,100,200,OUT,IN,5,3600,51304,ICMP,31.215.210.2,172.29.2.5,1034,21,0.0.0.0,hac
ked in,75736 ? E0D0A,
4,1028111,11/10/2002,22:07:02,11/22/2002,10008,100,200,OUT,IN,5,3600,51304,ICMP,31.215.210.2,172.29.2.5,1034,21,0.0.0.0,hac
ked in,75736 ? E0D0A,
4,1028112,11/10/2002,22:07:03,11/22/2002,10008,100,200,OUT,IN,5,3600,51304,ICMP,31.215.210.2,172.29.2.5,1034,21,0.0.0.0,hac
ked in,75736 ? E0D0A,
4,1028113,11/10/2002,22:07:05,11/22/2002,10008,100,200,OUT,IN,5,3600,51304,ICMP,31.215.210.2,172.29.2.5,1034,21,0.0.0.0,hac
ked in,75736 ? E0D0A,
4,1028114,11/10/2002,22:07:07,11/22/2002,10008,100,200,OUT,IN,5,3600,51304,ICMP,31.215.210.2,172.29.2.5,1034,21,0.0.0.0,hac
ked in,75736 ? E0D0A,
4,1028115,11/10/2002,22:07:08,11/22/2002,10008,100,200,OUT,IN,5,3600,51304,ICMP,31.215.210.2,172.29.2.5,1034,21,0.0.0.0,hac
ked in,75736 ? E0D0A,
4,1028116,11/10/2002,22:07:13,11/22/2002,10008,100,200,OUT,IN,5,3600,51304,ICMP,31.215.210.2,172.29.2.5,1034,21,0.0.0.0,hac
ked in,75736 ? E0D0A,
4,1028117,11/10/2002,22:07:14,11/22/2002,10008,100,200,OUT,IN,5,3600,51304,ICMP,31.215.210.2,172.29.2.5,1034,21,0.0.0.0,hac
ked in,75736 ? E0D0A,
4,1028118,11/10/2002,22:07:17,11/22/2002,10008,100,200,OUT,IN,5,3600,51304,ICMP,31.215.210.2,172.29.2.5,1034,21,0.0.0.0,hac
ked in,75736 ? E0D0A,
4,1028119,11/10/2002,22:07:20,11/22/2002,10008,100,200,OUT,IN,5,3600,51304,ICMP,31.215.210.2,172.29.2.5,1034,21,0.0.0.0,hac
ked in,75736 ? E0D0A,
4,1028120,11/10/2002,22:07:22,11/22/2002,10008,100,200,OUT,IN,5,3600,51304,ICMP,31.215.210.2,172.29.2.5,1034,21,0.0.0.0,hac
ked in,75736 ? E0D0A,
4,1028121,11/10/2002,22:07:25,11/22/2002,10008,100,200,OUT,IN,5,3600,51304,ICMP,31.215.210.2,172.29.2.5,1034,21,0.0.0.0,hac
ked in,75736 ? E0D0A,
5,1028122,11/10/2002,22:08:03,11/22/2002,10008,100,200,OUT,IN,5,4001,51304,UDP,31.215.210.2,172.29.2.5,53,987,0.0.0.0,hacke
d in,75736 ? E0D0A,
5,1028123,11/10/2002,22:08:12,11/22/2002,10008,100,200,OUT,IN,5,4001,51304,UDP,31.215.210.2,172.29.2.5,53,987,0.0.0.0,hacke
d in,75736 ? E0D0A,
5,1028124,11/10/2002,22:08:13,11/22/2002,10008,100,200,OUT,IN,5,4001,51304,UDP,31.215.210.2,172.29.2.5,53,987,0.0.0.0,hacke
d in,75736 ? E0D0A
5,1028125,11/10/2002,22:08:15,11/22/2002,10008,100,200,OUT,IN,5,4001,51304,UDP,31.215.210.2,172.29.2.5,53,987,0.0.0.0,hacke
d in,75736 ? E0D0A
5,1028126,11/10/2002,22:08:17,11/22/2002,10008,100,200,OUT,IN,5,4001,51304,UDP,31.215.210.2,172.29.2.5,53,987,0.0.0.0,hacke
d in,75736 ? E0D0A
5,1028127,11/10/2002,22:08:22,11/22/2002,10008,100,200,OUT,IN,5,4001,51304,UDP,31.215.210.2,172.29.2.5,53,987,0.0.0.0,hacke
d in,75736 ? E0D0A
5,1028128,11/10/2002,22:08:25,11/22/2002,10008,100,200,OUT,IN,5,4001,51304,UDP,31.215.210.2,172.29.2.5,53,987,0.0.0.0,hacke
d in,75736 ? E0D0A
5,1028129,11/10/2002,22:08:27,11/22/2002,10008,100,200,OUT,IN,5,4001,51304,UDP,31.215.210.2,172.29.2.5,53,988,0.0.0.0,hacke
d in,75736 ? E0D0A
5,1028130,11/10/2002,22:08:29,11/22/2002,10008,100,200,OUT,IN,5,4001,51304,UDP,31.215.210.2,172.29.2.5,53,988,0.0.0.0,hacke
d in,75736 ? E0D0A
5,1028131,11/10/2002,22:08:33,11/22/2002,10008,100,200,OUT,IN,5,4001,51304,UDP,31.215.210.2,172.29.2.5,53,988,0.0.0.0,hacke
d in,75736 ? E0D0A
5,1028132,11/10/2002,22:08:35,11/22/2002,10008,100,200,OUT,IN,5,4001,51304,UDP,31.215.210.2,172.29.2.5,53,988,0.0.0.0,hacke
d in,75736 ? E0D0A
5,1028133,11/10/2002,22:08:38,11/22/2002,10008,100,200,OUT,IN,5,4001,51304,UDP,31.215.210.2,172.29.2.5,53,988,0.0.0.0,hacke
d in,75736 ? E0D0A
5,1028134,11/10/2002,22:08:44,11/22/2002,10008,100,200,OUT,IN,5,4001,51304,UDP,31.215.210.2,172.29.2.5,53,988,0.0.0.0,hacke
d in,75736 ? E0D0A
5,1028135,11/10/2002,22:08::47,11/22/2002,10008,100,200,OUT,IN,5,4001,51304,UDP,31.215.210.2,172.29.2.5,53,989,0.0.0.0,hack
ed in,75736 ? E0D0A
5,1028136,11/10/2002,22:08:55,11/22/2002,10008,100,200,OUT,IN,5,4001,51304,UDP,31.215.210.2,172.29.2.5,53,989,0.0.0.0,hacke
d in,75736 ? E0D0A

EXCEL Table

The following files are extracted into EXCEL format and summarize in the following table:

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
4	1028109	11/10/2002	22:07:00	11/22/2002	10008	100	200	OUT	IN	5	3600	51304	ICMP	31.215.210.2	172.29.2.5	1034	21	0.0.0.0	hacked in	75736 ? E0D0A
4	1028110	11/10/2002	22:07:01	11/22/2002	10008	100	200	OUT	IN	5	3600	51304	ICMP	31.215.210.2	172.29.2.5	1034	21	0.0.0.0	hacked in	75736 ? E0D0A
4	1028111	11/10/2002	22:07:02	11/22/2002	10008	100	200	OUT	IN	5	3600	51304	ICMP	31.215.210.2	172.29.2.5	1034	21	0.0.0.0	hacked in	75736 ? E0D0A
4	1028112	11/10/2002	22:07:03	11/22/2002	10008	100	200	OUT	IN	5	3600	51304	ICMP	31.215.210.2	172.29.2.5	1034	21	0.0.0.0	hacked in	75736 ? E0D0A
4	1028113	11/10/2002	22:07:05	11/22/2002	10008	100	200	OUT	IN	5	3600	51304	ICMP	31.215.210.2	172.29.2.5	1034	21	0.0.0.0	hacked in	75736 ? E0D0A
4	1028114	11/10/2002	22:07:07	11/22/2002	10008	100	200	OUT	IN	5	3600	51304	ICMP	31.215.210.2	172.29.2.5	1034	21	0.0.0.0	hacked in	75736 ? E0D0A
4	1028115	11/10/2002	22:07:08	11/22/2002	10008	100	200	OUT	IN	5	3600	51304	ICMP	31.215.210.2	172.29.2.5	1034	21	0.0.0.0	hacked in	75736 ? E0D0A
4	1028116	11/10/2002	22:07:13	11/22/2002	10008	100	200	OUT	IN	5	3600	51304	ICMP	31.215.210.2	172.29.2.5	1034	21	0.0.0.0	hacked in	75736 ? E0D0A
4	1028117	11/10/2002	22:07:14	11/22/2002	10008	100	200	OUT	IN	5	3600	51304	ICMP	31.215.210.2	172.29.2.5	1034	21	0.0.0.0	hacked in	75736 ? E0D0A
4	1028118	11/10/2002	22:07:17	11/22/2002	10008	100	200	OUT	IN	5	3600	51304	ICMP	31.215.210.2	172.29.2.5	1034	21	0.0.0.0	hacked in	75736 ? E0D0A
4	1028119	11/10/2002	22:07:20	11/22/2002	10008	100	200	OUT	IN	5	3600	51304	ICMP	31.215.210.2	172.29.2.5	1034	21	0.0.0.0	hacked in	75736 ? E0D0A
4	1028120	11/10/2002	22:07:22	11/22/2002	10008	100	200	OUT	IN	5	3600	51304	ICMP	31.215.210.2	172.29.2.5	1034	21	0.0.0.0	hacked in	75736 ? E0D0A
4	1028121	11/10/2002	22:07:25	11/22/2002	10008	100	200	OUT	IN	5	3600	51304	ICMP	31.215.210.2	172.29.2.5	1034	21	0.0.0.0	hacked in	75736 ? E0D0A
5	1028122	11/10/2002	22:08:03	11/22/2002	10008	100	200	OUT	IN	5	4001	51304	UDP	31.215.210.2	172.29.2.5	53	987	0.0.0.0	hacked in	75736 ? E0D0A
5	1028123	11/10/2002	22:08:12	11/22/2002	10008	100	200	OUT	IN	5	4001	51304	UDP	31.215.210.2	172.29.2.5	53	987	0.0.0.0	hacked in	75736 ? E0D0A
5	1028124	11/10/2002	22:08:13	11/22/2002	10008	100	200	OUT	IN	5	4001	51304	UDP	31.215.210.2	172.29.2.5	53	987	0.0.0.0	hacked in	75736 ? E0D0A
5	1028125	11/10/2002	22:08:15	11/22/2002	10008	100	200	OUT	IN	5	4001	51304	UDP	31.215.210.2	172.29.2.5	53	987	0.0.0.0	hacked in	75736 ? E0D0A
5	1028126	11/10/2002	22:08:17	11/22/2002	10008	100	200	OUT	IN	5	4001	51304	UDP	31.215.210.2	172.29.2.5	53	987	0.0.0.0	hacked in	75736 ? E0D0A
5	1028127	11/10/2002	22:08:22	11/22/2002	10008	100	200	OUT	IN	5	4001	51304	UDP	31.215.210.2	172.29.2.5	53	987	0.0.0.0	hacked in	75736 ? E0D0A
5	1028128	11/10/2002	22:08:25	11/22/2002	10008	100	200	OUT	IN	5	4001	51304	UDP	31.215.210.2	172.29.2.5	53	987	0.0.0.0	hacked in	75736 ? E0D0A
5	1028129	11/10/2002	22:08:27	11/22/2002	10008	100	200	OUT	IN	5	4001	51304	UDP	31.215.210.2	172.29.2.5	53	988	0.0.0.0	hacked in	75736 ? E0D0A
5	1028130	11/10/2002	22:08:29	11/22/2002	10008	100	200	OUT	IN	5	4001	51304	UDP	31.215.210.2	172.29.2.5	53	988	0.0.0.0	hacked in	75736 ? E0D0A
5	1028131	11/10/2002	22:08:33	11/22/2002	10008	100	200	OUT	IN	5	4001	51304	UDP	31.215.210.2	172.29.2.5	53	988	0.0.0.0	hacked in	75736 ? E0D0A
5	1028132	11/10/2002	22:08:35	11/22/2002	10008	100	200	OUT	IN	5	4001	51304	UDP	31.215.210.2	172.29.2.5	53	988	0.0.0.0	hacked in	75736 ? E0D0A
5	1028133	11/10/2002	22:08:38	11/22/2002	10008	100	200	OUT	IN	5	4001	51304	UDP	31.215.210.2	172.29.2.5	53	988	0.0.0.0	hacked in	75736 ? E0D0A
5	1028134	11/10/2002	22:08:44	11/22/2002	10008	100	200	OUT	IN	5	4001	51304	UDP	31.215.210.2	172.29.2.5	53	988	0.0.0.0	hacked in	75736 ? E0D0A
5	1028135	11/10/2002	22:08:47	11/22/2002	10008	100	200	OUT	IN	5	4001	51304	UDP	31.215.210.2	172.29.2.5	53	989	0.0.0.0	hacked in	75736 ? E0D0A
5	1028136	11/10/2002	22:08:55	11/22/2002	10008	100	200	OUT	IN	5	4001	51304	UDP	31.215.210.2	172.29.2.5	53	989	0.0.0.0	hacked in	75736 ? E0D0A
5	1028137	11/10/2002	22:09:02	11/22/2002	10008	100	200	OUT	IN	5	4001	51304	UDP	31.215.210.2	172.29.2.5	53	989	0.0.0.0	hacked in	75736 ? E0D0A
5	1028138	11/10/2002	22:09:12	11/22/2002	10008	100	200	OUT	IN	5	4001	51304	UDP	31.215.210.2	172.29.2.5	53	989	0.0.0.0	hacked in	75736 ? E0D0A

Explanation on the columns

Column	Field	Comments
A	Record type	The numbers correspond to the logging level: 3 is error, 3 is command, 4 is events or alarms, and 5 is IP
B	Record Number	Begin at 1,000,000. Each time the sensord/packetd process is started, the record number is incremented by one.
C	Greenwich Mean Time(GMT) Date	This is the Greenwich Time
D	Local Time	This is the local time of the sensor
E	Local Date	This is the local date of the sensor
F	Application ID of the process that generated the log record. 10008 is the packetd service	The /usr/nr/etc/services file on both the Directors and Sensors provides the mapping of application Ids to CISCO Secure IDS services.
G	Host ID of the Sensor or the Director that generated the log record	The host ID was assigned by the us during system initialization (that is, through sysconfig-sensor). A mapping of the host ID to host names is provided in the /usr/nr/etc/hosts file on both the Directors and Sensors.
H	Organization ID of the Sensor or Director that generated the log record	The organization ID was assigned by the users
I	Event source location	The location is located outside the defined protected network
J	Event distinction location	The location is located inside the defined protected network.
K	Alarm Level	CISCO Secure IDS has five alarms level. Level 1 and higher events are logged on the sensor, and Level 2-5 events are forwarded to the log file on the director.
L	Signature ID	A mapping of signature Ids to signature names is provided in the /usr/nr/etc/signatures file on the Directors and the Sensors. Signature Ids range from 1000 to 10000.
M	Sub-signature ID	This ID is primarily found with string match signatures, which are user-customizable. String match sub-signature names can be found in the optional event detail field. In this example, the string “hacked in”

		triggered the logging event. For signatures that do not have sub-signatures, this field will be “0”. Sub-signature Ids are assigned by the system and user-defined string match signatures start with sub-signature ID 51304
N	Indicated IP traffic	ICMP and UDP are the traffic supported at this time
O	Source of the ID address that triggered the event	N/A
P	Destination of the ID address of the triggered event	N/A
Q	Source Port	N/A
R	Destination Port	N/A
S	External Data source IP address	IP address of the network device that detected the event when external data sources are used (for example, ACL syslog from a router) 0.0.0.0 signifies that the Sensor specified by the recorded host and organization ID detected this event.
T	Optional event detail	This field is not always populated. In this example, the string “hacked in” triggered the logging of this event.
U	Optional context data	This field is not always populated. In this example, it provides a snapshot of incoming and outgoing binary TCP traffic up to a maximum of 256

Conclusion

In closing, this paper demonstrated a comprehensive method to combat unauthorized intrusions. The Sensor detected the IOS DOS and UDP Port scan and refrained them for entering into our network. The main advantage we have observed from this implementation is we don't have to update the intrusion signature on a regular basis, the Sensor will dynamically update the router ACL's to spurn attackers from outside.

However, two problems were reported during the implementation of the system. We have listed the problems as follows together with the solutions:

The router performance has immediate impact once the CISCO Secure IDS Dynamic ACL is imposed. Some packets were dropped at inner interface of the router. The remedial solution is to add additional flash memory and DRAM memory for the router, the performance is greatly improved.

The second phenomenon is the CISCO IDS Dynamic ACL will replace the traditional ACLs already in place on the inner interface (172.29.2.4). The original inner interface facing the firewall is used to shield some of the traffic passing into the internal network. Our solution is to implement the ACL in the Firewall's interface that is facing the router(172.29.2.5). Then it will fix the problem for the ACL.

In addition, many security administrators prefer to have a dual-home Director machine so that the Sensor can communicate with it directly instead of going through the PIX firewall and the router. But due to the Corporate Standard of our company, we need to pass through the firewall and the router.

The only drawback we observed from the CISCO IDS Sensor and Director are both running on SUN Solaris platform, and it will be nice if both the Sensor and Director can run on the Windows platform for easy management purposes.

References

CISCO Systems, “Intrusion Detection Planning Guide”
URL:http://www.cisco.com/univercd/cc/td/doc/product/iaabu/idpg/ (Jan 2, 2003)

Michael Wilkison, “How to evaluating Network Intrusion Detection Systems?”
URL:http://www.sans.org/resources/idfaq/eval_ids.php (Oct 8, 2002)

Betsy Yocom, Kevin Brown and Dan Van Derveer, “CISCO offers wide speed Intrusion”
URL:http://www.nwfusion.com/reviews/2000/1218rev2.html (Dec 18, 2000)

Global Network Technology Services, “Intrusion Detection Case Studies”
URL:http://www.globalnts.com/ids.cfm (March, 2001)

Julia Allen and Alan Christie, “State of the Practice of Intrusion Detection Technologies”
URL:http://www.sei.cmu.edu/publications/documents/99.reports/99tr028/99tr028abstract.html (January 2001)

CISCO Systems, ”CISCO Secure Scanner Overview”
URL:http://www.cisco.com/en/US/customer/products/sw/secursw/ps2134/products_user_guide_chapter09186a008007d881.html (Dec 2001)

Greg Shipley, “Intrusion Detection, Take two”
http://www.nwc.com/1023/1023f1report2.html (November 15, 1999)

ASSIGNMENT 2 : Three network detects

Detect #1 – SHELLCODE x86 NOOP (False Alarm)

Source of trace:

The trace was downloaded from the incidents.org RAW log file, and the name of the file is 2002.10.15. The downloaded date code is Dec 2, 2002. The complete URL is http://www.incidents.org/logs/Raw/2002.10.15.

The binary tcpdump log file was then analyzed by snort using the following command:

Snort –r d:\snort\ 2002.10.15 –c snort.conf –l c:\mylog

Event Traces:

The following is Snort "Alert" data:

```
[**] [1:648:5] SHELLCODE x86 NOOP [**]
[Classification: Executable code was detected] [Priority: 1]
11/15-10:55:37.866507 129.118.2.10:57425 -> 170.129.50.120:63414
TCP TTL:51 TOS:0xA0 ID:46576 IpLen:20 DgmLen:1420 DF
***AP*** Seq: 0xA076B1D8  Ack: 0x90CF9E29  Win: 0x16D0  TcpLen: 20
[Xref => arachnids 181]

[**] [1:648:5] SHELLCODE x86 NOOP [**]
[Classification: Executable code was detected] [Priority: 1]
11/15-10:55:37.876507 129.118.2.10:57425 -> 170.129.50.120:63414
TCP TTL:51 TOS:0xA0 ID:46577 IpLen:20 DgmLen:1420 DF
***A**** Seq: 0xA076B73C  Ack: 0x90CF9E29  Win: 0x16D0  TcpLen: 20
[Xref => arachnids 181]

[**] [1:648:5] SHELLCODE x86 NOOP [**]
[Classification: Executable code was detected] [Priority: 1]
11/15-10:55:38.016507 129.118.2.10:57425 -> 170.129.50.120:63414
TCP TTL:51 TOS:0xA0 ID:46584 IpLen:20 DgmLen:1420 DF
***A**** Seq: 0xA076DCF8  Ack: 0x90CF9E29  Win: 0x16D0  TcpLen: 20
[Xref => arachnids 181]

[**] [1:648:5] SHELLCODE x86 NOOP [**]
[Classification: Executable code was detected] [Priority: 1]
11/15-10:55:38.936507 129.118.2.10:57425 -> 170.129.50.120:63414
TCP TTL:51 TOS:0xA0 ID:46646 IpLen:20 DgmLen:1420 DF
***A**** Seq: 0xA0782B30  Ack: 0x90CF9E29  Win: 0x16D0  TcpLen: 20
[Xref => arachnids 181]

[**] [1:648:5] SHELLCODE x86 NOOP [**]
[Classification: Executable code was detected] [Priority: 1]
11/15-10:55:39.306507 129.118.2.10:57425 -> 170.129.50.120:63414
TCP TTL:51 TOS:0xA0 ID:46667 IpLen:20 DgmLen:1420 DF
***A**** Seq: 0xA0789C64  Ack: 0x90CF9E29  Win: 0x16D0  TcpLen: 20
[Xref => arachnids 181]
```

```
------- <snipped> -------
```

The following is correlating TCPdump log data:

```
11/15-10:55:36.576507 129.118.2.10:57425 -> 170.129.50.120:63414
TCP TTL:51 TOS:0xA0 ID:46491 IpLen:20 DgmLen:1420 DF
***A**** Seq: 0xA074E7A4  Ack: 0x90CF9E29  Win: 0x16D0  TcpLen: 20
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=

11/15-10:55:36.676507 129.118.2.10:57425 -> 170.129.50.120:63414
TCP TTL:51 TOS:0x0 ID:46498 IpLen:20 DgmLen:1420 DF
***A**** Seq: 0xA0750D60  Ack: 0x90CF9E29  Win: 0x16D0  TcpLen: 20
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=

11/15-10:55:36.756507 129.118.2.10:57425 -> 170.129.50.120:63414
TCP TTL:51 TOS:0xA0 ID:46504 IpLen:20 DgmLen:1420 DF
***A**** Seq: 0xA0752DB8  Ack: 0x90CF9E29  Win: 0x16D0  TcpLen: 20
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=

11/15-10:55:37.026507 129.118.2.10:57425 -> 170.129.50.120:63414
TCP TTL:51 TOS:0x0 ID:46521 IpLen:20 DgmLen:1420 DF
***A**** Seq: 0xA075895C  Ack: 0x90CF9E29  Win: 0x16D0  TcpLen: 20
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=

11/15-10:55:37.146507 129.118.2.10:57425 -> 170.129.50.120:63414
TCP TTL:51 TOS:0xA0 ID:46529 IpLen:20 DgmLen:1420 DF
***A**** Seq: 0xA075B47C  Ack: 0x90CF9E29  Win: 0x16D0  TcpLen: 20
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=

11/15-10:55:37.216507 129.118.2.10:57425 -> 170.129.50.120:63414
TCP TTL:51 TOS:0xA0 ID:46532 IpLen:20 DgmLen:1420 DF
***A**** Seq: 0xA075C4A8  Ack: 0x90CF9E29  Win: 0x16D0  TcpLen: 20
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=

11/15-10:55:37.296507 129.118.2.10:57425 -> 170.129.50.120:63414
TCP TTL:51 TOS:0xA0 ID:46538 IpLen:20 DgmLen:1420 DF
***AP*** Seq: 0xA075E500  Ack: 0x90CF9E29  Win: 0x16D0  TcpLen: 20
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=

11/15-10:55:37.526507 129.118.2.10:57425 -> 170.129.50.120:63414
TCP TTL:51 TOS:0xA0 ID:46553 IpLen:20 DgmLen:1420 DF
***A**** Seq: 0xA07635DC  Ack: 0x90CF9E29  Win: 0x16D0  TcpLen: 20
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=

11/15-10:55:37.626507 129.118.2.10:57425 -> 170.129.50.120:63414
TCP TTL:51 TOS:0xA0 ID:46560 IpLen:20 DgmLen:1420 DF
***A**** Seq: 0xA0765B98  Ack: 0x90CF9E29  Win: 0x16D0  TcpLen: 20
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=

11/15-10:55:37.856507 129.118.2.10:57425 -> 170.129.50.120:63414
TCP TTL:51 TOS:0xA0 ID:46575 IpLen:20 DgmLen:1420 DF
***A**** Seq: 0xA076AC74  Ack: 0x90CF9E29  Win: 0x16D0  TcpLen: 20
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=

------- <snipped> -------
```

Detect Generated by:

This detect was generated by Snort and TCP dump running on a Intel PIII 800 machine.

The rule that generated this alert was as follows:

alert ip $EXTERNAL_NET any -> $HOME_NET $SHELLCODE_PORTS (msg:"SHELLCODE x86 NOOP"; content: "|90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90|"; depth: 128; reference:arachnids,181; classtype:shellcode-detect; sid:648; rev:5;)

Probability the source address was spoofed:

The source address of these packets was probably not spoofed. The attacker is probing for a service that runs over TCP. The alert file is not a sign of SSH attack in particular, it is a generic alert that the IDS detected 0X90 characters in the payload of a packet that is going to ingress or egress the network.

Besides, in order to be a useful and successful scan, the attacker must receive the response packets. As we can see no similar probes for the same device from a wide variety of IP addresses, we can assume the attacker was not trying to conceal the real source IP address of the scan by mixing with other spoofed IP addresses.

Description of the attack:

As part of the attack on a remote service, the attacker is attempted to take advantage of insecure coding practices in hopes of executing arbitrary code. The procedure generally make uses of NOPs.. The attacker fills an address space with a large number of NOPs followed by some junk messages by his or her code of choice. This will allow “sledding” into the attackers shellcode.

Attack Mechanism:

The attacker will use buffer overflows as an attempt to run an attack program code on the targeted machine. If a particular program or service was written without a proper bounds checking. Possibility for writing arbitrary data to the address space is relative high. This will directly cause the program to terminate in a horrible death. The logic is If we get the return address of the program pointing to the beginning of the newly written data, we can execute the code of our choice on condition that the newly written data is executable.

The vulnerability of this attack is the intruder will calculate exact where the return address may point to, then the person will pad the space leading up to your shellcode with NOPs. Under this circumstances, if the return address points anywhere in the series of NOPs, execution will slide down into your shellcode.

From a different perspective point of view, as we can see from the tcp dump below, the attack mechanism will create the NOOP "sled" in the hex output. The value 0X90 NOOP instruction for the Intel Processor.

```
1/15-10:55:37.026507 129.118.2.10:57425 -> 170.129.50.120:63414
TCP TTL:51 TOS:0x0 ID:46521 IpLen:20 DgmLen:1420 DF
***A**** Seq: 0xA075895C  Ack: 0x90CF9E29  Win: 0x16D0  TcpLen: 20
90 90 90 90 90 90 90 90 90 90 90 90 90 90  ..............
61 72 6D 65 64 20 73 69 67 6E 61 6C 5F 61 72 72  armed signal_arr
69 76 65 64 20 25 70 2C 20 72 65 73 20 25 64 00  ived %p, res %d.
64 69 64 6E 27 74 20 73 75 73 70 65 6E 64 20 6D  didn't suspend m
61 69 6E 20 74 68 72 65 61 64 2C 20 74 68 20 25  ain thread, th %
70 00 52 65 73 75 6D 65 54 68 72 65 61 64 20 72  p.ResumeThread r
65 74 75 72 6E 65 64 20 25 64 00 72 65 74 75 72  eturned %d.retur
6E 69 6E 67 20 25 64 00 55 89 E5 81 EC FC 00 00  ning %d.U.......
00 57 56 89 C7 53 31 F6 89 95 2C FF FF FF 89 8D  .WV..S1...,.....
28 FF FF FF C6 85 27 FF FF FF 00 C7 85 20 FF FF  (.....'...... ..
FF 00 00 00 00 83 3D FC DF 08 61 00 0F 85 5D 02  ......=...a...].
00 00 89 B5 1C FF FF FF 83 C4 F4 68 38 43 08 61  ...........h8C.a
E8 17 77 04 00 83 C4 0C 83 3D 34 43 08 61 00 74  ..w......=4C.a.t
0B BE 30 43 08 61 E9 99 01 00 00 90 83 C4 F4 31  ..0C.a.........1
F6 68 38 43 08 61 E8 F9 76 04 00 A1 38 4F 08 61  .h8C.a..v...8O.a
```

In fact, many IDS systems will alarm when they see a series of 0X90 bytes in the payload.

Another noticeable aspect of the NOOP is that string "bin/sh" is clearly visible in the ASCII decode (on the right). The is part of the "shellcode" of the exploit that is a dead giveaway to indicate the attacker is attempting an exploit that launches a common-line shell. For our case, we didn't see the bin/sh, which indicated that the possibility for being attack is relative low.

Correlations:

I found a couple references that can correlate the NOOP X86 on the internet which have similar situations:

- http://www.der-keiler.de/Mailing-Lists/securityfocus/focus-ids/2002-04/0035.html
  Description on the NOOP was reported by some user, typical situation is the connection of TCP from a source address high port to destination address high port.
- http://cert.uni-stuttgart.de/archive/incidents/2001/10/msg00030.html
  Another description on the NOOP without the "bin/sh" on the hex dump, probably downloading files from a web-mail server.

Evidence of active targeting:

There was no active targeting as this is just probably a download of large executable files, or part of a executable file for optimization and alignment purpose.

Severity:

Note: The scale is ranked from 1 (lowest) to 5 (highest)

Target Criticality = 2.	The target system is probably an email server, someone is tried to down or transmit some large jpg/gif files
Attack Lethality = 0	This is probably a false alarm
System Countermeasures = 1	The computer is not likely to be patched or protected.
Network countermeasures = 2	This is probably an unprotected network, or perhaps the blocking mechanism is not strong enough.

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

Attack severity : (2+0) – (1+2) = **-1** The severity of this traffic is low

Defensive Mechanism:

The best approach is to implement a firewall together with proper intrusion detection system. Then place the mail server or file server inside the firewall. The IDS will actively monitor the high level ports. In addition, we can try to restrict the size of the attachments or files for the users.

Test Question:

What of the following is correct description for a Shellcode X86 NOOP attack ?

A. The NOOP is a denial of service attack
B. The attacker is trying to take advantage of insecure code practices in the hope of executing arbitrary code. The procedures generally make use of NOOPs
C. Shellcode is an antivirus software
D. Shellcode is a proper Intrusion Detection for RingZero.

Answer: B

Detect # 2 - Proxy SCAN

Source of trace:

The trace was downloaded from the incidents.org under the RAW log home pagefile, and the name of the file is 2002.9.30. The downloaded date code is Dec 2, 2002. The complete URL is http://www.incidents.org/logs/Raw/2002.9.30

Event Traces:

The following is Snort "Alert" data:

```
[**] [1:615:3] SCAN SOCKS Proxy attempt [**]
[Classification: Attempted Information Leak] [Priority: 2]
10/30-20:32:40.696507 216.77.216.104:61578 -> 207.166.225.208:1080
TCP TTL:49 TOS:0x0 ID:7578 IpLen:20 DgmLen:40
******S* Seq: 0x7CA9D76C  Ack: 0x7CA9D76C  Win: 0x400  TcpLen: 20
[Xref => url help.undernet.org/proxyscan/]

[**] [1:615:3] SCAN SOCKS Proxy attempt [**]
[Classification: Attempted Information Leak] [Priority: 2]
10/30-20:36:59.326507 216.77.216.104:22621 -> 207.166.17.220:1080
TCP TTL:49 TOS:0x0 ID:20037 IpLen:20 DgmLen:40
******S* Seq: 0x20CE6E48  Ack: 0x20CE6E48  Win: 0x400  TcpLen: 20
[Xref => url help.undernet.org/proxyscan/]

[**] [1:615:3] SCAN SOCKS Proxy attempt [**]
[Classification: Attempted Information Leak] [Priority: 2]
10/30-20:41:18.046507 216.77.216.104:12482 -> 207.166.108.129:1080
TCP TTL:49 TOS:0x0 ID:21282 IpLen:20 DgmLen:40
******S* Seq: 0xC99E4CD  Ack: 0xC99E4CD  Win: 0x400  TcpLen: 20
[Xref => url help.undernet.org/proxyscan/]

[**] [1:615:3] SCAN SOCKS Proxy attempt [**]
[Classification: Attempted Information Leak] [Priority: 2]
10/30-20:45:36.696507 216.77.216.104:28053 -> 207.166.82.111:1080
TCP TTL:49 TOS:0x0 ID:48650 IpLen:20 DgmLen:40
******S* Seq: 0x3E606200  Ack: 0x3E606200  Win: 0x400  TcpLen: 20
[Xref => url help.undernet.org/proxyscan/]

<snipped>

[**] [1:618:2] SCAN Squid Proxy attempt [**]
[Classification: Attempted Information Leak] [Priority: 2]
10/30-23:17:48.026507 172.184.170.160:3057 -> 207.166.49.39:3128
TCP TTL:115 TOS:0x0 ID:51431 IpLen:20 DgmLen:48 DF
******S* Seq: 0x500FECF4  Ack: 0x0  Win: 0x7FFF  TcpLen: 28
TCP Options (4) => MSS: 1360 NOP NOP SackOK

[**] [1:618:2] SCAN Squid Proxy attempt [**]
[Classification: Attempted Information Leak] [Priority: 2]
10/30-23:17:51.026507 172.184.170.160:3057 -> 207.166.49.39:3128
TCP TTL:115 TOS:0x0 ID:51535 IpLen:20 DgmLen:48 DF
******S* Seq: 0x500FECF4  Ack: 0x0  Win: 0x7FFF  TcpLen: 28
```

```
TCP Options (4) => MSS: 1360 NOP NOP SackOK

[**] [1:618:2] SCAN Squid Proxy attempt [**]
[Classification: Attempted Information Leak] [Priority: 2]
10/30-23:17:56.516507 172.184.170.160:3057 -> 207.166.49.39:3128
TCP TTL:115 TOS:0x0 ID:51737 IpLen:20 DgmLen:48 DF
******S* Seq: 0x500FECF4  Ack: 0x0  Win: 0x7FFF  TcpLen: 28

<snipped>


TCP Options (4) => MSS: 1360 NOP NOP SackOK
[**] [1:620:2] SCAN Proxy (8080) attempt [**]
[Classification: Attempted Information Leak] [Priority: 2]
10/30-23:23:57.626507 24.90.122.137:1703 -> 207.166.38.83:8080
TCP TTL:112 TOS:0x0 ID:48157 IpLen:20 DgmLen:48 DF
******S* Seq: 0xC87018C1  Ack: 0x0  Win: 0x4000  TcpLen: 28
TCP Options (4) => MSS: 1460 NOP NOP SackOK

[**] [1:620:2] SCAN Proxy (8080) attempt [**]
[Classification: Attempted Information Leak] [Priority: 2]
10/30-23:23:57.626507 24.90.122.137:1588 -> 207.166.38.60:8080
TCP TTL:112 TOS:0x0 ID:48159 IpLen:20 DgmLen:48 DF
******S* Seq: 0xC801A198  Ack: 0x0  Win: 0x4000  TcpLen: 28
TCP Options (4) => MSS: 1460 NOP NOP SackOK

[**] [1:620:2] SCAN Proxy (8080) attempt [**]
[Classification: Attempted Information Leak] [Priority: 2]
10/30-23:23:57.626507 24.90.122.137:1713 -> 207.166.38.85:8080
TCP TTL:112 TOS:0x0 ID:48162 IpLen:20 DgmLen:48 DF
******S* Seq: 0xC8779B30  Ack: 0x0  Win: 0x4000  TcpLen: 28
TCP Options (4) => MSS: 1460 NOP NOP SackOK

<snipped>
```

Detected generated by:

The binary tcpdump log file was then analyzed by snort using the following command:

Snort –r d:\snort\ 2002.10.15 –c snort.conf –l c:\mylog

The rules that generated this alert were as follows:

alert tcp $EXTERNAL_NET any -> $HOME_NET 80 (msg:"SCAN cybercop os probe"; flags: SF12; dsize: 0; reference:arachnids,146; classtype:attempted-recon; sid:619; rev:1;)
alert tcp $EXTERNAL_NET any -> $HOME_NET 3128 (msg:"SCAN Squid Proxy attempt"; flags:S; classtype:attempted-recon; sid:618; rev:2;)
alert tcp $EXTERNAL_NET any -> $HOME_NET 1080 (msg:"SCAN SOCKS Proxy attempt"; flags:S; reference:url,help.undernet.org/proxyscan/; classtype:attempted-recon; sid:615; rev:3;)
alert tcp $EXTERNAL_NET any -> $HOME_NET 8080 (msg:"SCAN Proxy \(8080\) attempt"; flags:S; classtype:attempted-recon; sid:620; rev:2;)

Probability the source address was spoofed:

The intruder is probably looking for a proxy server and some responses back to complete the hand-shaking processes. The source address is probably not being spoofed.

Interesting thing to find out that the attackers are tried to explore the 207.166.X.X range network, and listened on the 1080, 3128 and 8080 respectively. One common thing we have observed is the sources are kept on sending out SYS to the destination address, and probably waiting for a SYN/ACK back, so that the source can send out the ACK and PUSH to send data to the proxy server, but what we have observed is only SYNs and no evidence that the three way hand shaking was established.

What we have suspected is the sources addresses either 216.77.216.104 or 172.184.170.160 were infected with the virus and tried to scan proxy servers.
The source IP: 216.77.216.104 resolves out to be from a company called Bellsouth Inc which is located in Georgia, USA.

```
OrgName:    BellSouth.net Inc.
OrgID:      BELL
Address:    575 Morosgo Drive
City:       Atlanta
StateProv:  GA
PostalCode: 30324
Country:    US

NetRange:   216.76.0.0 - 216.79.255.255
CIDR:       216.76.0.0/14
NetName:    BELLSNET-BLK5
NetHandle:  NET-216-76-0-0-1
Parent:     NET-216-0-0-0-0
NetType:    Direct Allocation
NameServer: NS.BELLSOUTH.NET
NameServer: NS.ATL.BELLSOUTH.NET



AbuseHandle: ABUSE81-ARIN
AbuseName:    Abuse Group
AbusePhone:   +1-404-499-5224
AbuseEmail:   abuse@bellsouth.net

TechHandle: JG726-ARIN
TechName:   Geurin, Joe
TechPhone:  +1-404-499-5240
TechEmail:  ipadmin@bellsouth.net
```

The another source IP: 172.184.170.160 resolves out to be an ISP called America Online in Virginia, USA

```
OrgName:    America Online
OrgID:      AOL
Address:    8619 Westwood Center Drive
Address:    Suite 200
City:       Vienna
StateProv:  VA
PostalCode: 22182
Country:    US

NetRange:   172.128.0.0 - 172.191.255.255
CIDR:       172.128.0.0/10
NetName:    AOL-172BLK
NetHandle:  NET-172-128-0-0-1
Parent:     NET-172-0-0-0-0
NetType:    Direct Allocation
NameServer: DAHA-01.NS.AOL.COM
NameServer: DAHA-02.NS.AOL.COM
Comment:    ADDRESSES WITHIN THIS BLOCK ARE NON-PORTABLE
RegDate:    2000-03-24
Updated:    2002-08-09

TechHandle: AOL-NOC-ARIN
TechName:   America Online, Inc.
TechPhone:  +1-703-265-4670
TechEmail:  domains@aol.net

OrgAbuseHandle: AOL382-ARIN
OrgAbuseName:   Abuse
OrgAbusePhone:  +1-703-265-4670
OrgAbuseEmail:  abuse@aol.net

OrgNOCHandle: AOL236-ARIN
OrgNOCName:   NOC
OrgNOCPhone:  +1-703-265-4670
OrgNOCEmail:  noc@aol.net

OrgTechHandle: AOL-NOC-ARIN
OrgTechName:   America Online, Inc.
OrgTechPhone:  +1-703-265-4670
OrgTechEmail:  domains@aol.net
```

Source: American Registry for Internet Numbers.

I have tried to search the history of abuse of the IP address through the dshield.org (http://www.dshield.org/ipinfo.php) , but ho abuse history is found.

Description of attack:

The sources were tried to scan the 1080 (socks), 3128 (SQUID PROXY) ,8080 (PROXY. The different source addresses were started to scan the destination IP addresses on Oct 30 but on different time periods. In addition, we also found out the time interval between each scan is about 4 to 5 minutes apart, and the attackers were interested in the 207.166.X.X range network.

Initially, we thought that the attack was categorized as Ring Zero attack, but that was not true at the later stage, because Ring Zero is usually running on Windows platform. As we can see from the Alert file, the attackers are coming from three different sources, each attack was equipped with different TTL (Time to Live), Windows sizes, Datagram lengths which is an indication that the attackers were running different operating systems,

What we suspected here is the attackers would have been looking for Trojans commonly configured to run on these ports. For instances, Winhole (1080), Subseven 2.2 (1080), Reverse WWW Tunnel Backdoor (3128), Brown Orifice (8080), and RemoConChubo (8080).

The above mentioned Trojans are available at the following web site:
http://www.neohapsis.com/neolabs/neo-ports/neo-ports.html

However, the straightforward answer is that the attacker was simply scanning for open proxies and try to exploit it.

Attack Mechanism:

The attackers were basically scanning three ports, they are:

Port 1080: Scanning this port are actually looking for Wingate, which is a very popular firewall/proxy running on Windows Platform.

Port 3128: Scanning this port are typically looking for Squid, Squid is another popular proxy servers running on LINUX platform. The attackers are trying to search for open proxy such that they can use to surf through in order to hide their tracks.

Port 8080: Scanning this port are looking for http server and proxies. Some proxy server may contain files that can be retrieved.

Proxy servers are wide used as a world wide web server for the entire domain or whatever clients that we have placed behind the firewall which is a logical block between the users and the rest of the internet. The Proxy servers are usually provided by the system administrators or network administrators as a security measures, and at the same time, it provides a better and faster web accessing experience.,

Most of network browser installed on the workstation or network are not connected directly to the web server. What we usually do in the office is to direct the connections to the local proxy server/socks server. The socks server/proxy server will then relay the network browser's request to the web server, and pass the results back to the network browser. At the same time, the proxy/socks server will keep a coup of the web page materials for future users to retrieve in order to shorten the time for sending.

Attackers are interested in the scanning the open ports is to hide their true identity. For the majority of the time, users accessing the web server through the proxy server do not require to disclose their true IP addresses, the web server will log the IP addresses of the proxy server only.

The most likely event is the attacker is looking for open proxies from which they can launch attacks against other vulnerable computer from a state of relative anonymity.

Correlations:

There was no other incidents reports for two of the source addresses 216.77.216.104 and 172.184.170.160 at dshield.org.

However I have found some interesting articles for proxy scan:

http://www.incidents.org/archives/intrusions/msg03148.html
scanning 8080 and 1080 ports from a same host, this is from one of the cable provider in Hong Kong. This is quite typical because the attacker wanted to use the proxy server of the provider, and tried to surf through the network for opportunities.

http://www.mcabee.org/lists/snort-users/Dec-01/msg00297.html
For this incident, the attacker even tried to look for individual user.

Evidence of active targeting:

There shouldn't be evidence of active targeting, the attackers are not from single source and at the same time, the attackers are tried to scan many machines in the network until they find a vulnerable machine.

Severity:

Note: The scale is ranked from 1 (lowest) to 5 (highest)

Target Criticality = 4	The target system is a proxy server, and this is considered to be quick critical to the company in the destination site.
Attack Lethality = 2	If the attackers are successfully, we should observe some other further activities, such as scanning to other sites or Denial of service attack etc. However, the attacker may also take this opportunity to chew up resources such as CPU or hard drive in the server.

System Countermeasures = 3	There are no responses from the destination machines. We conclude that the destination server has enough patches or hot fixes.
Network countermeasures = 2	There are no responses from the destination server. The network should installed with some form of defensive mechanism such as firewall or some routers with power access control list.

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

Attack severity : (4+2) – (3+2) = **1** The severity of this traffic is low

Defensive mechanism:

http://www.shield.org/ports/port1080.php - This web site indicated that most of the proxy server are configured to accept connections from anywhere. This shouldn't be allowed on the real network, the proxy server should only restrict connections to the users on the "inside" of the LAN (local area network).

If your site does not use proxies on port 8010 and 3128, or 1080. You can simply block these incoming services. If you do use these ports, ensure the ports are restricted for your site's use only, do not open to the public.

Finally, do remember to apply patches, hot fixes and upgrade your proxy server on a regular basis. Here is a web site that provided useful information for Wingate server, Squid Server, and HTTP proxy server.
http://leb-undernet.org/_Leb_Objective/Leb_Army_Seals/Proxy_Scan/proxy_scan.html

Multiple Choice test Question:

Which of the following statement is true for the proxy scan ?

A. It only runs on Windows platform
B. It scans the port 80, 1080 only
C. It scan the port 1080, 3128 and 8080
D. It looks all kinds of server, such as mail server, file server and proxy server
E. Proxy scan is a virus scan software

Answer: C

Questions for Detect 2:

Here is the link regarding the question asked by Don Murdoch:
http://cert.uni-stuttgart.de/archive/intrusions/2003/07/msg00374.html

One of the question he asked is about the Port 1080 where I described the attacker is looking for Wingate at port 1080:

What support in the alert makes you want to make such a positive statement ? Meaning how can you say the attacker is searching for Wingate, here ?

Answer: The 1080 port is the typical proxy protocol used by SOCKS. The design of the winsock is to allowed a workstation outside the network to connect securely through the firewall. Some network may open port 1080 to tailor for incoming connections to a system running a socks daemon. One of the more common uses of socks is to allow ICQ traffic to host that are behind the firewall.

The Alert file indicated the attacker attempting to perform tcp by issuing SYN request to different workstations with different addresses but targeting at 1080 port. This move is to further differentiate the attacker is trying to look for a telnet redirector (which is a significant implication as per wingate.)

Detect # 3 - Bad traffic tcp port 0

Source of trace:

The trace was downloaded from the incidents.org RAW log file, and the name of the file is 2002.9.15. The downloaded date code is Dec 2, 2002. The complete URL is http://www.incidents.org/logs/Raw/2002.9.15.

Event Traces:

The following is Snort "Alert" data:

```
[**] [1:524:4] BAD TRAFFIC tcp port 0 traffic [**]
[Classification: Misc activity] [Priority: 3]
10/15-12:19:52.376507 211.47.255.22:60921 -> 32.245.155.201:0
TCP TTL:47 TOS:0x0 ID:0 IpLen:20 DgmLen:52 DF
******S* Seq: 0x40F2156  Ack: 0x0  Win: 0x16D0  TcpLen: 32
TCP Options (6) => MSS: 1460 NOP NOP SackOK NOP WS: 0
[Xref => nessus 10074][Xref => cve CVE-1999-0675]

[**] [1:524:4] BAD TRAFFIC tcp port 0 traffic [**]
[Classification: Misc activity] [Priority: 3]
10/15-12:19:58.376507 211.47.255.22:60921 -> 32.245.155.201:0
TCP TTL:47 TOS:0x0 ID:0 IpLen:20 DgmLen:52 DF
******S* Seq: 0x40F2156  Ack: 0x0  Win: 0x16D0  TcpLen: 32
TCP Options (6) => MSS: 1460 NOP NOP SackOK NOP WS: 0
[Xref => nessus 10074][Xref => cve CVE-1999-0675]

[**] [1:524:4] BAD TRAFFIC tcp port 0 traffic [**]
[Classification: Misc activity] [Priority: 3]
10/15-12:20:10.376507 211.47.255.22:60921 -> 32.245.155.201:0
TCP TTL:47 TOS:0x0 ID:0 IpLen:20 DgmLen:52 DF
******S* Seq: 0x40F2156  Ack: 0x0  Win: 0x16D0  TcpLen: 32
TCP Options (6) => MSS: 1460 NOP NOP SackOK NOP WS: 0
[Xref => nessus 10074][Xref => cve CVE-1999-0675]

[**] [1:524:4] BAD TRAFFIC tcp port 0 traffic [**]
[Classification: Misc activity] [Priority: 3]
10/15-12:20:21.356507 211.47.255.22:33245 -> 32.245.155.201:0
TCP TTL:47 TOS:0x0 ID:0 IpLen:20 DgmLen:52 DF
******S* Seq: 0x6BC0169  Ack: 0x0  Win: 0x16D0  TcpLen: 32
TCP Options (6) => MSS: 1460 NOP NOP SackOK NOP WS: 0
[Xref => nessus 10074][Xref => cve CVE-1999-0675]

[**] [1:524:4] BAD TRAFFIC tcp port 0 traffic [**]
[Classification: Misc activity] [Priority: 3]
10/15-12:20:24.356507 211.47.255.22:33245 -> 32.245.155.201:0
TCP TTL:47 TOS:0x0 ID:0 IpLen:20 DgmLen:52 DF
******S* Seq: 0x6BC0169  Ack: 0x0  Win: 0x16D0  TcpLen: 32
TCP Options (6) => MSS: 1460 NOP NOP SackOK NOP WS: 0
[Xref => nessus 10074][Xref => cve CVE-1999-0675]

[**] [1:524:4] BAD TRAFFIC tcp port 0 traffic [**]
[Classification: Misc activity] [Priority: 3]
```

```
10/15-12:20:30.356507 211.47.255.22:33245 -> 32.245.155.201:0
TCP TTL:47 TOS:0x0 ID:0 IpLen:20 DgmLen:52 DF
******S* Seq: 0x6BC0169  Ack: 0x0  Win: 0x16D0  TcpLen: 32
TCP Options (6) => MSS: 1460 NOP NOP SackOK NOP WS: 0
[Xref => nessus 10074][Xref => cve CVE-1999-0675]

[**] [1:524:4] BAD TRAFFIC tcp port 0 traffic [**]
[Classification: Misc activity] [Priority: 3]
10/15-12:20:56.376507 211.47.255.22:33859 -> 32.245.155.201:0
TCP TTL:47 TOS:0x0 ID:0 IpLen:20 DgmLen:52 DF
******S* Seq: 0x91718D5  Ack: 0x0  Win: 0x16D0  TcpLen: 32
TCP Options (6) => MSS: 1460 NOP NOP SackOK NOP WS: 0
[Xref => nessus 10074][Xref => cve CVE-1999-0675]
```

The following is correlating TCPdump log data

```
[**] BAD TRAFFIC tcp port 0 traffic [**]
10/15-12:20:21.356507 211.47.255.22:33245 -> 32.245.155.201:0
TCP TTL:47 TOS:0x0 ID:0 IpLen:20 DgmLen:52 DF
******S* Seq: 0x6BC0169  Ack: 0x0  Win: 0x16D0  TcpLen: 32
TCP Options (6) => MSS: 1460 NOP NOP SackOK NOP WS: 0
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=

[**] BAD TRAFFIC tcp port 0 traffic [**]
10/15-12:20:24.356507 211.47.255.22:33245 -> 32.245.155.201:0
TCP TTL:47 TOS:0x0 ID:0 IpLen:20 DgmLen:52 DF
******S* Seq: 0x6BC0169  Ack: 0x0  Win: 0x16D0  TcpLen: 32
TCP Options (6) => MSS: 1460 NOP NOP SackOK NOP WS: 0
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=

[**] BAD TRAFFIC tcp port 0 traffic [**]
10/15-12:20:30.356507 211.47.255.22:33245 -> 32.245.155.201:0
TCP TTL:47 TOS:0x0 ID:0 IpLen:20 DgmLen:52 DF
******S* Seq: 0x6BC0169  Ack: 0x0  Win: 0x16D0  TcpLen: 32
TCP Options (6) => MSS: 1460 NOP NOP SackOK NOP WS: 0
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=

[**] BAD TRAFFIC tcp port 0 traffic [**]
10/15-12:20:42.356507 211.47.255.22:33245 -> 32.245.155.201:0
TCP TTL:47 TOS:0x0 ID:0 IpLen:20 DgmLen:52 DF
******S* Seq: 0x6BC0169  Ack: 0x0  Win: 0x16D0  TcpLen: 32
TCP Options (6) => MSS: 1460 NOP NOP SackOK NOP WS: 0
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=
```

Detected generated by

The binary tcpdump log file was then analyzed by snort using the following command:

Snort –r d:\snort\ 2002.9.15 –c snort.conf –l c:\mylog

The rules that generated this alert was as follows:

alert tcp $EXTERNAL_NET any <> $HOME_NET 0 (msg:"BAD TRAFFIC tcp port 0 traffic"; reference:cve,CVE-1999-0675; reference:nessus,10074; classtype:misc-activity; sid:524; rev:4;)

An alert message will generate when the packets match the rule “BAD Traffic tcp port 0 traffic”. The rule will only detect the attack which is involved port zero from a source to the destination network, the direction of the traffic will not be taken into consideration. Traffic can be sending from any ports on the source computer.

Probability that the source address was spoofed:

The chances for the source address was spoofed is low because the attacker only wanted to see if there was any responses from the destination computer. Looks like this is a port scan from the source to the destination. The attacker is expecting a response from the target computer in order to evaluate the response and to plan for the next action. However, looks like the target does not respond to the scan, and this is an indication that the machine is not what the attacker thought it might be. This is also not an effective reconnaissance method as the attacker performed port scanning and not seeing any response back.

Description of the attack:

This is a tcp port scan to port 0 on the destination computer. The attacker is 211.47.255.22 and the destination target is 32.245.155.201.
The attacker 211.47.255.22 is using four different ports to perform the scan, which are port number 60360, 60921, 33245 and 33859 respectively.
In addition, there are about 16 packets sending to the destination within two seconds (12:19:17 to 12:21:14). This is considered to be pretty fast port scan. Some products on the internet can perform such fast scanned function. For example, the hping2 uses port 0 as the default destination port when used as an automated port scanning tool. (http://www.hping.org/manpage.html)
On the other hand, the attackers should differentiate the times between sending the packets, as such an attack within a short period of time will can be easily picked up by and Intrusion detection system.

Attack mechanism:

Under normal circumstances, the TCP traffic to port 0 is not valid and this is a reserved port. However, some programming (such as UNIX socket programming) will use port 0 as a programming technique .(See article from Computer Networking) http://compnetworking.about.com/library/ports/blports_0.htm.
For our case, the attack mechanism is probably an automated scanning tool (hping, the latest version is 3). The attacker is trying to look for what type of operating system is running on the targeted machine. Once the targeted machine responds to the attacker,

the attacker will obtain an indication on the targeted machine. The attacker can plan for further attack. In other words, the attack is the reconnaissance for a further attack. In conclusion, the port scan 0 will not compromise the machine but it will help the attacker to obtain more information on the target machine for further attack.

Correlation:

This is an incident I found on the internet with port zero traffic. The writer (Ken Mckinlay) indicated that it was a hping2 port scan. Interesting thing is all 1480 bytes of the payload are empty except the last eight bytes.
http://www.incidents.org/archives/intrusions/msg02179.html
In addition the following document in CVE also illustrated the vulnerability on the port running on Checkpoint firewall. http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0675

Evidence of active targeting:

The attacker appears to be actively targeting the destination computer based up on the following facts:
1. Source address is the same
2. Target address is the same
3. The attacker sends sixteen packets in a unique pattern within a short period of time

Severity:

Note: The scale is ranked from 1 (lowest) to 5 (highest)

Target Criticality = 2	The attack will not compromise the target. The attacker is only attempted to obtain more information on the target computer.
Attack Lethality = 1	The attack only reveal more information about the targeted computer. The attacker will not compromise the target.
System Countermeasures = 3	The destination may generate responses to the attacker. As a result, information about the destination machine such as the OS version may reveal to the attacker.
Network countermeasures = 3	The network should have firewall or some form of access-list to filter out port 0 at the network boundary or at the host level. An additional intrusion detection would also help to catch this traffic.

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

Attack severity : (2+1) – (3+3) = **-3.** The severity of this traffic is low

Defensive mechanism:

Since port 0 is a reserved port, we suggested the administrator to block this port at the very entry level in the firewalls or routers etc. For added on security, additional intrusion detection should be employed. Making sure your intrusion detection sensor is placed in front of the firewall to allow proper sensing of the traffic.
In addition, if you are running some operating system such as UNIX, make sure you have fully employed the port scanning software such as nmap or hping3 to see any responses are generated. Finally, the administrator should apply updated OS patches on a regular basis.

Multiple Choice test question:

A number of tcp packets are sending to port 0 of a particular workstation on your computer, and logged by the Firewall. What of the following is/are correct explanation(s) ?

1. This is an active port scan
2. You need to apply patches on your firewall since your firewall is corrupted
3. Someone is using port 0 as a programming technique for specifying system-allocated (dynamic) ports.
4. Someone is trying to connect to your reserved service
5. Normal traffic between firewall and workstation

A. 1
B. 1 and 3
C. 2
D. 2, 4 and 5

Answer: C

Question 1 from Don Murdoch:
Here is the link regarding the question asked by Don Murdoch:
http://cert.uni-stuttgart.de/archive/intrusions/2003/07/msg00373.html

What types of information would an attacker need to better target a machine with specific exploits ? What would they need in order to know what OS the target is running ?

Answer:
Additional information such as the responses generated by the operating system installed on the target machine, the time taken to give a response to the request. In addition, the time difference between two successful responses etc. When the attacker sends a packet to a remote system, the responses the attacker gets back vary from one operating system to another operating system. Based on the above, the attacker can easily identify the operating system used by the remote target.

Question 3 from Rocker (starplanet1000@yahoo.com.hk)
Date: Thu, 31 Jul 2003 21:53:01 +0800 (CST)
Subject: Re: LOGS: GIAC GCIA Version 3.3 Practical Detect # 3

How do you know there is no response from the host. Remember the pcap file only capture the packets which triggers the snort rules…

Answer: If there are responses from the target, we believed further fingerprinting actions that will trigger other rules and generate different files. We believed the target machine is installed with Firewall, which will drop the traffic. Typical example is the CISCO firewall router, by putting access-list in the configuration, we can simple drop the traffic broadcast to outside.

ASSIGNMENT 3: Analyse this

Executive Summary:

This analysis presented the major issues reported on the MY.NET network. The report covered the five consecutive days from Sunday June 15,2003 to Thursday June 19, 2003. The target network is a university campus whereby the time frame is summer term. Although we believe the traffic in summer term is relatively small when compared with regular term such as Fall, Winter or Spring, we believed the traffic volume can still represent the normal traffic during normal semester.

Different type of network activities were captured by the Intrusion Detection System (IDS). Those type of activities area as follows:

Alerts - Alert are the traffic that matches an actual attack as defined in the intrusion detection software. The ISD software contained the appropriate signatures that look for the specific attacks

Scans – Scans are captured to indicate the attempt from the source to look for the availability and the associated services for the particular host (e.g. a host or a server) on the network

Out of specification errors (OSS) – OSS represents the traffic that does not conform to any normal network activities:

There are 492708 alerts and 5751529 scans reported during the 5 days observed period. Through the analysis on the log,. We found some anomalies behavior occurred in the following major areas:

Denial of Service (DDoS) attack
There are a large amount of information to prove that the denial of services are being launched from the network, main indications are as follows:

- There are more than 30% alerts coming from an ftp activities from external host to internal network. The address for the external host are well known on the dshield.org for attacking other computers on port 80 and port 21.
- More than 95% in the scan logs are UDP scan. Although it did not indicate further activities, we believe there is some unusual activities going on in the network

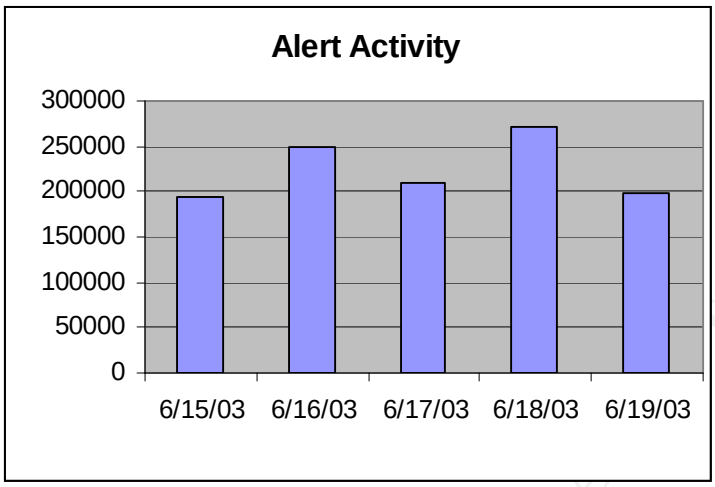
Security standard on the existing network

- The existing security standard is not “tighten enough” or “too loose” to allow other traffic on the network.
- Ftp is allowed on the between the inside network to the outside network
- 50% is targeted on port 6667 (where the port is opened for telnet.)
- 11% is targeted on 27374, which is the port vulnerable to Ramen worm on unpatched LINUX system

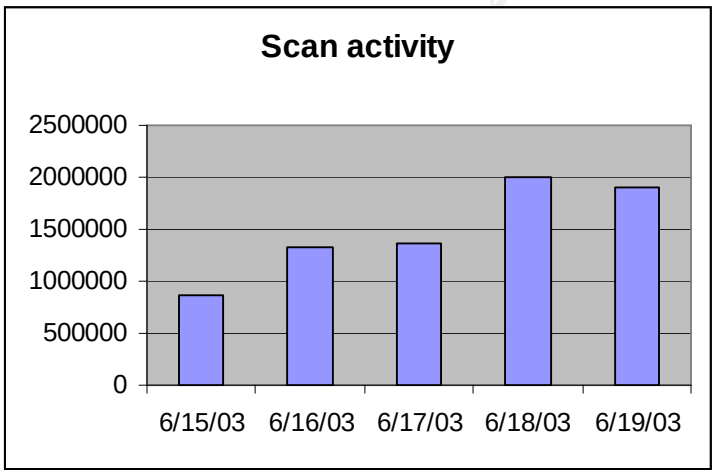
There should be some successfully compromised machine in the internal network, there are more than 80% of the alerts are associated with exploits or worms.

Reviewing the log also indicated some interesting things. The highest alerts and scans were reported on Wednesday, June 18 2003, while the lowest alerts and scans were reported on June 15, 2003. This is probably due to a Sunday, where most of the people not spent their time on the network.

The following graph indicated the alert activity:

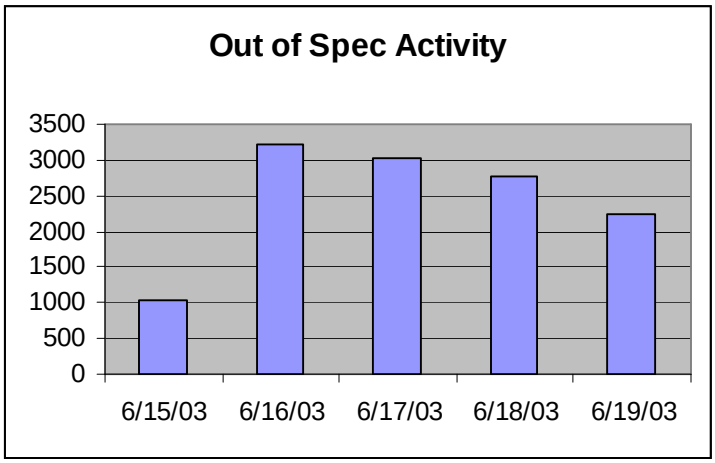


The following graph indicated the scan activity:



Out of specification (OOS) activity as generally less than scans or alerts traffic. The OOS activity reached the highest on June 16, 2003, around 3200 detects. The OSS activity is relatively insignificant when compared with the scans and alerts traffic.

The following graph indicated the OSS activity:



List of Files:

This analysis used files from http://www.incidents.org/logs in the month of June, 2003. Those specific ones used in the detailed analysis covered the dates from June 11, 2003 to June 15, 2003:

Here are the lists of the files :

Filename	Size(Bytes)	Date
alert.030611.gz	2,320,998	Sun Jun 15 05:00:43 2003
alert.030612.gz	2,667,783	Mon Jun 16 05:00:37 2003
alert.030613.gz	2,606,795	Tue Jun 17 05:00:45 2003
alert.030614.gz	3,241,395	Wed Jun 18 05:00:53 2003
alert.030615.gz	2,345,865	Thu Jun 19 05:00:51 2003
scans.030611.gz	6,252,031	Sun Jun 15 05:00:48 2003
scans.030612.gz	9,265,581	Mon Jun 16 05:00:42 2003
scans.030613.gz	10,141,045	Tue Jun 17 05:00:52 2003
scans.030614.gz	14,229,077	Wed Jun 18 05:01:04 2003
scans.030615.gz	13,794,682	Thu Jun 19 05:01:01 2003
OOS_Report_2003_06_11_13995	471,043	Wed Jun 11 00:05:20 2003
OOS_Report_2003_06_12_2042	1,448,963	Thu Jun 12 00:05:32 2003
OOS_Report_2003_06_13_16083	1,351,683	Fri Jun 13 00:05:33 2003
OOS_Report_2003_06_14_8730	1,228,803	Sat Jun 14 00:05:26 2003
OOS_Report_2003_06_15_31799	1,003,523	Sun Jun 15 00:05:30 2003

Except for the OSS files, all alert and scan files are zipped, we used winzip V8.1, unzipped the files and prepared the following tables:

LOG FILES					
ALERT		SCANS		OUT OF SPECIFICATION	
alert.030611	22,705,000	scans.030611	52,759,000	OOS_June11	461,000
alert.030612	27,818,000	scans.030612	81,710,000	OSS_June12	1,416,000
alert.030613	24,875,000	scans.030613	84,574,000	OSS_June13	1,321,000
alert.030614	32,487,000	scans.030614	124,027,000	OSS_June14	1,201,000
alert.030615	23,409,000	scans.030615	117,346,000	OSS_June15	981,000
total	131,294,000	Total	460,416,000	Total	5,380,000

The logs analyzed for this report represented the following sizes:

- Alert log of 131M size
- Scan log of 460M size
- Out of specification log of 5M size

Analysis Process:

To conduct the analysis, each set of files was concatenated into a single file covering the entire five days worth of data, before doing that, we used the following techniques to massage the data:

Step 1: copy alert.* alert_all.txt
This is to copy all the alert files into one single file

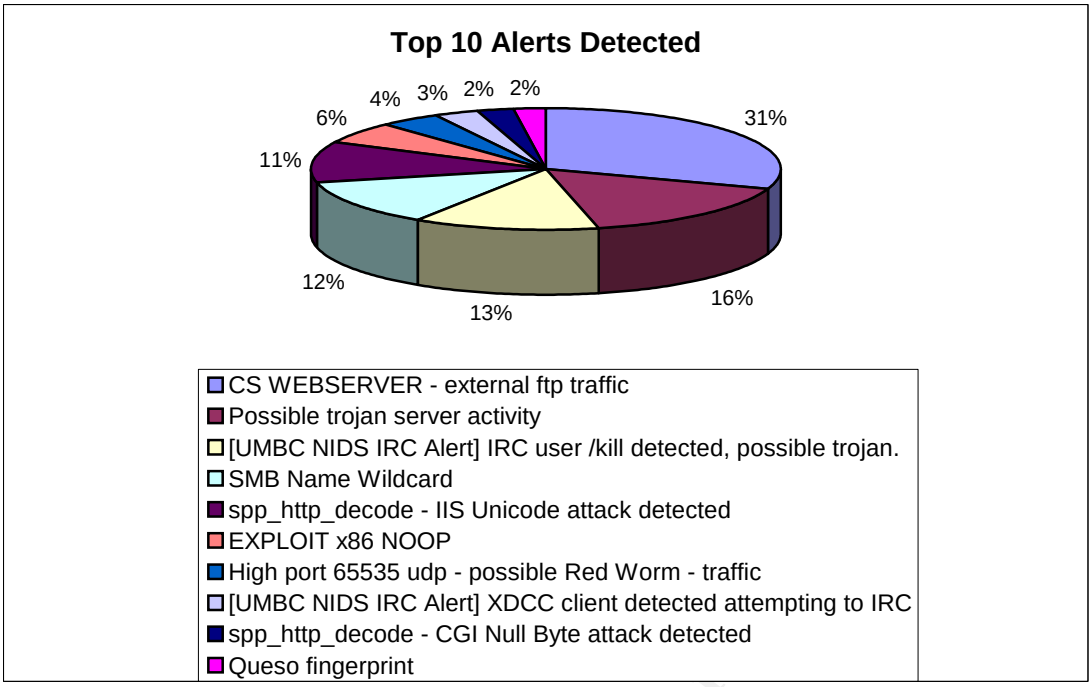
Step 2: sort alert_all.txt /o alert_sorted.txt
Sort the alert_all.txt file and output to an alert_sorted.txt file

Step 3: type alert-sorted.txt | t excl 'spp_portscan:' > salert.txt
We reduced the sorted file into salert.txt by removing all the lines containing : "spp_portscan" since these were caught in the scans file.

Step 4: type salert2.txt | t repl '[**]' ':' '#2D#3E' ':' > alert_final.txt
Finally, we further fine tuning the salert2.txt file by removing some special characters such as "**" or "->" and replaced by comma separated value ":" before imported to Microsoft Access Database for further mining of information.

Alert Activity:

The majority of the attacks on the network are mainly external ftp traffic between the webserver inside the network and the external hosts:



The following table indicated the top ten alerts logged over the reporting period and the number of occurrences they were generated. Detailed analysis of each attack will follow together with the relationships between the attacked machine and possibility compromised machine

	Alert generated	no. of Hits
1	CS WEBSERVER - external ftp traffic	137748
2	Possible trojan server activity	73355
3	[UMBC NIDS IRC Alert] IRC user /kill detected, possible trojan.	58514
4	SMB Name Wildcard	55621
5	spp_http_decode - IIS Unicode attack detected	48903
6	EXPLOIT x86 NOOP	26723
7	High port 65535 udp - possible Red Worm - traffic	19039
8	[UMBC NIDS IRC Alert] XDCC client detected attempting to IRC	13664
9	spp_http_decode - CGI Null Byte attack detected	11205
10	Queso fingerprint	10407

CS Webserver – external ftp traffic:

We found an enormous amount of FTP traffic are coming from the following IP addresses to the host. As refer to the following information from dshield.org, we believed the following addresses have bad records in attacking others.

- 213.156.38.104
- 213.156.38.106
- 213.156.38.107

- 213.156.35.135
- 213.156.35.136
- 213.156.35.138
- 213.156.35.139

We have done a quick lookup from the dshield.org, and found out quite a large number of attacks were reported from the address range, details are as follows:

IP Address: 213.156.38.104 – 213.156.38.107
HostName: 213-156-38-104.fastres.net
DShield Profile:

Country:	IT
Contact E-mail:	abuse@fastweb.it
Total Records against IP:	1874
Number of targets:	425
Date Range:	2003-05-20 to 2003-06-28

Update Summary

Top 10 Ports hit by this source:

Port	Attacks	Start	End
80	273	2003-06-28	2003-07-27
137	132	2003-06-29	2003-07-14
4662	15	2003-07-26	2003-07-28
21	13	2003-06-28	2003-06-30
1223	1	2003-07-08	2003-07-08
6697	1	2003-06-30	2003-06-30
19407	1	2003-07-24	2003-07-24

IP Address: 213.156.35.135 – 213.156.35.139
HostName: 213-156-35-135.fastres.net
Dshield Profile:

Country:	IT
Contact E-mail:	abuse@fastweb.it
Total Records against IP:	1190
Number of targets:	24
Date Range:	2003-05-20 to 2003-07-24

Update Summary

Top 10 Ports hit by this source:

Port	Attacks	Start	End
21	1242	2003-06-30	2003-07-28
1214	152	2003-07-22	2003-07-23
6346	55	2003-07-01	2003-07-24
2234	13	2003-06-29	2003-07-13
4662	11	2003-07-06	2003-07-25
139	9	2003-07-06	2003-07-23
13341	6	2003-07-17	2003-07-17
7274	6	2003-06-30	2003-06-30
80	6	2003-07-14	2003-07-16
6767	3	2003-07-27	2003-07-27

inetnum: 213.156.35.128 - 213.156.35.143
netname: FASTWEB-POP-4105-RESIDENTIAL
descr: Infrastructure for Fastweb's main location
descr: NAT IP addresses for residential customer, public subnet
country: IT
admin-c: IRS2-RIPE
tech-c: IRS2-RIPE
status: ASSIGNED PA
mnt-by: FASTWEB-MNT
changed: IP.RegistrationService@fastweb.it 20020408
remarks: In case of improper use originating from our network,
remarks: please mail customer or abuse@fastweb.it
remarks: INFRA-AW
source: RIPE

We believed the sources are attempting to establish ftp across the internet and tried to attack the inside network.

FTP (File transfer Protocol) is one of the oldest and most often used protocols on the internet, and is often integrated into our browsers. It will facilitate the file transfers across the internet.

Problems:
FTP will not count for host verification, authentication or data protection.
For basic FTP, server doesn't really have any means of verifying that the client's background and information. Same to the client, the ftp client never the server is who they are talking to. Both ends of this transaction are open to man-in-the-middle attack, the data can be monitored, modified or substituted.

The following link will give out more information on the FTP exploits and attacks

http://www.pintday.org/whitepapers/ftp-review.shtml#exploits

This link also highlights some of the vulnerabilities of the IIS server by DoS via malformed ftp connection http://www.kb.cert.org/vuls/id/412203

In addition, we also found there are still some web traffic between the server (MY.NET.100.165) and outside addresses, most of them are from the ISPs which are located all over the world. We suspected this is an IIS server which provided browsing function to the general public, and also enable the browser to upload or download information or file across the network.

We also believed that through the web browsing, and the file transfer. The attackers are trying to compromise the web server, and then begin to attack other machines on the network.

The link graph will give out more information in a later stage.

Possible Trojan server activity:

What we can see from the there are quite a number of activities between 211.217.184.210 and MY.NET.70.164.

211.217.184.210 → MY.NET.70.164 (41325 hits)
MY.NET.70.164 → 211.217.184.210 (32030 hits)

```
06/11-05:11:30.899129  : Possible trojan server activity :
211.217.184.210:27374 : MY.NET.70.164:4662
06/11-05:11:31.051127  : Possible trojan server activity :
211.217.184.210:27374 : MY.NET.70.164:4662
06/11-05:11:31.053919  : Possible trojan server activity :
MY.NET.70.164:4662 : 211.217.184.210:27374
```

As we can see from the above detect, in a relative short period of time, the source 211.217.184.210 attempted to explore the target MY.NET.70.164, and the target tried to respond back to the source. We suspected that the source is attempted to load code on the target, and the target should a Microsoft 2000 based server. This activity is typically associated that has been with IRC flood, which is matched with the following alert.

IRC User/Kill detected, possible Trojan:

The source did not appear to exploit any product related security vulnerabilities, and it also did not relate to viral or worm-like in nature. Instead, the source is trying to explore the weakness of the server, and perhaps to take advantage of situations where the standard precautions have not been taken in setting up the server. Looks like the source tried to compromise the target. Successful compromises will leave a distinctive pattern on the server.

Most of the traffic that triggered the snort rule matched the following representative traffic:

```
06/11-00:13:02.081070  [**] [UMBC NIDS IRC Alert] IRC user /kill detected,
possible trojan. [**] 195.159.0.86:6667 -> MY.NET.114.116:1810

06/11-00:31:56.109188  [**] [UMBC NIDS IRC Alert] IRC user /kill detected,
possible trojan. [**] 195.159.0.84:6667 -> MY.NET.114.116:1839
```

If MY,NET.114.116 is a domain controller, we also believed the security policies were modified. Type effects are as follows:

- Guest accounts that were previously disabled are re-enabled.
- New unauthorized accounts, possibly with administrative privileges.
- Security permissions are changed on servers or in Active Directory
- Users are not able to login the domain controller from their workstation

Recommendations:

- Do not use weak administrator password
- Deactivated the guest account
- Run updated anti virus program with latest patches applied
- Use firewall to protect the servers and the domain controller

SMB Name Wildcard:

The snort rules detect indicated the source is targeting the port 137 of the MY.NET.135.224

```
06/11-00:26:11.327494  [**] SMB Name Wildcard [**] 63.156.48.185:1026 ->
MY.NET.135.224:137
06/11-00:26:11.642795  [**] SMB Name Wildcard [**] 63.156.48.185:1026 ->
MY.NET.135.226:137
```

This is the basic behavior of Window servers that use NetBIOS (as well as DNS) to resolve IP addresses to names using the “gethostbyaddr()” function. The following link will provide more information.
http://www.iss.net/security_center/advice/Exploits/Ports/groups/Microsoft/default.htm
The remote users will try to surf the Windows-based web sites, and the server will respond with NetBIOS lookups.

This link will try to explore the vulnerability of the NetBIOS and the associated loophole:
http://www.sans.org/resources/idfaq/port_137.php

Correlations:
Jim Hindrick noticed similar traffic and indicated the traffic should not be seen from outside to inside network.
http://www.giac.org/practical/GCIA/Jim_Hendrick_GCIA.pdf

Recommendation:
Since there are quite a number of machines outside the network targeting the Windows network port 137. We suggested to block this port in the firewall access-list.

Spp_http_decode – IIS Unicode attack detected:

By accessing the ..// directory of the Microsoft IIS sever, the source attempted to exploit the vulnerabilities of the Microsoft IIS server. The following link highlighted some vulnerabilities of the IIS Unicode Attack: http://www.securityfocus.com/bid/1806

Our log indicated that 217 external outside addresses talking to 189 internal destination addresses, while 457 internal source addresses targeting 893 external destination addresses.

The following link will give out some vulnerabilities associated with IIS Unicode attack. http://www.sans.org/resources/idfaq/iis_unicode.php.
This link mentioned about Red Worm, Red Worm II, and Nimda Worm virus. However, RED worm virus is most likely the attack that contributed to this alert. As the following alert also picked up by the snort rules(we will provide more information on the following alert). In addition, there may be some false positive as the users inside the network are tried to browse the internet.

Correlations:
Tod Beardsley(http://www.giac.org/practical/Tod_Beardsley_GCIA.doc) pointed out a very good issue in his paper. We need to know the types of the machine running on the network. If there is no Microsoft IIS server on the network, the traffic will become suspicious.

Recommendations:
We can protect the system servers from the vulnerabilities by installing the recent service packs and security updates from the vendor. In addition, try to customize the directory name or locations instead of using the default one. Further, try to set permission carefully for shares. Finally, we can try to turn off all unneeded functions in the IIS.

EXPLOIT X86 NOOP:

This is probably a sign of buffer overflow, which the attacker can gain access and take control on the server if successful. The following two links will probably give some highlights on the attack
http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-1035
http://www.whitehats.com/info/IDS291

Here are some captures from the alert file:

```
06/11-04:21:08.375255  [**] EXPLOIT x86 NOOP [**] 195.24.203.122:10682 ->
MY.NET.111.21:80
```

```
06/11-04:21:08.387360  [**] EXPLOIT x86 NOOP [**] 195.24.203.122:10682 ->
MY.NET.111.21:80
06/11-04:21:08.394115  [**] EXPLOIT x86 NOOP [**] 195.24.203.122:10682 ->
MY.NET.111.21:80
```

We can see the connection is from a very high port (10682) to port 80. Looks like someone is trying to download files from the through the server from outside sources. Perhaps someone is downloading large files from an outside server.

Recommendations:
The best approach is to implement a firewall together with proper intrusion detection system. Then place the servers inside the firewall. The IDS will actively monitor the high level ports. In addition, we can try to restrict the size of the attachments or files for the users.

High port 635535 udp – possible Red worm – traffic:

Red worm virus is the predecessor of "Adore" worm. This is a worm that spreads in LINUX systems by using different types of vulnerabilities,e.g. lpd or we-ftpd. It searched the vulnerable hosts on the network, and download worm to the victim server, which ultimately opened a "backdoor". The backdoor will then get activated by receiving a ping packet with correct size, and opened a shell in the port 65535. The worm will eventually send out sensitive system data to four different email addresses.

The following link gave a very good write up on this worm:
http://www.sans.org/y2k/adore.htm

Additional information can be found on the following link:
http://www.f-secure.com/v-descs/adore.shtml

Correlations:
I have found some papers on the previous practical which have similar descriptions:
http://www.giac.org/practical/Matthew_Fiddler_GCIA.doc
http://www.giac.org/practical/Jim_Hendrick_GCFW.zip

Recommendation:
- Upgrade the version of BIND on your system.
- Modify the access-list in your router or firewall.
- Update your sensor in the Intrusion Detection equipment.
- Filter tcp 53 activities including zone transfers, data requests larger than 484 bytes and load balancing packages

XDCC Client detected attempting to IRC:

There are two machines (MY.NET.83.100 and MY.NET105.204) on the internal network attempted to distribute files in an IRC channel to outside parties. The machines are believed to be hacked.

```
06/12-09:00:24.821926  [**] [UMBC NIDS IRC Alert] XDCC client detected
attempting to IRC [**] MY.NET.83.100:1854 -> 64.235.110.34:6667
06/12-01:50:15.731842  [**] [UMBC NIDS IRC Alert] XDCC client detected
attempting to IRC [**] MY.NET.105.204:4242 -> 66.150.99.99:6667
```

We believe these two machines were compromised through the administrative share with no administrative password on Windows 2000. The two machines targeted the destination port 6667 which is the default IRC port.

Recommendation:

- Disable file sharing on the network
- Set stringent password, do not use default administrator password.
- If you have firewall, try to block outgoing sends on port 139 for all machines

The following paper provided more information on the XDCC
http://www.russonline.net/tonikgin/EduHacking.html

Spp_http_decode – CGI Null Byte attack detected:

CGI (Common Gateway Interface) script is a program written in Pearl and C script. It is the initial mechanism used to make web sites interact with databases and other written applications. The Null Byte attack masks command from CGI security checks by hiding the commands (most re executable) behind a “null byte” – a packet of data that CGI scripts do not detect.

The following five alerts examples indicated where most CGI Null byte traffic was traveling from and to. The largest contributor is MY.NET.97.218.

```
06/15-00:04:02.035803  [**] spp_http_decode: CGI Null Byte attack detected
[**] MY.NET.97.104:1729 -> 64.14.122.229:80
06/15-00:04:02.035803  [**] spp_http_decode: CGI Null Byte attack detected
[**] MY.NET.97.104:1729 -> 64.14.122.229:80
06/15-00:04:02.035803  [**] spp_http_decode: CGI Null Byte attack detected
[**] MY.NET.97.104:1729 -> 64.14.122.229:80
06/15-02:15:09.768797  [**] spp_http_decode: CGI Null Byte attack detected
[**] MY.NET.97.104:2426 -> 216.73.87.102:80
06/15-03:01:25.429398  [**] spp_http_decode: CGI Null Byte attack detected
[**] MY.NET.97.104:50848 -> 65.121.78.100:80
```

As these packets indicated, the traffic are coming from the MY.NET.97.104 network and pointing to the port 80 of external device, and it generated more than 1000 alerts during the past five days. There are other machines in the MY.NET network which caused more than 1000 alerts, all are directing to 80 ports of outside machine.
The machines are MY.NET.163.76, MY.NET.168.98, MY.NET.84.216. However, these machines may be compromised and tried to attack outsider machines at port 80. The following paper provided a good description.
http://archive.infoworld.com/articles/hn/xml/02/04/03/020403hniss.xml
http://www.cgisecurity.com/papers/fingerprint-port80.txt

Correlations:
From Brian Cahoon's paper,
(http://www.giac.org/practical/GCIA/Brian_Cahoon_GCIA.pdf) He also mentioned the large amount of alert files are generated from internal network, and targeted external computers.

Recommendation:
This is probably a normal web traffic, it is difficult to examine what actually caused the alerts with out looking at the payload of the data. Our suggestion is to investigate the four machines that generated the alert. In addition, periodically examine the log in the PROXY server, and revisit the IDS rules to ensure proper prevention.

Queso Fingerprint:

Queso is use to detect what types of operating system is running on the remote computer.

```
06/14-00:00:12.208141  [**] Queso fingerprint [**] 216.95.201.22:50000 ->
MY.NET.6.47:25
06/15-00:02:14.718112  [**] Queso fingerprint [**] 216.95.201.16:60479 ->
MY.NET.6.40:25
06/15-00:15:32.650884  [**] Queso fingerprint [**] 216.95.201.23:51850 ->
MY.NET.6.40:25
```

Joe Rayford's paper also covered the QUESO Fingerprint, and came to the same analysis.
http://www.giac.org/practical/Joe_Rayford_GCIA.doc

Although this is not a big concern to the network, it is a better idea to check the source of the fingerprinting and applied access list in the firewall to protect the inside network. The following link provided some technical background on the Queso fingeprint.
http://xforce.iss.net/xforce/xfdb/2048.

Alert traffic Top talkers and ports:

The following table indicates the top ten IP addresses where the alerts were coming from.

	Source	Arin.net/RIPE/APNIC Information	Number of Count
1	66.207.164.23	Cologuys	11422
2	150.214.191.55	Red Information	3516
3	211.217.184.210	Korean Telecom	2718
4	68.49.35.0	Comcast Cable Communications, Inc	2441
5	MY.NET.70.164	Internal Network	1932
6	142.59.75.234	AGT	1393
7	211.104.1.5	Korean Network Information Center	1289
8	81.51.23.240	France Telecom	1226
9	MY.NET.83.100	Internal Network	1177
10	68.54.94.58	Comcast Cable Communications, Inc	996

The majority of alerts are came from the Service Providers across the world, namely Korea, France as well as USA. I think probably the attackers are attempted to conduct the attack through the Internet service provider.

I looked up the IP address from the following three sources:
ARIN – American Registry for Internet Numbers
http://www.arin.net/tools/whois_help.html

APNIC – Asia Pacific Network Information Center
http://www.apnic.net/

RIPE - IP Europeens Network Co-ordination Centre
www.ripe.net/perl/whois

```
OrgName:    ColoGuys
OrgID:      CLGY
Address:    8101 Chapin Rd
City:       Fort Worth
StateProv:  TX
PostalCode: 76116
Country:    US


NetRange:   66.207.160.0 - 66.207.175.255
CIDR:       66.207.160.0/20
NetName:    COLOGUYS-1
NetHandle:  NET-66-207-160-0-1
Parent:     NET-66-0-0-0-0
NetType:    Direct Allocation
NameServer: NS1.COLOGUYS.COM
```

```
NameServer: NS2.COLOGUYS.COM
NameServer: NS3.COLOGUYS.COM
Comment:    ADDRESSES WITHIN THIS BLOCK ARE NON-PORTABLE
RegDate:    2001-12-20
Updated:    2001-12-27

TechHandle: JM3108-ARIN
TechName:   Montroll, Jon
TechPhone:  +1-817-560-0305
TechEmail:  Noc@cologuys.com
OrgTechHandle: JM3108-ARIN
OrgTechName:   Montroll, Jon
OrgTechPhone:  +1-817-560-0305
OrgTechEmail:  Noc@cologuys.com

# ARIN WHOIS database, last updated 2003-08-01 19:15
# Enter ? for additional hints on searching ARIN's WHOIS
database.
```

```
inetnum:      150.214.0.0 - 150.214.255.255
netname:      RICA
descr:        Red Informatica Cientifica de Andalucia
descr:        Sevilla
country:      ES
admin-c:      MAO4-RIPE
tech-c:       ALG9-RIPE
status:       ASSIGNED PI
remarks:      mail spam reports: abuse@rediris.es
remarks:      security incidents: cert@rediris.es
notify:       iris-nic@rediris.es
notify:       lizana@cica.es
mnt-by:       REDIRIS-NMC
changed:      miguel.sanz@rediris.es 19941213
changed:      iris-nic@rediris.es 19980622
changed:      iris-nic@rediris.es 19980727
changed:      iris-nic@rediris.es 19990823
changed:      iris-nic@rediris.es 20021128
changed:      er-transfer@ripe.net 20030416
```

```
netname:      KORNET
descr:        KOREA TELECOM
descr:        KOREA TELECOM Internet Operating Center
country:      KR
admin-c:      DL276-AP
tech-c:       WK81-AP
remarks:      ****************************************
remarks:      Allocated to KRNIC Member.
remarks:      If you would like to find assignment
remarks:      information in detail please refer to
remarks:      the KRNIC Whois Database at:
remarks:      http://whois.nic.or.kr/english/index.html
remarks:      ****************************************
mnt-by:       MNT-KRNIC-AP
mnt-lower:    MNT-KRNIC-AP
```

```
changed:      hostmaster@apnic.net 20000901
changed:      hostmaster@apnic.net 20000912
changed:      hostmaster@apnic.net 20010627
status:       ALLOCATED PORTABLE
source:       APNIC
person:       Dongjoo Lee
address:      Korea Telecom
address:      128-9 Youngundong Chongroku
address:      SEOUL
address:      463-711
country:      KR
phone:        +82-2-747-9213
fax-no:       +82-2-766-5901
e-mail:       ip@ns.kornet.net
nic-hdl:      DL276-AP
mnt-by:       MNT-KRNIC-AP
changed:      hostmaster@nic.or.kr 20010523
source:       APNIC
```

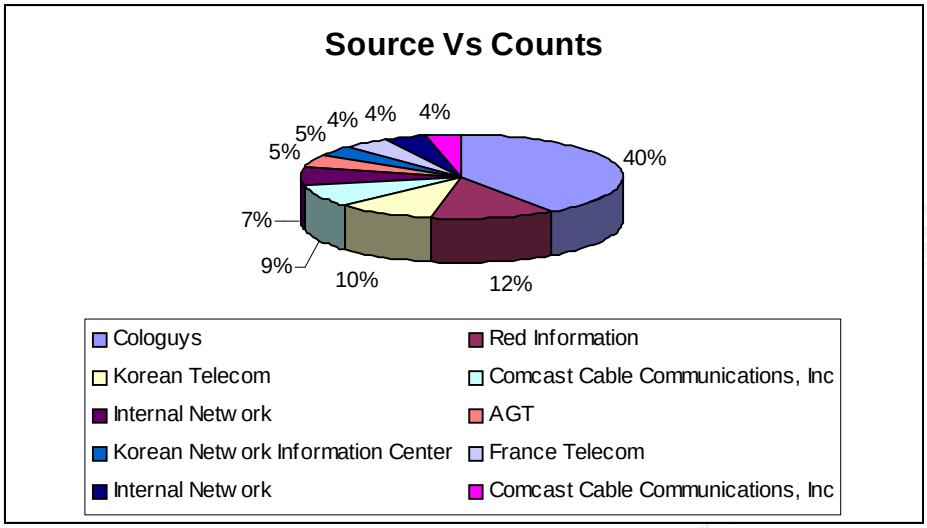
```
 Comcast Cable Communications, Inc. JUMPSTART-1 (NET-68-32-0-0-1)
                                  68.32.0.0 - 68.63.255.255
 Comcast Cable Communications, Inc. DC-3 (NET-68-48-0-0-1)
                                  68.48.0.0 - 68.49.255.255

 # ARIN WHOIS database, last updated 2003-08-01 19:15
 # Enter ? for additional hints on searching ARIN's WHOIS database.
```

```
     OrgName:    AGT
     OrgID:      AGT-2
     Address:    10035 - 102 Avenue
     City:       Edmonton Alberta
     StateProv:  AB
     PostalCode: T5J-0E5
     Country:    CA

     NetRange:   142.59.0.0 - 142.59.255.255
     CIDR:       142.59.0.0/16
     NetName:    AGT
     NetHandle:  NET-142-59-0-0-1
     Parent:     NET-142-0-0-0-0
     NetType:    Direct Assignment
     NameServer: LITHIUM.BCTEL.NET
     NameServer: SODIUM.BCTEL.NET
     RegDate:    1994-05-23
     Updated:    2001-03-30

     TechHandle: ZA75-ARIN
     TechName:   AGT
     TechPhone:  +1-877-310-4638
     TechEmail:  hostmaster@telusplanet.net

     OrgTechHandle: IA86-ARIN
     OrgTechName:   IP Admin, IP
     OrgTechPhone:  +1-403-503-3800
     OrgTechEmail:  add-req.tac@telus.com
```

```
inetnum:        211.104.0.0 - 211.119.255.255
netname:        KRNIC-KR
descr:          KRNIC
descr:          Korea Network Information Center
country:        KR
admin-c:        HM127-AP
tech-c:         HM127-AP
remarks:        ****************************************
remarks:        KRNIC is the National Internet Registry
remarks:        in Korea under APNIC. If you would like to
remarks:        find assignment information in detail
remarks:        please refer to the KRNIC Whois DB
remarks:        http://whois.nic.or.kr/english/index.html
remarks:        ****************************************
mnt-by:         APNIC-HM
mnt-lower:      MNT-KRNIC-AP
changed:        hostmaster@apnic.net 20000414
changed:        hostmaster@apnic.net 20010606
status:         ALLOCATED PORTABLE
source:         APNIC
```

```
inetnum:        81.51.23.0 - 81.51.23.255
netname:        IP2000-ADSL-BAS
descr:          BSNCY107 Nancy Bloc1
country:        FR
admin-c:        WITR1-RIPE
tech-c:         WITR1-RIPE
status:         ASSIGNED PA
remarks:        for hacking, spamming or security problems send mail to
remarks:        postmaster@wanadoo.fr AND abuse@wanadoo.fr
mnt-by:         FT-BRX
changed:        gestionip.ft@francetelecom.com 20021212
changed:        gestionip.ft@francetelecom.com 20030318
source:         RIPE
route:          81.51.0.0/16
descr:          France Telecom
descr:          Wanadoo Interactive
origin:         AS3215
remarks:        -----------------------------------------
remarks:        For Hacking, Spamming or Security problems
remarks:        send mail ONLY to      abuse@wanadoo.fr
remarks:        -----------------------------------------
notify:         addr-reg@rain.fr
mnt-by:         RAIN-TRANSPAC
changed:        karim@rain.fr 20021126
source:         RIPE
```

The following graph highlighted the number of counts for the top ten addresses against the ownership information.



Top ten source ports for Alerts:

We also interested the ports where most of the alert traffic coming from. The following table presented the top ten ports for the alerts. We also found out the corresponding services from the following link :
http://www.iana.org/assignments/port-numbers

	Source port	Count	Service
1	6667	11920	IRC
2	27374	2750	TCP port
3	1309	2360	J-TAG server
4	4662	1910	Unassigned
5	1027	1810	Exosee
6	0	1312	Rserved TCP/IP port
7	1025	899	Network Blackjack
8	1026	772	Calender Access Protcol
9	1028	680	Unassigned
#	137	597	NetBIOS-ns

The port 6667 was reported to be the highest count. Port 6667 is usually found associated with Windows 2000 telnet service, and is subject to the denial of service attack. The problem is if the remote attacker tried to open multiple telnet session and leave the session idle, the server does not know how to terminate the session. The

attacker may use the vulnerability to deny service to legitimate users. One remedial solution is to apply patches on the server,

The second highest count is the port 27374. This is the port where the Ramen worm would like to attack via the unpatched Linux systems. It will probably defaces the web servers, and tried to interfere with some networks traffic.

The third highest count is the port 1309, which is the nt-passfilt-not-found. The attacker will look for the special permission (by default) to add files into the system32 directory in order to install a Trojan password filter. The administrator should install a strong password filter.

Other ports such as 4662, 1027, 1025, 1026, 1028 are basically low ephemeral port. These ports are generally used by the workstations to connect to the servers.

Port zero is a reserved port in TCP/IP networking. The attacker is attempted to connect to port 0 and performing reconnaissance work. If a response is triggered, the far end should be a UNIX machine. In addition, denial of service can be done by sending enormous amount of TCP port 0 to the system. We also suggested to block this port under the firewall

TOP ten addresses where the alerts are going:

The following table presented the top ten addresses where the alerts are going:

	Destination	Count	ARIN.net Information
1	MY.NET.100.165	31595	Inside network
2	MY.NET.190.95	11429	Inside network
3	MY.NET.30.4	4357	Inside network
4	MY.NET.70.164	2772	Inside network
5	MY.NET.30.3	2501	Inside network
6	211.217.184.210	1907	Korean Telecom
7	MY.NET.114.116	1014	Inside network
8	MY.NET.24.15	996	Inside network
9	MY.NET.88.223	982	Inside network
10	208.194.163.37	561	UUNet Technologies

The majority of the traffic are pointing to the MY.NET inside network, where the rest of the traffic pointing to the Korean telecom and UUNET Technologies.

Top ten destination ports for Alert traffic:

The following table presented the top 10 destination ports for the alert traffic:

	Source port	Count	Service
1	80	38773	http
2	137	9048	NetBIOS - NS
3	4662	2774	Unassigned
4	524	2623	NCP
5	27374	1939	TCP port
6	21	1417	FTP
7	0	1310	Reserved TCP/IP port
8	6667	1203	IRC
9	25	1155	SMTP
10	515	996	Unix Lp, Lpr line printer

The number one source port for the alert traffic is the http port, which indicated the MY.NET has a web server for browsing. The second source port is the port 137, which is used for NETBIOS, that also indicated that the MY.NET network contains a Microsoft Windows Server. As we have explained before, the Window server should only serve as file and print sharing. Under normal circumstances, there is no need to share with the computer outside the network.

We also seen port 27374 being triggered, this is one of the most commonly probed ports on the internet, due to the inclusion within the subseven Trojan. Subseven provides the ability to manipulate the compromised system to scan on its behalf. This allow the attackers to scan with impunity. We suggested to block this port in the firewall or the access list in the router.

FTP is the most common exploited service on the internet. It allows the outside to open a connection to exchange files to the server. However, we suggested do not configure the FTP service for someone who is able to read and write to the same directory. In addition, the administrator should move the incoming files to an outgoing directory for security purposes.

Port 0 is also being captured under the alert traffic. Port zero is a reserved port in TCP/IP networking. The attacker is attempted to connect to port 0 and perform reconnaissance work. If a response is triggered, the far end should be a UNIX machine. In addition, denial of service can be done by sending enormous amount of TCP port 0 to the system. We also suggested blocking this port under the firewall

SCAN activity:

The following table highlighted the top TEN scans detected on the network, and the number of counts associated with the scans.

	Attacks	Counts
1	UDP	5638301
2	SYN	1821284
3	NULL	662
4	INVALIDACK	578
5	FIN	461
6	NOACK	363
7	UNKNOWN	152
8	VECNA	80
9	XMAS	16
10	NMAPID	8

As we can see from the above table, the majority of the scan is the UPD scan (roughly 79%), followed by SYN scan (20%), The next eight top scans only accountable for 1%. Obviously, the UPD and SYN scanning are the dominated scans on the internal network.

Definitions for each scan are listed as follows:

UDP Scan:
UDP Scan is a method trying to locate what types of UDP services are available on a computer

SYN Scan:
SYN scan only sends out packets with only SYN flag set. During the three steps handshaking, the machine only sends the SYN packet; this is prevent the service from even being notified of the incoming connection. By doing so, the attacker can easily identify what machine is listening, and without being logged by those services listening at the ports. This scan also names as half open scan.

NULL scan:
Null scan will unset all flags available in the TCP header which makes ACK, FIN, RST, SYN, URG, PSH all become assigned. By sending this packet to the server, it will informs the kernel to drop the incoming call if the port is opened. This is also called the null scan.

INVALIDACK scan
INVALIDACK scan consisted the invalid combination of other flags together with ACK flag sets.

FIN scan
FIN scan sends out FIN packets to the host, which attempted to close a connection that is not open. If there is a service is listening, the operating system will silently drop the incoming packet. On the other hand, the operating system will generate an error message if there is no service is listening. By doing so, the attacker can easily identify the host on the network.

NOACK scan
If the ACK bit is not set, and the packet is not a SYN , RST or any known scan type. The packet is flagged as a NOACK scan. This is used to identify the host or service being deployed on the network.

UNKNOWN scan
Unknown scan does not follow the standard rules for the flag combination. For example, it will detect one reserved bit in addition to the six defined flags.

VECNA scan
Vecna scan is to set all the bits even the reserved bit. However, the PUSH flag without ACT is often identified as VECNA. Nmap, queso and others do this as part of os detection, and this is used to identified the existence of the host or service.

XMAS scan
XMAS scan turns on the invalid flag combination such as Urgent, Push and FIN flags. This may allow an attacker to circumvent the host or service.

NAMPID scan
NAMPID scan turns on the Urgent, Push, SYN and FIN flags. This is also an invalid flag combination which allows the attacker to identify the host or services.

Scan traffic Top talkers and ports:

The following table indicated the top ten sources for scanning. Obviously, all sources are came from the University of Maryland Baltimore County. The university staff should further trace down the workstation, and to determine what types of services they are running, and to ensure the workstations are configured properly with appropriate patches applied.

	Source	Count	Arin.net information
1	130.85.1.3	2402227	University of Maryland Baltimore County
2	130.85.1.4	383011	University of Maryland Baltimore County
3	130.85.153.190	338943	University of Maryland Baltimore County
4	130.85.83.170	247649	University of Maryland Baltimore County
5	130.85.100.230	220700	University of Maryland Baltimore County
6	130.85.97.222	207834	University of Maryland Baltimore County
7	130.85.97.37	191322	University of Maryland Baltimore County

8	130.85.97.129	160003	University of Maryland Baltimore County
9	130.85.153.223	152590	University of Maryland Baltimore County
10	130.85.97.81	150580	University of Maryland Baltimore County

Top ten source ports by scanning traffic:

The following table indicated the top 10 source ports by scanning traffic:

	Count	Source Port	Service
1	2390354	32814	Reserved Port
2	493506	6257	Unassigned
3	379054	32852	Unassigned
4	147515	22321	Unassigned
5	129144	7674	Unassigned
6	63049	12203	Unassigned
7	43106	12300	Unassigned
8	10163	61132	Unassigned
9	9499	4672	Remote file access server
10	1134	4593	Unassigned

Almost all of the ports are the high ephemeral ports. Ephemeral ports are generally used by the client applications to open and run on the servers. The port is available for reuse after the connection is terminated.

TOP ten addresses where the scans are going:

The following table presented the top ten addresses where the scans are going :

Destination Address	Count	APIN Informat
192.26.92.30	81585	Versign Global Registry Services
205.231.29.244	68815	UUNET Technologies
205.231.29.243	60683	Robert E Seastrom
192.148.252.171	52337	Versign Global Registry Services
130.94.6.10	50301	Verio Inc.
192.5.6.30	33327	Versign Global Registry Services
216.109.116.17	32343	Hotjobs.com
192.52.178.30	29287	Versign Global Registry Services
213.130.63.233	26344	RIPE Network Coordination Centre
66.33.98.17	26183	Dialtone Inc
209.208.92.254	25006	Internet Connect Company, Inc.

The top address is the Versign Global Registry services. In fact, there are four sources pointing towards Versign. This is a company that provides services on network, DNS services. For the rest of the companies, they are also sort of ISP companies, the attacker are trying to look for vulnerabilities existed in the ISP, and try to explore more targets for attack.

Top ten destination ports for scan traffic:

The following table presented the top ten destination ports for scan traffic:

	Destination Port	Count	Service
1	53	2797814	Domain Name Server
2	137	1160009	NetBIOS name server
3	80	651075	World Wide Web - http
4	6257	464878	WinMX file sharing
5	25	237444	SMT (simple mail transfer)
6	445	175940	Mircosoft DS
7	22321	145238	Unassigned
8	7674	130924	IMQTunnels
9	4000	91799	ICQ
10	443	91560	SSL encrypted http

Traffic using the Domain name server port makes up the majority of the scan traffic in the top ten destination ports. Interestingly, we found out the port 80 is not the top priority for the destination ports. Most of the ports are associated with Windows server, which implied that the network is consisted of Windows 2000 server.

Out of Specification Traffic:

Out of specification (OOS) packets are packets that do not conform to TCP/IP specifications.

A majority of the OSS packets looked similar to the following:

```
06/10-12:54:41.520737 212.106.150.180:3602 -> MY.NET.88.223:1624
TCP TTL:42 TOS:0x0 ID:20952 IpLen:20 DgmLen:60 DF
12****S* Seq: 0x4EE3E803  Ack: 0x0  Win: 0x16D0  TcpLen: 40
TCP Options (5) => MSS: 1460 SackOK TS: 159103744 0 NOP WS: 0

06/11-00:00:01.744061 12.255.198.216:47052 -> MY.NET.24.44:80
```

```
TCP TTL:40 TOS:0x0 ID:54464 IpLen:20 DgmLen:60 DF
12****S* Seq: 0x54E080CC  Ack: 0x0  Win: 0x16D0  TcpLen: 40
TCP Options (5) => MSS: 1460 SackOK TS: 28464140 0 NOP WS:

06/12-00:06:01.390098 216.95.201.32:40659 -> MY.NET.6.47:25
TCP TTL:48 TOS:0x0 ID:53502 IpLen:20 DgmLen:60 DF
12****S* Seq: 0x67CB9532  Ack: 0x0  Win: 0x16D0  TcpLen: 40
TCP Options (5) => MSS: 1380 SackOK TS: 783304796 0 NOP WS: 0
```

As the common thing we observed is the SYN flag is turned on together with two reserved bits.
All the traffic are coming from an external address to the inside MY.NET network. The targeting destination ports are 25, 80, 113, 1624 and 8088 respectively.
Port 25 is used for mail server and port 80 is used for World Wide Web connection. Port 113 is used for ident server authentication service. Port 1624 is a port for udp service, while 8088 is an ephemeral port. Looks like the outsider is trying to fingerprint and attempting to explore the vulnerability of the workstations inside the network. The amount of traffic reported by OSS is comparative small to alert and scan files.

Top ten IP addresses for Scan and Alert traffic:

The following table presented the top ten relationships for Alert traffic:

	Count	source IP -> destination IP
1	55345	66.207.164.23 -> MY.NET.190.95
2	50120	68.170.69.138 -> MY.NET.30.4
3	11440	68.49.35.0 -> MY.NET.30.3
4	7026	MY.NET.83.100 -> 64.235.110.34
5	5775	MY.NET.83.100 -> 208.194.163.37
6	4688	68.81.2.19 -> MY.NET.30.3
7	3860	MY.NET.97.104 -> 203.161.233.132
8	3526	68.48.110.245 -> MY.NET.30.4
9	3516	150.214.191.55 -> MY.NET.100.165
10	2864	207.151.67.140 -> MY.NET.100.158

The table indicated the number of counts occurred in the alert traffic. The attacker logon the ISP (Cologuys and Comcast Cable) and tried to explore the vulnerability in MY.NET.X.X network. The infected servers/workstations on the MY.NET network will further exploit outside network.

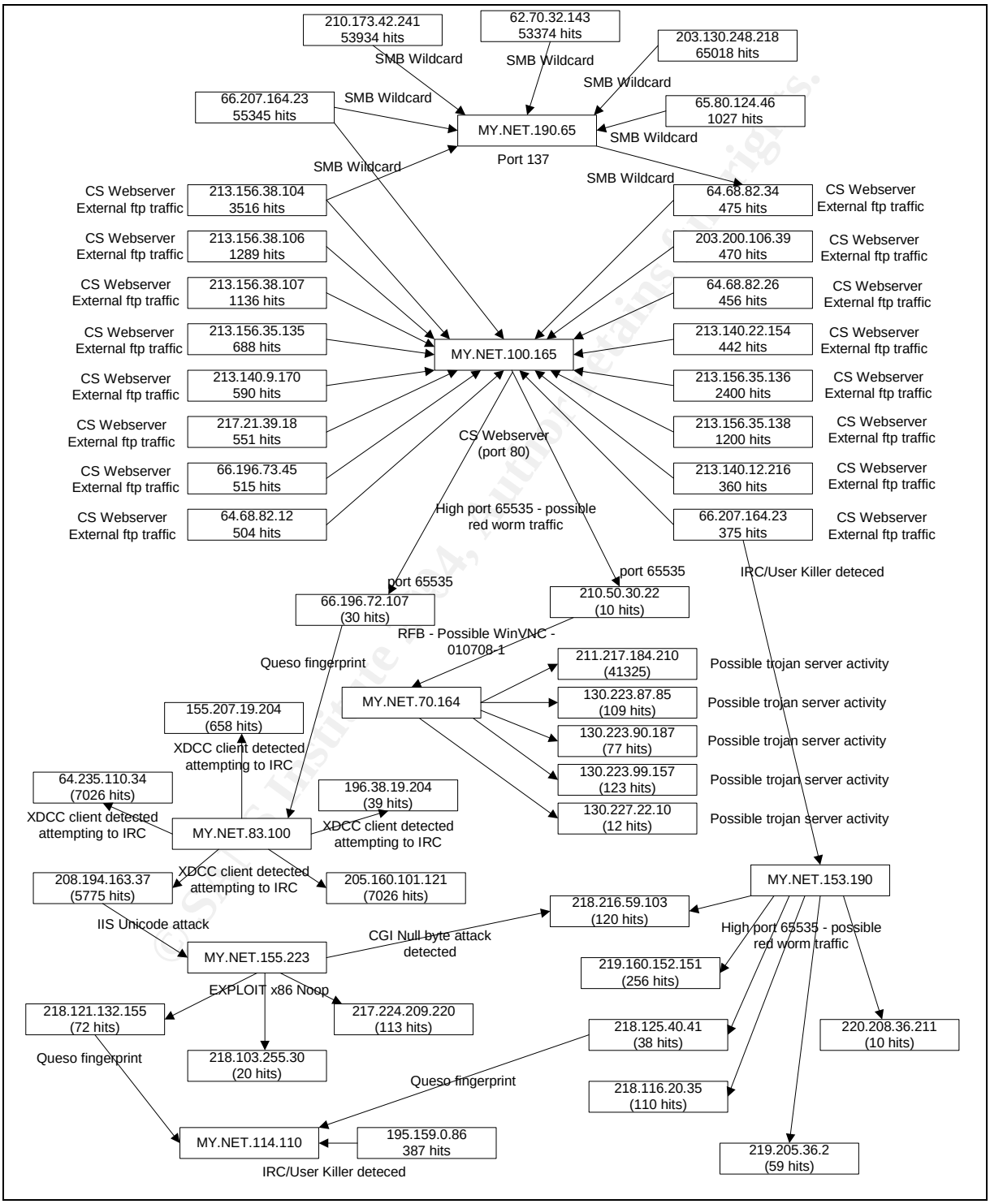
The following table presented the top relationship for scan activity:

	Count	Source IP -> destination IP
1	70703	130.85.1.3 -> 192.26.92.30
2	68815	130.85.1.3 -> 205.231.29.244
3	60683	130.85.1.3 -> 205.231.29.243
4	52337	130.85.1.3 - 192.148.252.171
5	50298	130.85.1.3 -> 130.94.6.10
6	29625	130.85.1.3 -> 192.5.6.30
7	28634	130.85.1.3 -> 216.109.116.17
8	26339	130.85.100.230 -> 213.130.63.233
9	26172	130.85.1.3 -> 66.33.98.17
10	26129	130.85.1.3 -> 192.52.178.30

We found out all the traffic are came from 130.85.1.3 which belongs to the University of Maryland, Baltimore County. Looks like this workstation is trying to explore the outside networks on a frequent basis. The network administrator should further trace down to see what activities this particular station are trying to do.

Link Graph Analysis:

By using the information from the alert log files, the link graphs shown below were generated. The direction of the arrows indicated the packet flow direction. The figures with bracket indicated the number of hits received from a source. The figures without bracket indicated the number of hits actually initiated.



Defensive mechanism:

As discussed before, we have provided quite a number of suggestions and defensive mechanisms on the activities that we have captured and deduced on the logs. However, the following defensive mechanisms are worthwhile to mention:

We believe the first step in the process, is to perform a through scan on the network, especially targeting the internet computers such as Windows server, web server, and email server. This is to make sure they are not being used as unwitting attack platform. Besides, remember to document the vulnerability for reference.

The second step is to secure the servers. For UNIX based server, we can simply turn off all the unnecessary server services. Many default services accompanied with your operating system are not required by the server, and can be turned off. For instances, basic ftp and telnet services are being turned on by default and are easily found as executable programs in the file system. Besides, you can consult with your software vendor on a regular basis to find out if there are additional kernel services that are not run in the system documentation, and the corresponding way to disable them.

For Windows based server, we can download updated patches from the internet. In addition, we can simply turn off all unnecessary services in your web server services. E.g. Java support, and CGI support.

Finally, we can take appropriate action to ensure the server is only accessible to the designated personals such as the administrator.

On the other hand, individual computer serves as a workstation can perform the following :
For those workstations with internet access, we need to inform the computer users that their workstations could be used as the attack agents, and they must install the latest detection software. Besides, they need to install the latest anti virus program.

Finally, remember to perform the following act ion items on your routers, firewalls and intrusion detection system:

- Disable IP directed broadcast functions on your routers
- If the packets are not originate from your network, simple filter it out by access list
- Disable all ICMP echo and echo reply packets onto the network
- Filter echo reply packets are the router facing the outside network and drop them
- Monitor the logs on a regular basis, and pay special attentions on the packets which do not originate from the network.
- Monitor for exceptional high volumes of echo reply and request packets.

List of references:

Some references are generic links as they were used many times for different researches:

http://www.siliconedefense.com

http://www.iss.net/

http://www.google.com.au/

http://www.securityfocus.com/search

http://www.dshield.org/

http://packetstormsecurity.nl/

http://www.infosyssec.net/

http://www.fr1.cyberabuse.org/

http://xforce.iss.net/index.php

http://www.snort.org/

http://www.treachery.net/tools/ports/lookup.cgi

http://www.seifried.org/security/ports/

http://www.w3.org

http://www.giac.org/practical/GCIA/Brian_Cahoon_GCIA.pdf

http://www.giac.org/practical/Hee_So_GCIA.doc

http://www.giac.org/practical/Kyle_Haugsness_GCIA.zip

http://www.giac.org/practical/Roland_Lee_GCIA.doc

http://www.giac.org/practical/Todd_Chapman_GCIA.doc

http://www.giac.org/practical/GCIA/Paul_Bradley_GCIA.doc

http://www.giac.org/practical/GCIA/Freeland_Chew_GCIA.pdf

http://www.giac.org/practical/GCIA/Jim_Hendrick_GCIA.pdf

http://www.giac.org/practical/Tod_Beardsley_GCIA.doc

http://www.giac.org/practical/GCIA/Ricky_Smith_GCIA.pdf

http://www.hkcert.org/