ANALYSIS AND DESIGN OF INTRUSION DETECTION SYSTEM IMPLEMENTATION

A Thesis Submitted to the

Faculty of Computer Science and Information Technology

University of Malaya

By

CHIA FOOK KEONG

In Partial Fulfillment of the Requirement for the Degree of

Master of Information Technology

(6/36 credit hours)

JUNE 2002/2003

ABSTRACT

Nowadays, firewall has been widely used to enforce network security policy in organizations. However, maintaining a good and up to date security policy is not an easy task. Furthermore, maintaining a firewall is even harder. A slip of mouse will let the hackers to drive through the firewall easily. Sometime, a badly configured firewall will engender a false sense of security. This can be worse than no firewall at all. As such, Intrusion Detection System (IDS) has been introduced as a second line of defense to protect an organization. IDS can be either host-based, network based or integrated. The functions of IDS include continuous monitoring and analysis of users and system activities as well as auditing system configurations and vulnerabilities.

This report studies the implementation issues of IDS. The IDS chosen was Snort, which is a free, open source, lightweight, multi-platform and customizable software. The Faculty of Computer Science and Information Technology (FCSIT), University of Malaya network has been chosen as the testing site. First, this study analyzes the environment and protocols run in the FCSIT network. The study finds that FCSIT network has multiple virtual local area networks (VLANs) and is running Hot-Standby Routing Protocol (HSRP) and Network Address Translation (NAT). Through the analysis, both HSRP and NAT affect the IDS implementation. Secondly, IDS is implemented in selected locations and the data gathered are analyzed. Network and system weaknesses discovered are rectified. The IDS is then fine tuned to reduce false alarm and improve detection performance. Through this, FCSIT network security is further enhanced.

Acknowledgement

First, I wish to thank God, the creature of this universe, for giving me the opportunity to commence and complete this project.

I wish to express my deepest gratitude to my supervisor Mr. Phang Keat Keong for his guidance, support and encouragement given throughout the entire project. Besides that, I also wish to thank Mr. Ling Teck Chaw for his extraordinary assistance in reading and commenting on this dissertation.

Last but not least, I also wish to record my thanks to my parents Chia Yim Choy and Low Yit Ho who have always been supportive.

Contents

List of Figures

List of Tables

Abbreviations

CPU	Central Processing Unit
CRM	Customer Relation Management
DDOS	Distributed Denial of Service
DOS	Denial of Service
FCSIT	Faculty of Computer Science and Information Technology
HIDS	Host-based Intrusion Detection System
HSRP	Hot-Standby Routing Protocol
HTTP	Hyper Text Transfer Protocol
ICMP	Internet Control Message Protocol
IDS	Intrusion Detection System
IP	Internet Protocol
NAT	Network Address Translation
NIDS	Network-based Intrusion Detection System
SCM	Supply Chain Management
SPAN	Switched Port Analyzer
TCP	Transmission Control Protocol
VLAN	Virtual Local Area Network
VMPS	VLAN Membership Policy Server
VPN	Virtual Private Network
VTP	VLAN Trunking Protocol

Chapter 1 Introduction

The development of client/server technology has led to a myriad of new security problems. More so, the growth in Internet application has driven organizations to open their networks to wider audiences in order to remain competitive. Applications such as Customer Relation Management (CRM), Supply Chain Management (SCM) and E-Procurement are being widely used and deployed. Such open networks have exposed the organizations to intrusions. Intrusions include attempts to compromise the confidentiality, integrity, availability, or to bypass the security mechanisms of a computer system or network.

Computer Security Institute (CSI) and Federal Bureau of Investigation (FBI) of San Francisco (CSI/FBI Computer Crime and Security Survey, 2002) have reported that in 1996, 16% of organizations surveyed reported intrusion to law enforcement. This figure has increased to 34% in year 2002. The intrusion rate is likely to be higher since many organizations are reluctant to admit or failure of detecting attacks.

1.1 Introduction to network security threats

William S. (2000) has reported that two of the most publicized threats to network security are intruders and malicious codes.

1.1.1 Intruders

There are two main types of intruder: **hacker** and **cracker**. Dictionary.com defines hacker as "One who is proficient at using or programming a computer, a computer

buff'. Hackers are persons who like to know how things work without malicious intention. On the contrary, crackers are malicious hackers with malicious intent. (CSI/FBI Computer Crime and Security Survey, 2002) reported that 90% of the respondents detected computer security breaches in the past 12 months. Intruders could be outsiders and insiders.

Outsiders are from outside of a corporate network. These intruders may attack both the external presence like deface web servers or the machines on the internal network. The attacks could be originated from the Internet, dial-up lines, physical break-ins, or extranet i.e., connections to business partner's network.

Insiders are legitimate internal network users. These include users who misuse privileges or who impersonate higher privileged users.

Besides stealing corporate information or data for profit, intruders may also launch denial of services (DOS) attacks. (CSI/FBI Computer Crime and Security Survey, 2002) reported that 40% of the respondent detected DOS attacks.

1.1.2 Malicious Code Attacks

Viruses, worms and Trojan horses are malicious codes that hide within files or programming codes. These codes are able to self-replicate, self-propagate or be spread by ignorant computer users. Nimda and CodeRed for example are worms, which do not require human interaction to spread, instead using known software vulnerabilities and multiple vectors of infection. (CSI/FBI Computer Crime and

Security Survey, 2002) has reported that 85% of the respondents detected computer viruses.

The attacks are capable of damaging or compromising the security of individual computers as well as the entire networks. Computer Economics (2001) estimates that virus and worm attack costs of $10.7 billion in year 2001. These costs were incurred to clean-up infected servers, as well as inspection of other servers.

1.2 Impact of network security threats

The direct impacts of network security threats include financial loss, business disruptions, legal liability and potential lawsuits. The indirect impacts are loss of competitiveness and damages to organization's brand name. Generally, (CSI/FBI Computer Crime and Security Survey, 2002) have shown an up trend on the losses reported by the respondents. Below is the statistic of the annual losses reported.

Table 1: Total annual losses caused by computer crime

Attacks	Total annual losses (USD$ 'million)					
	1997	1998	1999	2000	2001	2002
Theft of proprietary information	20	33	42	66	151	170
Denial of services	n/a	2	3	8	4	18
Financial fraud	24	11	39	55	92	115
Inside abuse of net access	1	3	7	27	35	50
Sabotage of data or networks	4	2	4	27	5	15
Virus	12	7	5	29	45	49

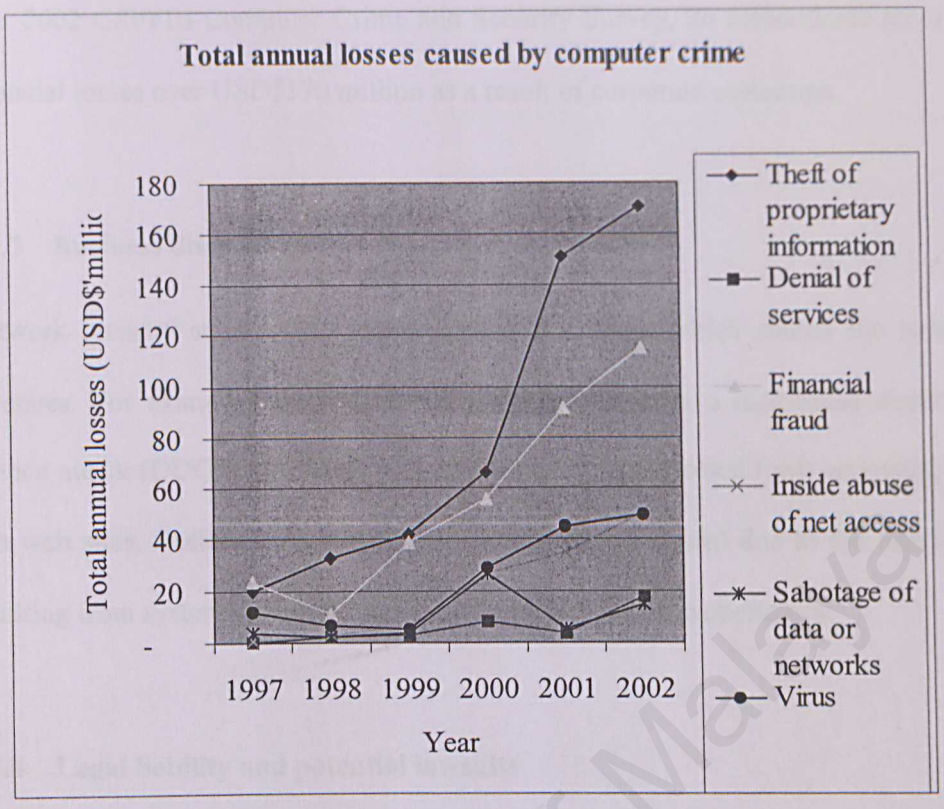


Figure 1: Total annual loss caused by computer crime

1.2.1 Financial fraud

Network security threats may cause unauthorized transaction being effected or critical data being modified. In the year 2002 CSI/FBI Computer Crime and Security Survey, 25 respondents reported losses of USD$115 million due to financial fraud.

1.2.2 Loss of competitiveness

American Society for Industrial Security and PricewaterhouseCoopers (1999) "Trends in Proprietary Information Loss" Survey Report cited that 70% or more of an organization's value rests in intellectual property assets. However, this critical and sensitive information may be stolen and causing reduced market share. In the

year 2002 CSI/FBI Computer Crime and Security Survey, 26 respondents reported financial losses over USD$170 million as a result of corporate espionage.

1.2.3 Business disruptions

Network security attacks may cause system downtime, which causes the loss of revenues. For example, Amazon and CNN suffered from a distributed denial-of service attack (DDOS) in February 2000 and users were denied from accessing the two web sites. Besides that, tremendous resources were burnt due to the idle time resulting from system downtime and the efforts to restore the operations.

1.2.4 Legal liability and potential lawsuits

Under the privacy and security regulations, organizations may need to compensate customers for losses suffered arising from intrusions. Also, organizations need to prove due diligence in prudent security governance or face potential lawsuits.

1.2.5 Damages to organization brand name

Successful intrusions may tarnish an organization's corporate image, which was built over long time and after spending millions on advertising programs. For example, organization that suffered theft of customer data like credit card information, will face adverse publicity and loss of both existing and potential customers.

1.3 Introduction to Security Measures

Since the attacks may come either from internal or external and the severe impacts, organizations are inevitably need to design and implement security measures to ensure data security. The IDC (2002) reported that 91% of the organizations surveyed in April 2002 would invest in designing and implementing security measures. In July 2002, IDC reported that 94% of the same organizations remain with the decision to invest in security measures. These security measures include anti-virus tools, firewall appliances and Intrusion Detection System (IDS). This study, however concentrates on IDS.

1.3.1 Introduction to IDS

Robert Graham (2000) defined **intrusion** as an attempt to break into or misuse a system like stealing confidential data or distributing Spam. Intrusion detection is the process of monitoring the events occurring in a computer system or network and analyzing them for signs of intrusion. The concept of intrusion detection was first introduced by John Anderson (1980) through technical report "*Computer Security Threat Monitoring and Surveillance*".

Most of the research was dated between 1980s and 1990s. Among the research, the Intrusion Detection Model proposed by Dorothy Denning (1987) provided a methodological framework for further researches in IDS. The model also formed the road map for IDS commercial products. According to Ant Allan (2002), IDS is commercially used from the mid 1990s. IDS products have evolved from the

automation of log parsing activity to more sophisticated functions to monitor network traffics to detect malicious activities.

IDS can provide assurance whether system configuration is secure and other security measures such as firewall operates properly. The assurance is provided through the following IDS functions:

- Monitoring and analysis of users and system activities
- Auditing of system configurations and vulnerabilities
- Detecting changes of critical system and data files
- Statistical analysis of network traffic patterns via matching known attacks
- Abnormal network traffic analysis

1.4 The importance of IDS

While deploying security measures like anti-virus tools, firewalls and Virtual Private Network (VPN) can minimize the network exposure, are these security measures adequate? Information systems and networks are subject to attacks, especially in an open network environment like Internet. Perhaps, an analogy helps to better assess the situation.

Imagine of a house in real life that normally is guarded by a wall and solid front gate with locks. However, it is always possible that a thief can probe the possible loopholes or weaker links such as back door, windows and the roof for intrusion. Eventually, the thief may sneak into the house and steal away valuable goods without being noticed. Hence, an alarm system that is capable of alerting the house owner of any probing and intrusion is essential.

The above analogy of a house's security depicts an organization's network security. An intruder may be observing the organization's network for a long time without being noticed. While firewalls are protecting the main gateways, firewalls do not alert the organization of any possible vulnerability in the network infrastructure. As such, it is essential and timely to have IDS in place. This detection process monitors computer system or network activities. These activities are analyzed to detect possible intrusions.

1.5 Motivation

The site selected for this study is the Faculty Computer Science and Information Technology (FCSIT) network. The main motivations for this study on IDS are as follows:

- Quite a number of the FCSIT computers are open to Internet and hence, vulnerable to attacks from outsiders.
- Firewall is unable to detect system misconfiguration, which may pose security risks and allow gaps for intruders. More so, firewall is unable to detect any problems in the internal networks.
- Quite often, some of the FCSIT computers are infected by malicious codes like CodeRed, Nimda and other computer viruses. Should these attacks be not detected promptly, more computers will be affected. Tremendous resources in cleaning up and rectifying the damages may be incurred.

1.6 Objective

The main objectives of this project are as follows:

- To study the needs for IDS
- To study IDS
- To analyze the FCSIT networks environment and protocols run
- To design the IDS implementation
- To implement the designed IDS
- To rectify the system weaknesses noted and fine tune the IDS design

1.7 Scope

The scope of this project covers the network security threats and the conventional network security measures like firewall and anti-virus tools. Various push and pull factors for implementing IDS are discussed. The study reviews the selection criteria of IDS both available at commercial off-the shelves and open source. An IDS will be chosen for testing.

This study analyzes the FCSIT network topology, hardware, software and the protocols running which affect the IDS implementation. Based on the analysis, locations to implement IDS are identified. The selected IDS is implemented at the FCSIT networks. The data collected from the IDS implementation is analyzed to identify the network and system weaknesses. These discovered weaknesses would be rectified. In addition, the IDS will be fine tuned to reduce false alarm and improve detection performance.

1.8 Report organization

Chapter 2 covers the strength and weaknesses of various types of the IDS. Popular IDS software available at both commercial off-the shelves and open source are assessed against certain evaluation criteria. An IDS is to be selected for implementation in FCSIT networks.

Chapter 3 analyzes the network environment of the FCSIT, including the network protocols, software deployed and topology that affecting the design of IDS implementation. This chapter will also identify the resources that need to be protected and the sources of security threats. Accordingly, the implementation plan for IDS is designed.

Chapter 4 details the implementation of the selected IDS in FCSIT networks. The details include the steps of IDS set-up at designated locations and the configurations of IDS.

Chapter 5 shows the results of the implementation of the selected IDS in FCSIT networks. The results gathered will be analyzed and the IDS design will be fine-tuned to achieve the desire results i.e., strengthen the FCSIT networks security.

Chapter 6 provides a conclusion for this project.

1.9 Summary

This chapter explores the existing network security threats and conventional network security measures. IDS is highlighted as one of the key network security component complementing the conventional network security measures.

The project motivation, objectives, scope and report organization are also presented.

Chapter 2 Literature Review

The main objectives of this chapter are to discuss the types of Network Intrusion Detection Systems and the corresponding strength and weaknesses. Various IDS available either commercial off-the shelves or open source are reviewed and assessed based on certain evaluation criteria. A suitable IDS will be selected for implementation in FCSIT networks.

2.1 Network security attacks

There are 2 broad category of network security attacks i.e., active attacks and passive attacks. Active attacks are the deliberate actions of an attacker to gain access to compromise the confidentiality, integrity or availability of data or system like DOS and break ins. Passive attacks are actions in the nature of eavesdropping to gather information like sniffing and information gathering. Most often, passive attacks are more difficult to detect since there are no traceable activities. Passive attacks are used together with active attacks to gain unauthorized access or launching DOS attacks.

2.1.1 DOS attacks

A DOS attack is aimed to deprive user from using the available resources of an organization. The attack normally will flood a resource with excessive data or Transmission Control Protocol (TCP)/Internet Protocol (IP) packets that can be handled. This attack will prevent a user from gaining access to computer resources such as email server, web server or database server. Besides attacks

from a single host, the attacker may also use a group of computers in different locations to launch a DDOS attack against a target.

According to Mandy Andreas (2001), there are three main types of DOS:

1. Buffer overflows – This is the most common type of denial of service attacks which attacker sends more data than an application's buffer can hold. The extra data overflow may cause the system to crash or execute certain machine codes. For example, an oversized Internet Control Message Protocol (ICMP) packet may cause a target system to crash.

2. SYN attack – This attack exploits the weakness of TCP. An attacker by sending a large number of SYN packets with no corresponding ACK will cause the receiving system to run out of memory.

3. Teardrop attack – This attack exploits the IP. An attacker by putting a confusing offset value in some of the fragment packets will cause the receiving system unable to reassemble the packets. This will lead to system crash.

2.1.2 Sniffing

It is common that passwords or sensitive files are not encrypted while these data are transmitted via network. Sniffing traffic involves monitoring the network traffics of a targeted organization. An attacker can plant a sniffer in a network via a Trojan Horse program and distributed through emails or embedded into certain programs. Upon opening up the email or executing the program, the sniffer will

be installed on the user's computer. This sniffer will collect the network traffic and send back to the attacker.

This attack normally is overlooked since no noticeable damages are made. However, the information gathered through sniffing can then be used to launch active attacks or information stolen.

2.1.3 Information gathering

Before an attacker can hack into a network, the attacker normally gathers as much information as possible for the targets. This information includes the environment either human or system. Human environment is referred to the behavior of the personnel involved and operations. System environment includes the type of machines, operating systems and versions used and services running.

Based on the information gathered, an attacker will map out the network of an organization. Attempts are then can be made to identify the vulnerabilities that can be exploit to gain access.

According to Eric Cole (2001), the steps to gather information may includes the followings:

- Determine target's IP address by using command like Whois or Nslookup.
- Determine network address range by using American Registry for Internet Numbers (ARIN) or Traceroute.

- Look out for active machines through Ping command. Ping War can be used to do a ping sweep.
- Map out the open ports or access points by using Portscanners, Nmap, ScanPort, War Dialers or THC-Scan.
- Identify the operating system used via Queso or Nmap
- Identify the services running via the default port and operating system, Telnet and vulnerability scanners like SARA, SAINT and Nessus.
- Develop the network map by using Traceroute, Visual ping or Cheops.

2.2 Network security measures

The basic network security measures are anti-virus tools, firewall and recently IDS.

2.2.1 Anti-Virus Tools

Anti-virus tools are critical in the organizations' network security in defending against known viruses and other malicious code. The IDC (2002) reported that 80% of the organizations surveyed are investing in anti-virus tools. However, these tools have their limitations since the viruses are evolved in a fast pace and becoming more sophisticated and widely spread. As a result, anti-virus tool vendors and organizations are forced to catch up with the anti-virus updates.

2.2.2 Firewall Appliances

Firewalls are also widely deployed by many organizations. The IDC (2002) reported that about 60% of the surveyed organizations are investing in firewall

appliances. These appliances function as the front-line security mechanisms in protecting the organizations' networks against intrusions from external sources, like the Internet. Nevertheless, firewall may not cover all the external connections, for example unauthorized dial-ups to the Internet. Also, intrusions may also originate from internal of the organization, i.e., behind the firewall either due to accidentally or maliciously attempts to access files or systems.

2.2.3 IDS

An "Intrusion Detection System (IDS)" is a system collecting information from a variety of systems and network sources. The information gathered is further analyzed for detecting intrusions. IDS can be broken down into the following categories:

1. Network intrusion detection systems (NIDS) which monitor the network traffics and attempts to discover attempts to break into a system or launching a denial of service attack. An example is a system that monitor large number of TCP connection requests (SYN) to many different ports on a target machine. The unusual large number of requests may suggest attempts of TCP port scan. A NIDS may run either on a target machine which watches its own traffic or on an independent machine promiscuously monitoring all network traffic.

2. System integrity verifiers (SIV) monitors system files to detect intrusions when these files were changed. Currently, "Tripwire" is the most well known SIV system. A SIV may watch components such as Windows registry and configuration in defining well-known signatures. A SIV may also be set to

detect escalation of privileges from a normal user to root/administrator privileges. SIV, hence, is best to detect changes in critical system components. However, SIV does not provide real-time alerts upon an intrusion.

3. Log file monitors (LFM) monitor network services log files. Similar to NIDS, LFM match log files patterns, which may suggest attacks by an intruder. An example is a parser for HTTP server log files that detect intruders who probe known security holes, such as the "phf" attack. These deception systems like `honeypots' contains pseudo-services, which emulate well-known vulnerabilities to trap hackers.

2.3 Pull and push factors for implementing IDS

There several push and pull factors that makes IDS a must, nowadays.

2.3.1 Push factors

The attacks on organization's systems are on the rise and even more pressing in Internet era due to the followings:

1. Attackers become smarter

   Through information sharing, attackers have evolved in executing attacks and network subversion methods. Instead of direct attacks, which will most likely be filtered by firewall, attackers may deploy indirect techniques. These techniques include e-mail based Trojan horses and stealth scanning.

Attackers may also launch tunneling attacks whereby traffic is masked. The mask can be created via encapsulation within packets corresponding to another network protocol, such as ICMP or domain name system (DNS).

2. Vulnerabilities are common

Through information sharing, common vulnerabilities are well known. Attackers may exploit the vulnerabilities arising from system misconfiguration, poorly designed software, user negligence, and well-known protocols and operating systems weaknesses. An example is HTTP, which is normally not filtered by most of the firewall.

Besides that, free Vulnerability Assessment tools like Nessus (http://www.nessus.org) are widely available on the Internet. Script kiddies may make use of these tools to constantly scan the systems facing Internet for known vulnerabilities. Also, professional crackers may also be engaged to break into an organization's networks to steal information or data.

3. Availability of easy-to-use "Hacker" tools

Previously, network scanning and attack techniques are only known to limited computer elites. However recently, easy-to-use "Hacker" tools that can be deployed to conduct sophisticated analysis of a victim network are widely available via Internet. A few example of these tools are SubSeven, BackOrifce, Nmap, L0ftCrack. Attackers without high level of technical knowledge can use these tools to scan, identify, probe, and penetrate an organization's systems.

4. Potential of insider attacks

   Besides attacks from outsiders, insiders also could launch attacks against an organization's systems or carry out malicious activities. The threat is real because insiders are mostly legitimate users who are familiar with the organization's operations and security measures as well as physical access. Perimeter defenses like firewall cannot protect an organization from insider's malicious activities since these activities are executed behind the firewall.

2.3.2 Pull factors

There are valid benefits by deploying IDS ass follows:

1. Deter intruders both outsiders and insiders from attacks

   Through implementation of IDS, intruders have more worry be detected and face punishments for attacks. This can serve as deterrent to intruders.

2. Detect attacks that other security measures cannot prevent

   In reality, some system vulnerabilities may not be tackled due to reasons such as follows:

   - operating systems of legacy systems that cannot be patched or updated.
   - lack of manpower resources to update patches or overlooks by system administrators.

An IDS in detecting attacks will highlight the vulnerabilities to the administrators for corrective actions. Also, IDS provides the trails for monitoring system usage and can be used to detect design flaws and assist in the forensic investigation of intrusions.

3. Early detection of attempts of intrusions

   Intrusions normally involve scanning for information and probing for vulnerabilities in predictable stages. An IDS with proper monitoring could detect such suspicious activities and alert the system administrator to investigate further.

From the above discussions, anti-virus and firewalls are basic defenses against hackers. However, these measures are inadequate to counter attacks that are dynamic and require in-depth network analysis. Thus, an IDS is needed and plays an important role in protecting networks from intrusion both from insiders as well as outsiders. In short, IDS should be deployed and complement other network security measures—firewalls, anti-virus tools, vulnerability assessment products, etc.—as a component of defense in depth. By combining these network security measures, an organization is in a better position to safeguard the information and systems against a realistic range of security attacks.

2.4 IDS monitoring approaches

Broadly, there are 2 IDS monitoring approaches i.e., Host-based Intrusion Detection System (HIDS) and Network-based Intrusion Detection System (NIDS). Each of the approach has strengths and weaknesses.

1. Host-based Intrusion detection

   HIDS uses detection agents to **collect information from host either server or network device**. HIDS agent monitors operating systems, applications and other related processes events and log files as well as tracing problems to individual user. HIDS distributes the load of monitoring across available hosts and operates in encrypted and/or switched network environments. The strengths and weaknesses of HIDS are as shown in Table 2.

Table 2: Strengths and Weaknesses of HIDS

Strengths	Weaknesses
HIDS monitors events local to specific host(s) and hence, capable of tracking behavioral changes that may indicate misuse of the system	More maintenance required since each monitored host requires an agent be installed, configured and maintained.
Better load distribution since HIDS distributes the monitoring load across available hosts on a large network.	HIDS agents consume resources of the host being monitored and may cause performance issue.
HIDS is network independence i.e., unaffected by encrypted/switched network traffic	The required events and log files must be logged correctly for HIDS monitoring.
More accountability since HIDS is able to monitor data access and trace problems to known users.	HIDS may be disabled since the host and network security are maintained by different personnel.
-	HIDS may be attacked and disabled as

	part of an attack on the host like via denial-of-service attacks

2. Network-based Intrusion detection

NIDS agents **collect information from the network or subnets**. NIDS detects attacks at the network level, before the attacks reach the targeted host or applications. The dedicated NIDS sensor deployed on a network segment monitors and analyses every network packet for possible attacks. The strengths and weaknesses of NIDS are as shown in Table 3.

Table 3: Strengths and Weaknesses of NIDS

Strengths	**Weaknesses**
More cost effective since well-placed NIDS sensors can monitor a large network.	NIDS may be overloaded by high traffic volumes and miss an attack.
Minimum impact on operations since NIDS sensors are passive devices without consuming hosts resources.	Switched networks without monitoring or scanning port features may limit the coverage of NIDS sensors. As a result, more sensor need to be deployed.
Stealth and invisible to attackers.	NIDS cannot analyze encrypted network traffic, like VPNs.
Timeliness in detecting attacks and enable prompt corrective actions to be taken.	NIDS cannot determine whether intrusion was successful.

NIDS is platform-independent and is relatively easy to deploy and maintain.	NIDS sensors may generate voluminous data to the management console resulting in latency problems.

2.5 IDS detection approaches

There are 2 main detection approaches i.e., misuse and anomaly detection approaches. Each of the approach has strengths and weaknesses.

1. Misuse intrusions are attacks on known system weaknesses and vulnerabilities. These intrusions can be detected through monitoring and matching of defined pattern or signatures against trails gathered by the IDS sensors or agents. The strengths and weaknesses of misuse detection are as shown in Table 4.

Table 4: Strengths and weaknesses of misuse detection

Strengths	Weaknesses
Misuse detection is more effective without generating a large number of false positives.	Misuse detection is confined to detecting known attacks. Hence, the misuse signatures need to be updated frequently.
Misuse detection is straightforward and specific in diagnosing attacks. This enables administrator to react quickly and correctly.	Tightly defined signatures may render misuse detection fail to detect variants of common attacks.

2. Anomaly intrusions are attacks, which are related to abnormal activities. Hence, the intrusions could be detected through observations of significant deviations from normal system behavior. For this approach, a profile of the system like average CPU usage, average hard disk space and number of processes per user needs to be defined. The metrics for the profile are normally formulated from analysis of cumulated system operation data. The strengths and weaknesses of anomaly detection are as shown in Table 5.

Table 5: Strengths and weaknesses of anomaly detection

Strengths	Weaknesses
Anomaly detection is able to observe abnormal behavior, which may be symptoms of attacks.	Anomaly detection approach is likely to produce a large number of false positives because of unpredictable behaviors of users and networks.
Information gathered through anomaly detection may be used to develop signatures for misuse detection.	Anomaly detection approach requires much more data to define the normal behavior pattern.

2.6 IDS architecture

According to Ant Allan (2002), the main thrust of IDS is automating the intrusion detection process. This is because the network data and/or events and log files involved are voluminous. More so, the detection is rather time sensitive to enable earlier corrective actions and minimize losses or damages. Hence, intrusion detection is much more effective with automation.

An IDS has 3 main functional components i.e.,

- information sources - IDS obtains inputs or event data from one or more information sources like packets that flow through certain monitored network segment.
- analysis - Based on the data gathered, IDS performs a pre-configured analysis of the event data.
- response - The IDS generates specified responses. The responses may include reporting and active intervention like denying access of certain users or data originated from certain IP.

Besides the 3 main components, there is also a IDS management system. This system facilitates a security or network administrator to monitor and configure IDS and analyze data.

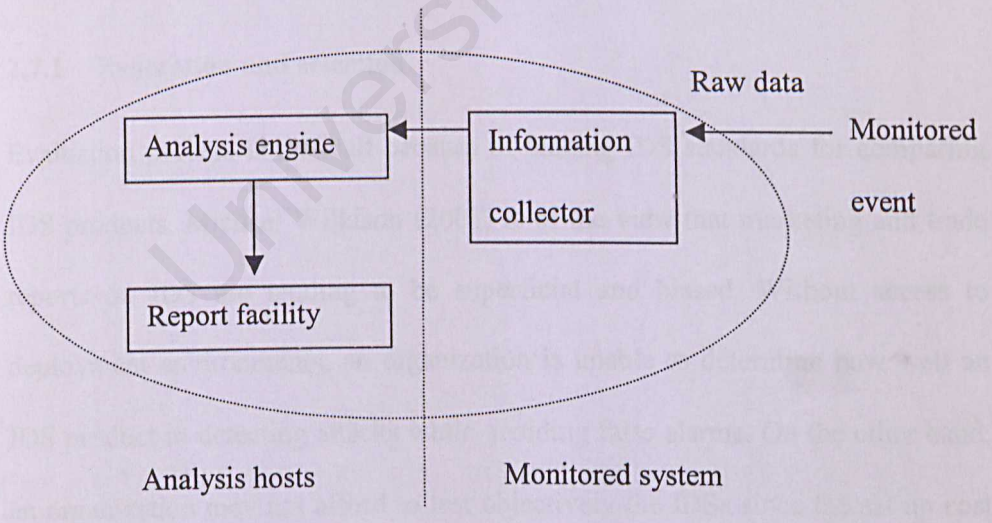


Figure 2 : A generic IDS model

According to Chenxi Wang, John C. Knight (2000), Figure 2 illustrates a generic IDS model adopted in many IDS implementation. In this model, information is gathered from the monitored sources. The information collected is then analyzed for intrusions. The results of the analysis are presented via a report facility either display on the management console or reports.

2.7 The IDS life cycle

With the connection of corporate network to Internet, more new IDS products are introduced to meet the increasing demand of network security. Selecting a right product for implementation is difficult because credible and comprehensive evaluation information is lacking. Also, IDS implementation and maintenance requires specialized skills. Hence, the life cycle of IDS need to be understood. According to John McHugh et al. (2000), the IDS life cycle includes evaluation and selection, deployment, operation and use, and monitoring.

2.7.1 Evaluation and selection

Evaluation process is difficult because of lacking IDS standards for comparing IDS products. Michael Wilkison (2001) is of the view that marketing and trade reports on IDS are tending to be superficial and biased. Without access to deployment environments, an organization is unable to determine how well an IDS product in detecting attacks while avoiding false alarms. On the other hand, an organization may not afford to test objectively the IDSs since the set up cost for testing is too expensive.

G. A. Fink et al. (2002) stated that some work has been done on formal taxonomy of IDSs. However, there is no definitive general accepted metrics on IDS. Ranum (2001) has defined various metrics for evaluating the performance of IDSs.

The key issues to be considered are accuracy of detection, detection and response characteristics, ease of use, user friendliness and supports in signature updates. Edward Amoroso and Richard Kwapniewski (1998) and Ant Allan (2002) have provided guidance in selecting an IDS. The NSS Group has since year 2000 conducted evaluation tests on the major commercial and open source IDS products.

An organization should consider the security needs and the resources available especially the IDS specialist for IDS operation and maintenance. The one selected should meet the organization's needs within the constraints of resources.

2.7.2 Deployment

Deployment process needs to consider the followings:

- Placement of sensors to protect critical assets
- Configure IDS in accordance to the network security policy
- Install appropriate signatures and other initial conditions
- Formulate IDS alerts handling and incident response procedures
- Preserve evidence for possible forensic investigations and prosecutions

2.7.3 Operation and use

Upon implementation, IDS outputs i.e., alerts must be constantly monitored, analyzed and responded to. The monitoring of IDS alerts is performed through a centralized IDS management console. The console is essential because the alerts are normally voluminous which will be summarized and displayed.

2.7.4 Maintenance

Over time, people, technology and processed will change. As and when new IDS or signatures are available, the existing IDS should be upgraded and the new signatures should to be installed. In addition, the IDS deployment including the placement of sensors should be reviewed to ensure the continuous effectiveness of the IDS.

2.8 Evaluation of NIDS

Evaluation of NIDS is the process of benchmarking the NIDS features and capabilities against a set of metrics. . On the contrary, a comparison is a process of 'comparing' two or more products in order to differentiate these products. A proper evaluation process will increase the success of NIDS implementation.

2.8.1 Criteria for evaluating NIDS

The followings are some of the evaluation criteria for NIDS. However, these criteria are dependent on the NIDS's configuration, the network that NIDS is monitoring and the placement of sensors.

1. Ability to detect attacks accurately

   NIDS must be accurate i.e, low false alarm rate. The potential errors of NIDS are false positive and false negative.

   A false positive is an error which NIDS classifies a legitimate action as anomalous action. Excessive false positive errors will tend to cause users to ignore the NIDS outputs. This may lead to an intrusion being detected by NIDS but ignored by the users. Hence, false positive errors should be minimized.

   A false negative is an error, which NIDS fails to recognize an anomalous action. This error is more serious than false positive error because a false negative error gives a misleading sense of security. Hence, NIDS will not be able to highlight a suspicious action to the users.

2. Detecting unknown attacks

   IDS must be able to observe deviations from normal system behavior. This should be the most important NIDS feature because new vulnerabilities are discovered every day. Nevertheless, these new attacks are the most difficult to detect.

3. Stability and reliability

   NIDS must run reliably and continually in the background of the system being monitored without human supervision. NIDS must be fault tolerant i.e., surviving a system crash without losing data and able to rebuild at restart.

The application and operating system should be capable of running for years without segmentation faults or memory leakage.

4. Consistent reporting of identical events

   NIDS should be able to detect known vulnerabilities and report identical events in a consistent manner. Securityfocus and CVE provide databases of known vulnerabilities and exploits.

5. Security

   IDS must be accurate and difficult to fool by subversion error. A subversion error is an error which intruders force IDS to make a false negative error by modifying or blinding the IDS operations. This error is very complex and involves attacks on IDS operations. Hence, IDS seems to be working as intended but actually not. As such, IDS security should be able to withstand all types of attacks. *ADMmutate* and *fragroute* are two programs that are available and claim capable to cause the "*death of the IDS*".

6. Information provided to administrator

   NIDS alerts should provide sufficient information for tracking purposes like the reason the event causing the event to be raised, the source of the alert and the target system. NIDS alerts should also have links vulnerability databases, such as Bugtraq or CVE. The links will facilitate the administrator in assessing the relevance of the alerts and the appropriate reaction to be taken.

7. Severity and potential damage

NIDS should has the ability differentiate the relative importance and severity of attack in the alerts. Some alerts are triggered by information gathering events like port scanning. However, it may not be practical to investigate every time the network is scanned. On the other hand, indication that a local host has been infected by CodeRed should be given higher priority. Snort's alert has been ranged between 1 and 10, which 1 representing a point of interest and 10 representing a major security threat.

8. Legal validity of data collected

NIDS should be able to capture and store sufficient evidence like network traffic or alerts that can be used for forensic investigation and prosecutions. The data collected by NIDS must be legal validity of the data to enable effective legal actions taken against the attacker.

9. Customization

NIDS should be flexible and customizable to monitor the targeted networks. The variation could be running variety of services, different traffic volume and transmission speed at different network segment.

10. Scalability

NIDS must be able to handle changing system behavior, including addition of network segments. As new applications are added, the NIDS must be able to adapt to the changed system profile. The capability should cover the consolidation of reports from multiple NIDS and storage of information.

11. Interoperability

The most effective intrusion detection is correlating information from NIDS, HIDS, system logs, firewall logs and any other sources available. NIDS should be able to communicate with other IDSs of different vendors.

12. Vendor support

NIDS vendor should has the number of qualified staff to support implementing the system.

13. Signature updates

NIDS signature must be available as new vulnerabilities and exploits are discovered. NIDS should allow the administrator to create their own signatures without relying fully on the vendor to supply updates. Alternatively, there should be support from the NIDS community like Snort.

2.9 Selection of IDS product

The NSS Group (2001) (2002) has conducted evaluation tests on the major commercial and open source IDSs annually since year 2000. Michael Wilkison (2001) stated in a SANS paper that the NSS report is the most comprehensive i.e., products coverage and scientific evaluation of IDS. In the year 2002 IDS tests, the NIDS products that have been tested are as follows:

- Cisco Secure IDS 2.5 Model 4230 (formerly known as NetRanger)
- Internet Security Systems RealSecure 7.0
- Snort 1.8.6

These tests cover both qualitative and quantitative analysis. The qualitative analysis on IDS features and functions was performed by IDS specialists. The quantitative analysis involved controlled laboratory tests. The earlier versions of test were conducted fully in laboratory by using traffic generated by Adtech AX/4000 broadband test system and a Smartbits SMB6000. The shortcomings were false alarms and the generated traffic would be differing from the actual attacks. However, in year 2002, the test methodology has brought in real world element i.e., real traffic and real sessions.

These tests include checks on specific IDS performance indicators as follows:

- Test 1: Attack recognition
- Test 2: Stress tests
- Test 3: Ability to resist fools
- Test 4: Stateful operations

A summary of the compiled test results are shown in Table 6, 7, 8 9 and 10.

Table 6: Test 1

Attack Recognition	Attacks	No. of attacks detected by NIDS		
		Cisco	ISS	Snort
Application bugs	5	0	4	4
Back Doors/Trojans/DDOS	11	8	9	11
DOS	16	10	13	10
Finger	7	2	7	6
FTP	11	9	11	6
HTTP	18	11	15	12
ICMP	2	2	0	2
Mail	7	6	6	4
Malicious Data Input	3	0	2	1
Reconnaissance	10	8	8	5
SNMP	2	1	2	0
SANS Top 20 (network-based attacks only)	17	10	17	13
Total	**109**	**67**	**94**	**74**

Table 7: Test 2

Stress tests	NIDS	Network utilization				
		0%	25%	50%	75%	100%
Small (64 byte) packet test (max 148,000pps)	Cisco	100%	100%	100%	100%	100%
	ISS	100%	100%	100%	100%	100%
	Snort	100%	100%	76%	46%	37%
"Real world" packet test (max 40,000pps)	Cisco	100%	100%	100%	100%	97%
	ISS	100%	100%	100%	100%	100%
	Snort	100%	100%	15%	10%	6%
Large (1514 byte) packet test (max 8172pps)	Cisco	100%	100%	100%	100%	100%
	ISS	100%	100%	100%	100%	100%
	Snort	100%	100%	100%	100%	100%

Table 8: Test 3

IDS Evasion Techniques	NIDS	Attacks	Detected?	Decoded?
Fragroute	Cisco	17	13	13
	ISS	17	17	17
	Snort	17	14	14
Whisker	Cisco	9	7	7
	ISS	9	9	9
	Snort	9	6	5
Other techniques	Cisco	9	8	8
	ISS	14	14	14
	Snort	11	9	9

Table 9: Test 4 (a)

Stateful Operation (attack replay)	NIDS	False Pos?	DOS?
Stick	Cisco	Y	N
	ISS	Y	N
	Snort	Y	N
Snot	Cisco	Y	N
	ISS	Y	N
	Snort	Y	N

Table 10: Test 4 (b)

Stateful Operation		**('000)**						
No. of simultaneous open connections	**NIDS**	**10**	**25**	**50**	**100**	**250**	**500**	**1,000**
Detecting attacks	Cisco	Y	Y	Y	Y	Y	Y	Y
	ISS	Y	Y	Y	Y	Y	Y	N
	Snort	Y	Y	Y	Y	Y	Y	Y
Maintaining state	Cisco	Y	Y	Y	Y	N	N	N
	ISS	Y	Y	Y	Y	Y	Y	N
	Snort	Y	Y	N	N	N	N	N

Table 11 : Summary of evaluation

	Criteria	Cisco	ISS	Snort	Notes
0	Cost	√	√	X	*Snort is a freeware*
1	Ability to detect attacks accurately	√	√	√	
2	Detecting unknown attacks	X	√	√	*No anomaly detection feature*
3	Stability and reliability	√	√	√	
4	Consistent reporting of identical events	X	√	√	*Signatures are generic "group". Hence, the reported attack names are very non-specific.*
5	Security (resist evasions)	√	√	√	
6	Information provided to administrator	√	√	√	
7	Severity and potential damage	√	√	√	
8	Legal validity of data collected	X	√	√	*Cisco has no adequate reporting and analysis functions.*
9	Customization	√	√	√	
10	Scalability	√	√	√	
11	Interoperability	√	X	X	*Cisco product is packed with Cisco firewall, router and VPN.*
12	Vendor support	√	√	N/A	
13	Signature updates	√	√	√	*Snort allows users to write new signatures. ISS has included a 'Trons module' to import Snort signatures.*

Based on the evaluation made by NSS as summarized in Table 10, Snort is an open source freeware and free to download from http://www.snort.org. Nevertheless, Snort is compatible to the major NIDS in the market. Thought Snort is lacking of the graphical and user-friendly interfaces, there are full instructions and adequate documentations for installing and configuring Snort.

More so, Snort has a large number of signatures available and the users can develop new signatures or modify signatures to tune for reduced false positives. The users also can download new and latest signatures from http://www.snort.org since there is large support community. For reporting and analysis, there are numerous freeware solutions available like ACID and SnortSnarf.

Hence, this study selects Snort as the NIDS for implementation at the FCSIT networks. At the time of writing, Snort 1.9 is the latest Unix-based of Snort available.

2.10 Summary

This chapter explains the network security attacks and security measures. From there, the needs of IDS as the second line of defense on top of firewall and anti-virus tools are explored.

Various approaches of IDS in monitoring and detecting attacks as well as the corresponding strengths and weaknesses are studied. This chapter also covered the IDS architecture and the life cycle of IDS. The evaluation criterion of IDS is reviewed and the evaluation results released by NSS Group is taken as the basis for selecting Snort for this project.

Chapter 3 Analysis and Design

This chapter is to present the analysis and design of implementing Snort in FCSIT network. The first section will discuss the FCSIT physical and virtual network environment. The network environment encompasses the networks topology, network peripherals deployed and the protocols running. Then, the next section will discuss the Snort configurations and signature updating. Lastly, the chapter will cover the design of Snort implementation in FCSIT networks.

3.1 FCSIT physical networks

Overall, FCSIT networks operate on Fast Ethernet (FE) and Gigabit Ethernet (GE) technology. FCSIT adopted star network topology and the networks are connected via switches and routers. The FCSIT physical networks are shown in Figure 3.

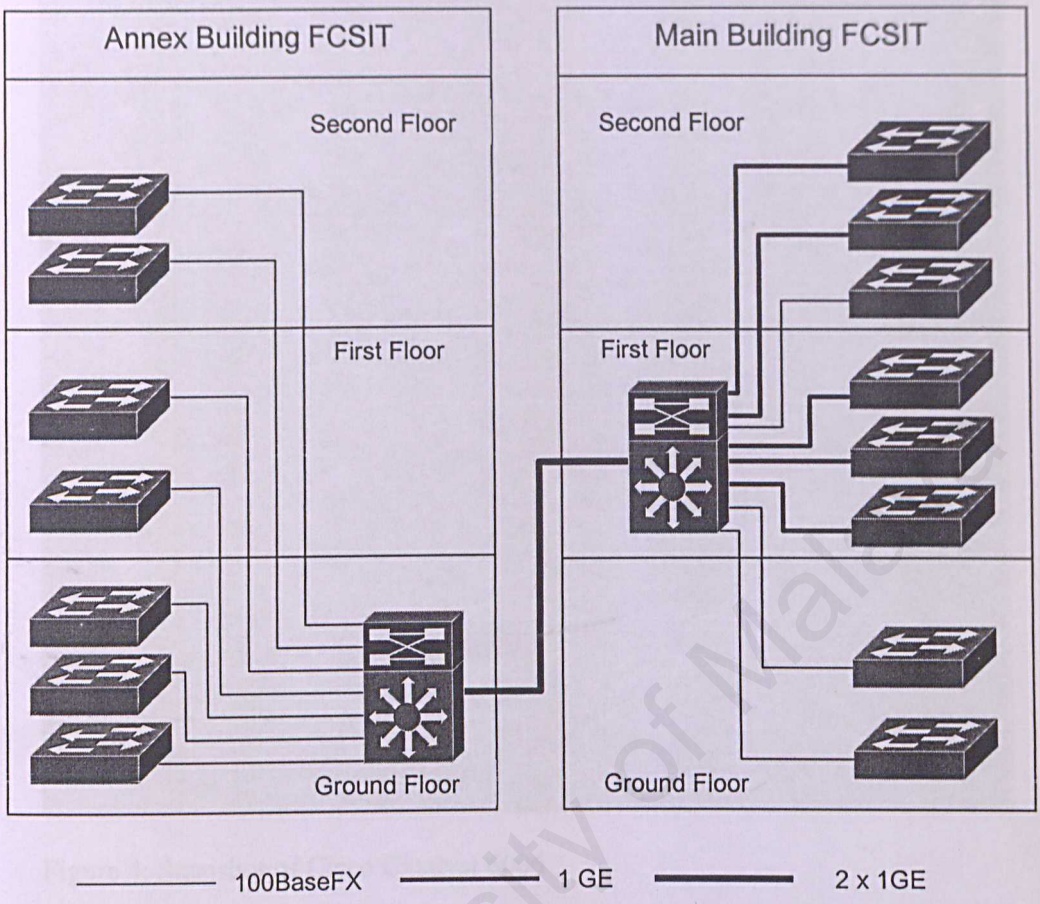


Figure 3 : FCSIT physical networks

Based on Figure 3, the Main Building and Annexed Building are connected through GE. The Main and annex building are connected by two core switches. These are CISCO Catalyst 6006 switches, as shown in Figure 4. Both core switches are equipped with MSFC module which support layer 3 routing. These two core switches are connected through two pairs of multimode Fiber. The total backbone throughput is 2Gbps.

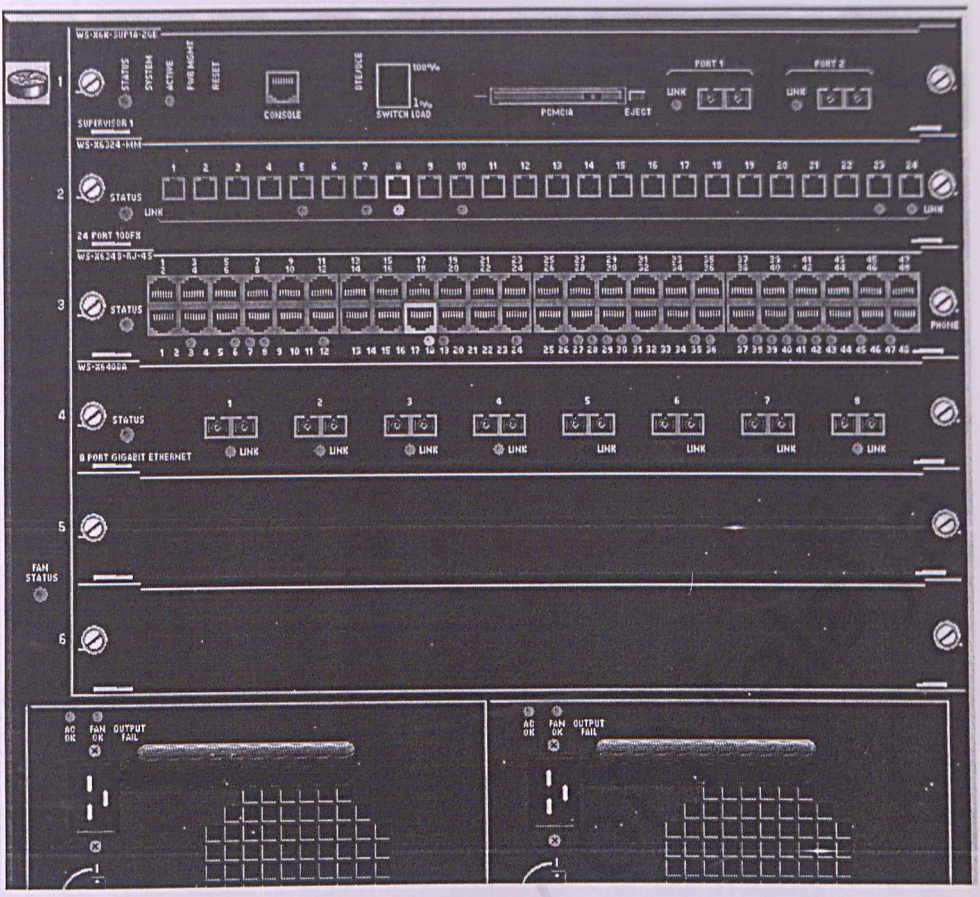


Figure 4: Snapshot of Cisco Catalyst 6006

Access switches connect to the two core switches through FE or GE. These switches are either Cisco Catalyst 2924XL or Catalyst 2950G switches as shown in Figure 5 and 6.

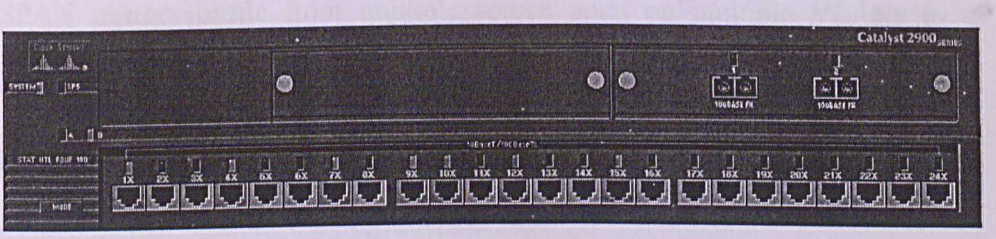


Figure 5: Snapshot of Cisco Catalyst 2924XL

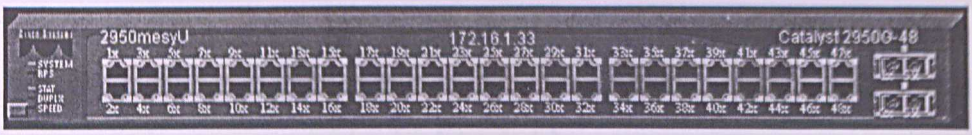


Figure 6: Snapshot of Cisco Catalyst 294450G

3.1.1 Switches and routers

Switches operate in both the physical and data-link layers and divides a large network into smaller segments. The traffics of each segment are separated and switches are able to filter traffics by relaying frames from an originating node of a segment to the recipient at the other segment. The routing between VLANs is done through routers. In a switched network environment, network traffics of different segments are separated since these segments are in different collision domains. This feature will cause Snort unable to sniff traffics from certain network segments.

This weakness in switches for implementing Snort is overcome by built in port monitoring or mirroring function, i.e., Switched Port Analyzer (SPAN). The network traffic that needs to be monitored can be selected by a network analyzer such as a SwitchProbe device or other Remote Monitoring (RMON) probe. SPAN mirrors traffic from multiple source ports on multiple VLANs to a destination port for analysis. However, the source ports and the destination port must be on the same switch.

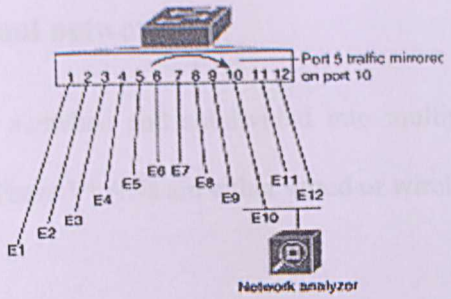


Figure 7: Example SPAN Configuration

In Figure 7 for example, source port is E5 and the traffic is mirrored to port 10. A network analyzer on port 10 receives all network traffic from port 5 without being physically attached to it.

The advantages of span are:

- Easy to implement. Snort can be placed on the switch without modifying the core infrastructure.
- No additional hardware or special configuration changes to manage Snort.

The disadvantages of span are:

- Only one port is available to be spanned at a time. Hence, monitoring of multiple machines can be difficult or impossible.
- According to Arhijit Sarmah (2001), Span more than one port can overload the span port, causing dropped traffic.
- Span will open Snort to attacks. However, this can be rectified by implementing the Snort in stealth mode.
- Degrade switch performance.

3.2 FCSIT virtual network

FCSIT networks are switched and subdivided into multiple virtual local area networks (VLANs). These VLANs are either wired or wireless.

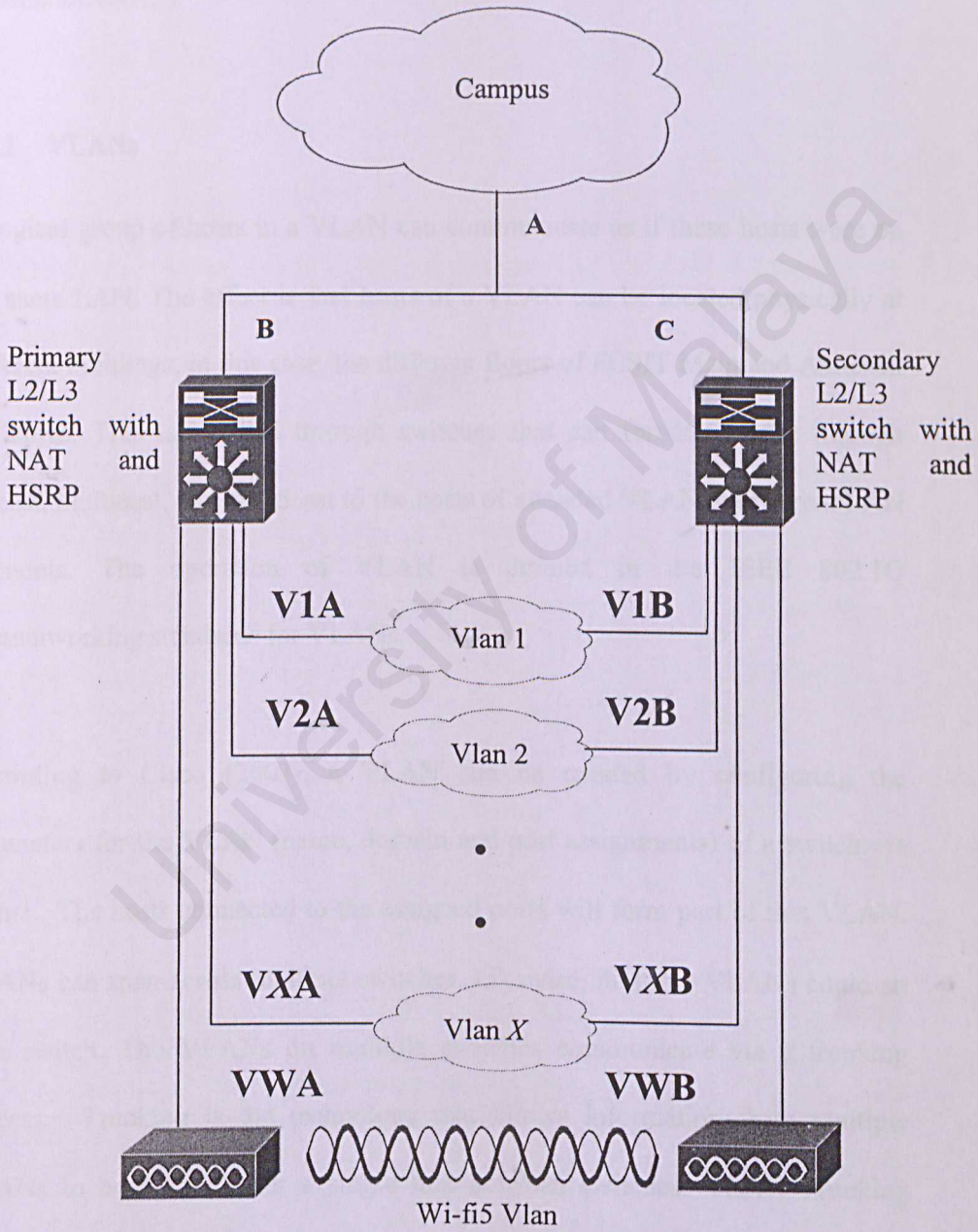


Figure 8: FCSIT virtual networks

Based on Figure 8, the VLANs are connected to the Campus network through two core switches that support layer 3 routing. These two switches are defined as primary and secondary in supporting different user groups. Each of these switches runs Hot Standby Router Protocol (HSRP) and Network Address Translation (NAT).

3.2.1 VLANs

A logical group of hosts in a VLAN can communicate as if these hosts were on the same LAN. The effect is that hosts of a VLAN can be located physically at different buildings, in this case, the different floors of FCSIT Main and Annexed Buildings. This is handled through switches that can forward traffic through unicast, multicast, and broadcast to the hosts of intended VLAN at different LAN segments. The operation of VLAN is defined in the IEEE 802.1Q internetworking standards for VLANs.

According to Cisco (2003), a VLAN can be created by configuring the parameters for the VLAN (name, domain and port assignments) of a switch via Telnet. The hosts connected to the assigned ports will form part of that VLAN. VLANs can span across multiple switches. Likewise, multiple VLANs could sit in a switch. The VLANs on multiple switches communicate via a trunking process. Trunking is the technology that allows information from multiple VLANs to be carried over a single link between switches. VLAN Trunking Protocol (VTP), a layer 2 messaging protocol is the protocol that enables the communication of VLAN configuration among switches. These switches are

managed within VTP domain i.e., network devices that share the same VTP domain name and that are interconnected with trunks.

VLANs are connected to each others through EtherChannel which connected the LAN switches via unshielded twisted-pair (UTP) wiring or single mode and multimode fiber. EtherChannels could be Fast EtherChannel and Gigabit EtherChannel. These EtherChannel allows multiple physical Ethernet links to be combined into one logical channel. This allows load sharing of traffic among the links in the channel as well as redundancy in the event that one or more links in the channel should fail.

In VLANs, VLAN Membership Policy Server (VMPS) assigns switch ports dynamically based on the source MAC address of the device connected to the port. When a host is moved from a port of a switch to a port of another switch, the switch assigns the new port to the proper VLAN for that host dynamically.

Snort can be installed and connected to a destination port to monitor traffics of various switch ports for VLANs. SPAN can be set to monitor the followings:

- Monitor a single source port by using **set span** command and the syntax is: **set span** ***<source port>*** ***<destination port>***.
- Monitor more than one source ports by using **set span** command and the syntax is: **set span** ***<source ports>*** ***<destination port>***.
- Monitor local traffic for an entire VLAN by using **set span** command and the syntax is: **set span** ***<source vlan(s)>***

- Monitor a Trunk i.e., the traffic for all the VLANs on this trunk will be monitored.

3.2.2 HSRP

In order to provide almost full uptime of the routers, Cisco Catalyst 6006 router provides HSRP as one of the redundancy technique. This redundancy enables two or more routers act as a single "virtual" router via sharing of an IP and MAC addresses. These routers exchange status messages periodically. The Active router is enabled by configuring the highest priority value. When a higher priority router preempts a lower priority router, the higher priority Active router will send a coup message. The lower priority active router will revert to speak state and send a resign message upon discover another router with higher priority.

If the Active router ceases functioning, the Standby router can take over the routing responsibility of the Active router. In the case of Standby router failure, another router can be select from a defined group of routers to be the new Standby router. With this, packets can be consistently sent to the same IP and MAC addresses and the take over of router is transparent.

However, the above fail-over service imposes a challenge to Snort implementation. This is because a single attack may flow through different routes and required multi-probe correlation, which is difficult and sometime impossible to reconstruct the attacks. On the other hand, if the assumption is that system failure is unlikely, the challenge will be to ensure that the traffic to/from both

systems is covered. To monitor the traffics of either one of these routers, Snort can be installed at location B or C in the FCSIT virtual networks. However, due to the HSRP, Snort may not be able to capture certain traffics should any one of these routers fails. Hence, location A i.e., the between gateway to Campus network and the two routers will be appropriate to place the Snort. At location A, Snort is able to see all in and out of FCSIT via the Campus network.

3.2.3 Network Address Translation (NAT)

For global communications on Internet, unique global IP addresses registered with IANA are required. Most of the time, the number of global IP addresses is much smaller than the number of hosts that are maintained in an organization's networks. Hence, private IP addresses are used for internal network communications. These hosts however, need to use NAT for communication with external networks via Internet.

NAT can work in the following ways:

1. Static NAT – A local IP address is mapped to a global IP address. This is useful for hosts that require direct access from public networks.
2. Dynamic NAT – A local IP address is mapped to one of a group of global IP addresses.
3. Overloading – Multiple local IP addresses are mapped to a single global IP address by using different port numbers. Hence, this NAT is also known as PAT (Port Address Translation), single address NAT or port-level multiplexed NAT.

For the FCSIT networks, NAT will impact on the analysis of Snort outputs if the traffics behind the core switches are monitored. These locations are either A, B or C. The IP addresses shown in the Snort outputs are global IP addresses and the specific host behind the core switches cannot be determined. Hence, the Snort outputs that involved NAT will require further analysis to gauge the possible VLANs that are affected. Snort will then be installed to monitor the suspected VLANs in order to identify the source hosts.

3.3 Analysis of Snort

Snort as a NIDS focus on performance, simplicity, and flexibility.

3.3.1 Snort Architecture

Snort architecture is made up of three key subsystems namely packet decoder, detection engine, and logging and alerting.

1. Packet Decoder - Supports the Ethernet, SLIP and PPP mediums. The packet decoder prepares the data for the detection engine to perform pattern matching in an expedient manner.
2. Detection Engine - Analyzes every packet based on the Snort rules that are loaded at runtime. The detection engine recursively analyzes every packet based on the loaded Snort rule files. An action specified in the rule definition is triggered as soon as the first rule matches the decoded packet. However, a packet that does not match the Snort rule set is discarded.

3. Logger/Alerter – Makes up of logging and alerting subcomponents. By default, Snort logs are written in the /var/log/Snort folder, and Snort alerts are written to the /var/log/Snort/alerts file.

Nalneesh Gaur (2001) has provided a simplified representation of these components are shown in Figure 9.

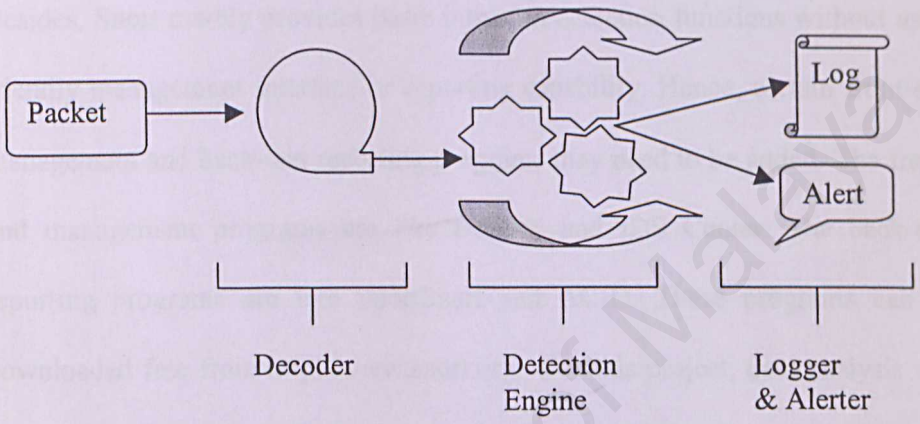


Figure 9 : Snort architecture

3.3.2 Installation

Snort was written for Unix/Linux systems and currently support Windows platform. However for this project, Snort which runs on Unix environment is used. The latest Snort source programs and detection rules can be downloaded free from http://www.snort.org. The installation instructions are also available for download. In addition, more details are also available as comments in Snort source code and configuration files. Snort can be either installed through pre-compile packages or by following the usual configure, make and make install process.

Since Snort is a libpcap-based packet sniffer and logger, libpcap is also required. Libpcap is a library interfacing with the operating system's device drivers to capture packets. Libpcap program can be easily downloaded from Internet through Google Search.

Besides, Snort merely provides basic intrusion detection functions without user-friendly management interface or reporting capability. Hence, certain front-end management and back-end reporting programs may need to be added. The front-end management programs are like Demarc and IDS Center. The back-end reporting programs are like SnortSnarf and ACID. These programs can be downloaded free from http://www.snort.org. For this project, data analysis will be performed via Silicon Defense's perl-based snort log analyzer.

3.3.3 Configuration

Snort can run in one of three modes i.e., sniffer, packet logger and intrusion detection through command line. Among the three modes, intrusion detection is the most complex which match network traffics against user defined rule sets and generates alerts as defined

Snort is configured through the ASCII *snort.conf* file. This file contains several sections i.e., network variable definitions, preprocessors, output modules, log and alert settings, and rule sets to include. This snort.conf file must be configured by using Editor to reflect the local network needs. More detailed options can be found in the Snort Users Manual available in http://www.snort.org.

1. Snort Variables

   Network variables such as HOME_NET and EXTERNAL_NET must be defined to reflect the network topology to be monitored. In a complex environment, defining HOME_NET and EXTERNAL_NET may be difficult. Hence, Sandro Poppi (2002) suggest that these variables can be set to ANY.

2. Snort Preprocessors

   Preprocessors make Snort functionality modular and can be extended through plug-ins. There are nine Snort preprocessors available. These processors are port scan detector, fragmentation reassembly and stateful analysis and BackOrifice detector and traffic normalization for HTTP, RPC, FTP and Telnet sessions. Also, there is SPADE (Statistical Packet Anomaly Detection Engine), a detection engine to perform statistical anomaly detection on the network.

   Preprocessor codes are run after packets have been decoded, but before the detection engine is called. Preprocessors are included and configured using the keyword - preprocessor. The preprocessor directive format in the Snort rules file is:

   preprocessor <name>: <options>

   The more preprocessors loaded will lead to more triggers for alarms. This however will slow down Snort processing performance.

3. Snort Output Modules

Snort output modules are run after the preprocessors and detection engine and the outputs are packet logs and alerts. Alerting will generates both alerts and logs of the same packet. However, logging does not automatically generate an alert. There are twelve Snort alert alerting and logging options. The output processor options set in *snort.conf* can be modified via command line switches.

Alerting can be either "fast", "full" or turned off. Alerts can be written to numerous destinations such as ASCII log files, binary files, syslog, SQL databases, XML files, Unix sockets, SNMP traps or Windows pop-up messages. E-mail alerting can also be included with third party tools. Alerts are written to a single file.

Packet logs can be set to record complete or brief packet contents to the same range of output locations and, like alerts, can be disabled if chosen. However, packet logs are written to individual ASCII files contained within an extended hierarchical directory structure based on source IP address and alert type.

Since writing large numbers of ASCII files into numerous subdirectories is slow, Snort provides a binary logging option to write logs to a *tcpdump* binary file. This file can be analyzed through a back-end analysis tool such as *ACID* or *Snortsnarf*. However, packet logs can also be written to SQL database directly.

4. Snort Rule Sets

Snort rules are stored in ASCII file and can either be created or edited using editor. Snort rules consist of two logical parts: the rule header and options.

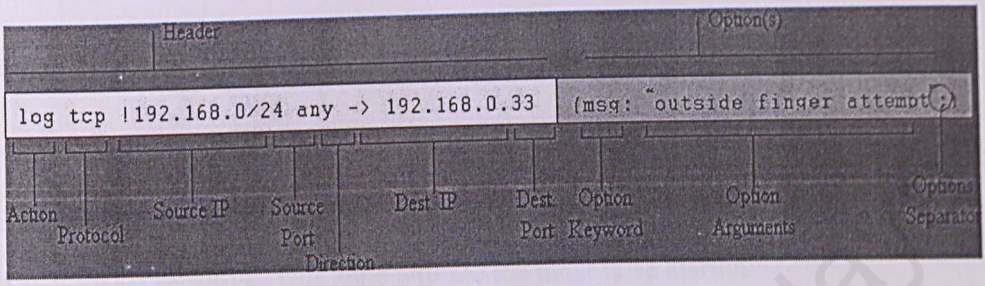


Figure 10: Rule Header and Options Details

From Figure 10, Snort rule is simple and the language is relatively easy to learn. More details on writing rules can be found at http://www.snort.org/writing_snort_rules.htm. In addition, new rules can be found at the arachNIDS database, maintained by Max Vision as well as the Snort Web site. These rules can be incorporated into the snort.conf file by using the "include" directive. When including the new rules, full-qualified pathnames should be used. For example:

include /etc/snort/vision.rules

Since this project uses the shipped Snort, all the rule files and classification.config have been pre-configured and can be included into the Snort configuration. However, the configuration of classification types can be done in /etc/snort/classification.config.

3.4 Implementation design

Design of Snort implementation is largely dependent on FCSIT network environment. However, there are some common locations that NIDS can be installed to monitor network traffics in an organization.

These common logical locations for NIDS are as follows:

- Between internal network and Public network. This is the "red zone" where most of the traffic can be seen. IDS must be configured to be the least sensitive. This zone will see the most false alarms.
- In the DMZ behind the Firewall to identify the attacks on servers in DMZ that have penetrated firewall. This is the "green zone". A well configured firewall only allows known/authorized traffic to enter this zone and less false alarms are expected in this zone
- Between the firewall and internal network to identify attacks that have penetrated the firewall. This is the trusted, "blue zone". Attacks that reaches this zone is considered hostile and require reactions. This zone should have the least number of false alarms.

For the FCSIT network environment, firewall and DMZ are not deployed. However, FCSIT has implemented multiple VLANs supported by Cisco switches and run HSRP for fail over services. Hence, Snort at location A i.e., between the two core switches and the Campus network will be able to see all the inbound and outbound FCSIT traffic.

Nevertheless, Snort may also need to monitor the traffics at various VLANs. This is because FCSIT networks apply NAT and the only way to identify the source host is to span the VLANs or suspected ports. At times, there are also occasions where Snort need to be installed at VLANs to monitor the traffics of internal network to detect insider attacks.

3.5 Summary

This chapter studies the FCSIT network environment for both the physical and virtual networks. The key elements of FCSIT networks like switches, VLAN, HSRP and NAT as well as the impacts on Snort implementation are discussed.

This chapter also analyzes Snort in more details as a NIDS. These details include Snort configurations, signature updates and reporting functions. Lastly, the potential locations for monitoring are discussed.

Chapter 4 Implementation

The chapter presents the implementation of Snort in FCSIT networks. The implementation involved installing Snort as the intrusion detection system. This includes configuring switches to perform ports monitoring and analysis on different VLANs. Lastly, SnortSnarf is used to convert the log file into HTML format.

4.1 Snort installation

Snort is implemented on SUN Solaris platform with two network interface cards (NICs) installed. One of the NIC is used to monitor the network, and the second is used for normal networking purposes. The following details the hardware and software specifications.

4.1.1 Hardware specifications

- SUN Ultra 10
- SUN SparcIIi 445MHz processor
- 1 Gbytes RAM memory
- 2 NIC 10/100 Mbps
- 40 GB hard disk storage

4.1.2 Software specifications

- Solaris 8 (SunOS 5.8) Operating Environment (SPARC platforms)
- Snort 1.9.0

- SnortSnarf 021111.1
- Apache 2.0

4.1.2.1 Solaris 8 (SunOS 5.8)

Solaris 8 is shipped with Practical Extraction and Report Language (Perl) 5.005_03 and Apache web server.

Perl is a free general purpose programming software to develop complex system administration tasks, such as graphic, network, and web programming. Some of the core modules included with this Solaris Perl installation are `CGI`, `NDBM_File`, and `Getopt`.

Apache is an open source HTTP web server and is one of the most popular web servers on the Internet. The shipped version includes all the standard Apache modules, which include proxy server support.

4.1.2.1 Snort installation

Snort 1.9.0 binaries can be downloaded from http://www.snort.org. However, in order for Snort to sniff packets, libpcap is required. Libpcap0.6.2 can be downloaded from http://www.tcpdump.org.

Installing libpcap-0.6.2 by running the following commands (# is the prompt sign) :

```
# tar -zxvf libpcap-0.6.2.tar.gz
# ./configure
# make
# make install
```

Installing Snort by running the following commands (# is the prompt sign) :

```
# tar -zxvf Snort-1.9.0.tar.gz
# ./configure
# make
# make install
```

After finished installing Snort, the Snort.conf file needs to be configured in order to customize the network variables, preprocessors, output options and the rule sets. The four major steps involved are as follows:

Step #1: Set the network variables:

```
var HOME_NET
[10.100.0.0/16,202.185.107.0/24,202.185.108.0/24,202.185.109.0/24
]
var EXTERNAL_NET any
var DNS_SERVERS $HOME_NET
var SMTP_SERVERS $HOME_NET
var HTTP_SERVERS $HOME_NET
var SQL_SERVERS $HOME_NET
var TELNET_SERVERS $HOME_NET
var HTTP_PORTS 80
var SHELLCODE_PORTS !80
var ORACLE_PORTS 1521
var AIM_SERVERS
[64.12.24.0/24,64.12.25.0/24,64.12.26.14/24,64.12.28.0/24,64.12.
29.0/24,64.12.161.0/24,64.12.163.0/24,205.188.5.0/24,205.188.9.0/
24]
var RULE_PATH ../rules
```

Step #2: Configure preprocessors

```
preprocessor frag2
preprocessor stream4: detect_scans, disable_evasion_alerts
preprocessor stream4_reassemble
preprocessor http_decode: 80 unicode iis_alt_unicode
double_encode iis_flip_slash full_whitespace
preprocessor rpc_decode: 111 32771
preprocessor bo: -nobrute
preprocessor telnet_decode
preprocessor asn1_decode
preprocessor conversation: allowed_ip_protocols all, timeout 60,
max_conversations 32000
preprocessor portscan2: scanners_max 3200, targets_max 5000,
target_limit 5, port_limit 20, timeout 60
```

Step #3: Configure output plugins

Set to default

Step #4: Customize rule set

```
include classification.config
include reference.config
include $RULE_PATH/bad-traffic.rules
include $RULE_PATH/exploit.rules
include $RULE_PATH/scan.rules
include $RULE_PATH/finger.rules
include $RULE_PATH/ftp.rules
include $RULE_PATH/telnet.rules
include $RULE_PATH/rpc.rules
include $RULE_PATH/rservices.rules
include $RULE_PATH/dos.rules
include $RULE_PATH/ddos.rules
include $RULE_PATH/dns.rules
include $RULE_PATH/tftp.rules
include $RULE_PATH/web-cgi.rules
include $RULE_PATH/web-coldfusion.rules
include $RULE_PATH/web-iis.rules
include $RULE_PATH/web-frontpage.rules
include $RULE_PATH/web-misc.rules
include $RULE_PATH/web-client.rules
include $RULE_PATH/web-php.rules
include $RULE_PATH/sql.rules
include $RULE_PATH/x11.rules
include $RULE_PATH/icmp.rules
include $RULE_PATH/netbios.rules
include $RULE_PATH/misc.rules
include $RULE_PATH/attack-responses.rules
include $RULE_PATH/oracle.rules
include $RULE_PATH/mysql.rules
include $RULE_PATH/snmp.rules
include $RULE_PATH/smtp.rules
include $RULE_PATH/imap.rules
include $RULE_PATH/pop3.rules
include $RULE_PATH/nntp.rules
include $RULE_PATH/other-ids.rules
# include $RULE_PATH/web-attacks.rules
# include $RULE_PATH/backdoor.rules
# include $RULE_PATH/shellcode.rules
# include $RULE_PATH/policy.rules
# include $RULE_PATH/porn.rules
# include $RULE_PATH/info.rules
# include $RULE_PATH/icmp-info.rules
# include $RULE_PATH/virus.rules
# include $RULE_PATH/chat.rules
# include $RULE_PATH/multimedia.rules
# include $RULE_PATH/p2p.rules
include $RULE_PATH/experimental.rules
include $RULE_PATH/local.rules
```

In order to simplify the execution of Snort at command line, a batch file (snort_command) has been created with the content as follows:

```
./snort -d -i hme1 -l /export/home/idslog -c ../rules/snort.conf
```

The mode of this file has to be changed to executable (# is the prompt sign):

```
# chmod +x snort_command
```

This file has to be placed in the same directory with the Snort executable program. The two NICs are represented with hme0 and hme1. The command above shows that the sniffing process is run on interface hme1 and the log file is stored in `/export/home/idslog` directory. The rule sets in `../rules/snort.conf` are applied.

To run Snort, `./snort_command` is executed and Snort is initiated. The results are as follows :

```
# ./snort_command

Initializing Output Plugins!
Log directory = /export/home/idslog

Initializing Network Interface hme1

        --== Initializing Snort ==--
Decoding Ethernet on interface hme1
Initializing Preprocessors!
Initializing Plug-ins!
Parsing Rules file ../rules/snort.conf

+++++++++++++++++++++++++++++++++++++++++++++++++++
Initializing rule chains...
No arguments to frag2 directive, setting defaults to:
    Fragment timeout: 60 seconds
    Fragment memory cap: 4194304 bytes
    Fragment min_ttl:   0
    Fragment ttl_limit: 5
    Fragment Problems: 0
Stream4 config:
    Stateful inspection: ACTIVE
    Session statistics: INACTIVE
    Session timeout: 30 seconds
    Session memory cap: 8388608 bytes
    State alerts: INACTIVE
    Evasion alerts: INACTIVE
    Scan alerts: ACTIVE
    Log Flushed Streams: INACTIVE
    MinTTL: 1
    TTL Limit: 5
    Async Link: 0
No arguments to stream4_reassemble, setting defaults:
     Reassemble client: ACTIVE
     Reassemble server: INACTIVE
     Reassemble ports: 21 23 25 53 80 143 110 111 513
```

```
     Reassembly alerts: ACTIVE
     Reassembly method: FAVOR_OLD
http_decode arguments:
    Unicode decoding
    IIS alternate Unicode decoding
    IIS double encoding vuln
    Flip backslash to slash
    Include additional whitespace separators
    Ports to decode http on: 80
rpc_decode arguments:
    Ports to decode RPC on: 111 32771
telnet_decode arguments:
    Ports to decode telnet on: 21 23 25 119
Conversation Config:
   KeepStats: 0
   Conv Count: 32000
   Timeout    : 60
   Alert Odd?: 0
   Allowed IP Protocols:  All

1323 Snort rules read...
1323 Option Chains linked into 140 Chain Headers
0 Dynamic rules
+++++++++++++++++++++++++++++++++++++++++++++++++++

Rule application order: ->activation->dynamic->alert->pass->log

        ---== Initialization Complete ==---

-*> Snort! <*-
Version 1.9.0 (Build 209)
By Martin Roesch (roesch@sourcefire.com, www.snort.org)
```

To stop the snort process, press Ctrl-c.

4.1.2.3 SnortSnarf installation

SnortSnarf is a Perl program, which grabs snort's log file and produce HTML web pages for further analysis. The latest copy of SnortSnarf 021111.1 can be downloaded from http://www.silicondefense.com.

SnortSnarf-021111.1 is installed by using the following commands (# is the prompt sign):

```
# tar -zxvf SnortSnarf-021111.1.tar.gz
# perl Makefile.PL
# make
# make install
```

The installed directory of SnortSnarf is in `/export/home/root/SnortSnarf-021111.1`. To view the contents of the directory, the commands below are applied:

```
# cd /export/home/root/SnortSnarf-021111.1
# ls
cgi                          README              snorthtml
Changes                      README.nmap2html    snortsnarf.pl
COPYING                      README.SISR         Time-modules
include                      sisr                Usage
new-annotation-base.xml      snfout.alert        utilities
nmap2html                    snortbatch
```

In order to simplify the execution of SnortSnarf at command line, a batch file (snorthtml) has been created with the content as follows:

```
./snortsnarf.pl /export/home/idslog/alert
```

The mode of this file has to be changed to executable (# is the prompt sign):

```
# chmod +x snorthtml
```

This file has to be placed in the same directory with the SnortSnarf executable program. SnortSnarf's SnortFileInput module reads alerts from Snort's alert output files stored at `/export/home/idslog/alert`.

To run SnortSnarf, `./snorthtml` is executed and SnortSnarf is initiated (# is the prompt sign).

```
# ./snorthtml
```

The outputs of HTML web pages are stored in `snfout.alert` directory. To view the contents of the directory, the commands below are applied (# is the prompt sign):

```
# cd snfout.alert
# ls
202     208         index.html      sig             topsrcs.html
203     209         SDlogo.gif      topdests.html
#
```

The web pages of the alerts are stored into the respective directories such as 202, 203, 208 and 209. The main page is presented `index.html`, which indicates summary of all the signatures. A summary of the top 20 source IPs is presented in `topsrcs.html`. A summary of the top 20 destination IPs is presented in `topdests.html`.

4.1.2.4 Apache installation

Apache is the most popular open-source HTTP web server. Apache 2.0.36 binaries can be downloaded from http://www.apache.org. `snfout.alert` needs to be linked to Apache server default documents directory.

Installing Apache-2.0.36 by running the following commands (# is the prompt sign) :

```
# tar -zxvf apache_2.0.36.tar.gz
# ./configure
# make
# make install
```

The default documents directory for Apache server is in `/var/apache/htdocs`.

The contents of `/var/apache/htdocs` are as follows (# is the prompt sign):

```
# cd /var/apache/htdocs
# ls -l
total 214
-rw-r--r--   1 root  bin   2326 May 16  2001 apache_pb.gif
```

```
dr-xr-xr-x   2 root  other 512 Aug 13  2000 Apps
dr-xr-xr-x   2 root  other 512 Feb  6  2002 Data
-rw-r--r--   1 root  bin   1622 Jan 22  2002 index.html
-rw-r--r--   1 root  bin   1877 May 16  2001 index.html.ca
-rw-r--r--   1 root  bin   1623 May 16  2001 index.html.cz
-rw-r--r--   1 root  bin   2263 May 16  2001 index.html.de
-rw-r--r--   1 root  bin   1550 May 16  2001 index.html.dk
-rw-r--r--   1 root  bin   1870 May 16  2001 index.html.ee
-rw-r--r--   1 root  bin   1358 May 16  2001 index.html.en
-rw-r--r--   1 root  bin   1772 May 16  2001 index.html.es
-rw-r--r--   1 root  bin   1794 May 16  2001 index.html.fr
-rw-r--r--   1 root  bin   1826 May 16  2001 index.html.it
-rw-r--r--   1 root  bin   1623 May 16  2001 index.html.ja.jis
-rw-r--r--   1 root  bin   1887 May 16  2001 index.html.lu
-rw-r--r--   1 root  bin   2010 May 16  2001 index.html.nl
-rw-r--r--   1 root  bin   1951 May 16  2001.index.html.po.iso-pl
-rw-r--r--   1 root  bin   1825 May 16  2001 index.html.pt
-rw-r--r--   1 root  bin   1920 May 16  2001 index.html.pt-br
-rw-r--r--   1 root  bin   1683 May 16  2001 index.html.se
-r--r--r--   1 root  other 15037 Jul 22  1999 ReadMe.html
-r--r--r--   1 root  other 11327 Jul 22  1999 ReadMe.txt
drwxr-xr-x   5 root  other 512 Mar 19 15:01 routersim
lrwxrwxrwx   1 root  other 50 Jan 27 11:53 snort ->
/export/home/root /SnortSnarf-021111.1/snfout.alert
-r--r--r--   1 root  other 35028 Aug 31  1998 splash.JPG
-r--r--r--   1 root  other 2162 Jul 20  1999 start.html
#
```

The snort directory is linked to `/export/home/root/SnortSnarf-021111.1/snfout.alert`.

4.2 Switches configuration

The switches used by FCSIT is supporting multiple VLANs are core switches and access switches. Core switches are Catalyst 6006 and the access switches are either Cisco Catalyst 2924XL or Catalyst 2950G.

4.2.1 Core Switch configuration

The SPAN of core switch is configured through telnet to the Cisco systems console. The command of `set span` is issued to activate the switch analyzer and define both the sources and destination port for monitoring. The sources can be

either a port, a range of ports or VLANs. The direction of traffics to be monitored can be further detailed into transmit (tx), receive (rx) or both.

For the monitoring of VLAN 5 through port 3, the configurations are as follows:

```
Enter password:
PrimaryCat6 enable

Enter password:
PrimaryCat6 (enable) set span 5 3/3

Destination      : Port 3/3
Admin Source     : VLAN 5
Oper Source      : Port 1/1-2,2/5,2/23-24,4/1-4,4/8,15/1
Direction        : transmit/receive
Incoming Packets: disabled
Learning         : enabled
Multicast        : enabled
Filter           : -
Status           : active
```

The results of the implementation of configured monitoring by core switch are as follows:

```
PrimaryCat6 (enable)
2003 May 29 15:43:01 %SYS-5-SPAN_CFGSTATECHG:local span session
active for destination port 3/3
```

To cease the earlier set monitoring by core switch, `set span disable` is issued and the results are as follows:

```
PrimaryCat6 (enable) set span disable
This command will disable all span session(s).
Do you want to continue (y/n) [n]?y
Disabled all local span sessions
PrimaryCat6 (enable) 2003 May 29 15:43:12 %SYS-5-
SPAN_CFGSTATECHG:local span session inactive for destination port
3/3

PrimaryCat6 (enable)
```

4.2.2 Access Switch configuration

The port monitoring session of access switch is configured by defining a destination port with source ports and source VLANs. The monitoring is not activated unless the destination port and at least one source port or VLAN for that session are enabled

To set up a port monitoring session 1, the existing port monitoring session configuration is cleared by setting `no monitor session`. In order to mirror bi-directional traffics to port 1 from port 10 on interface 0, `monitor session` command is used for both source and destination port. The entries are as follows:

```
Switch(config)# configure terminal
Switch(config)# no monitor session 1
Switch(config)# monitor session 1 source interface
fastEthernet0/1
Switch(config)# monitor session 1 destination interface
fastEthernet0/10 encapsulation dot1q
Switch(config)# end
```

To end port monitoring, the existing port monitoring session configuration is cleared by setting `no monitor session`. The entries are as follows:

```
Switch(config)# configure terminal
Switch(config)# no monitor session
Switch(config)# end
```

4.2.3 Snort Implementation Location

The location of implementing Snort is very much dependant on the network segments that need to be monitored. As HSRP, NAT and VLAN design will affect the sniffing of traffics, the Snort implementation location must be correctly identified.

4.2.3.1 HSRP, NAT, VLAN

From the analysis presented in Chapter 3, whenever the core switch is down, HSRP will cause the traffic to be rerouted to the standby switch. If the monitoring is done at the primary core switch, traffics will not be captured. Hence, to monitor the connection between FCSIT network and Campus

backbone, the Snort should be implemented in location A of Figure 8. This location will ensure all inbound and outbound traffics of FCSIT networks are captured.

However, implementing Snort in location A may cause inaccurate results. This is because some of the sources may not be ascertained since NAT has converted the private IP addresses into global IP addresses for outbound traffics. Hence, at times, monitoring at per VLAN basis is also necessary in order to determine the private IP addresses of source.

4.3 Alert file and log files analysis

After configure the switch to monitor the identified traffic, Snort can be activated by issuing `snort_command`. A statistic of the network traffics sniffed and analyzed are as follows:

```
===================================================================
Snort analyzed 915 out of 915 packets, dropping 0(0.000%) packets

Breakdown by protocol:                  Action Stats:
    TCP: 0          (0.000%)            ALERTS: 0
    UDP: 8          (0.874%)            LOGGED: 0
   ICMP: 0          (0.000%)            PASSED: 0
    ARP: 0          (0.000%)
  EAPOL: 0          (0.000%)
   IPv6: 0          (0.000%)
    IPX: 9          (0.984%)
  OTHER: 898        (98.142%)
DISCARD: 0          (0.000%)
===================================================================
Wireless Stats:
Breakdown by type:
    Management Packets: 0           (0.000%)
    Control Packets:    0           (0.000%)
    Data Packets:       0           (0.000%)
===================================================================
Fragmentation Stats:
Fragmented IP Packets: 0          (0.000%)
    Fragment Trackers: 0
   Rebuilt IP Packets: 0
   Frag elements used: 0
Discarded(incomplete): 0
```

```
   Discarded(timeout): 0
  Frag2 memory faults: 0
===================================================================
TCP Stream Reassembly Stats:
        TCP Packets Used: 0          (0.000%)
         Stream Trackers: 0
          Stream flushes: 0
           Segments used: 0
   Stream4 Memory Faults: 0
===================================================================
Snort received signal 2, exiting
#
```

For the events when the monitored traffics match with the applied rule sets, alerts are initiated and are stored in `alert` file. The events are also logged and the details are stored into the individual log file, which identified by source IP address. For example, one of the source IP is 202.185.107.204. The log files and alert file generated are stored at `/export/home/idslog` directory. The contents of `/export/home/idslog` are as follows (# is the prompt sign):

```
# cd /export/home/idslog
# ls
202.100.169.3     202.185.107.73    202.88.151.73     alert
202.185.107.204   202.185.109.30    202.94.160.103
202.185.107.69    202.54.38.108     202.97.244.31
```

Alerts are the records indicating a match with the rule sets. The contents of the alerts are issued according to time sequence. The alerts indicate both the source and destination IPs and ports. The events are classified and prioritized according to the applied definitions in Snort `classification.conf`. To facilitate the analysis of alerts, references such as Snort Signature Database (SID), SecurityFocus' Bugtraq, Common Vulnerabilities and Exploits (CVE) and Arachnids are also indicated. The contents of the `alert` file are as follows (# is the prompt sign):

```
# more alert
[**] [1:1560:4] WEB-MISC /doc/ access [**]
[Classification: access to a potentially vulnerable web
application][Priority:2]
```

```
05/26-15:13:47.396888 202.185.107.204:2795 -> 209.202.218.12:80
TCP TTL:127 TOS:0x0 ID:40392 IpLen:20 DgmLen:326 DF
***AP*** Seq: 0x6AC3F692  Ack: 0x362D8388  Win: 0xFAF0 TcpLen: 20
[Xref => bugtraq 318][Xref => cve CVE-1999-0678]

[**] [1:1560:4] WEB-MISC /doc/ access [**]
[Classification: access to a potentially vulnerable web
application][Priority:2]
05/26-15:13:47.832807 202.185.107.204:2801 -> 209.202.218.10:80
TCP TTL:127 TOS:0x0 ID:40411 IpLen:20 DgmLen:366 DF
***AP*** Seq: 0x6AC7D630  Ack: 0x395C51BF  Win: 0xFAF0 TcpLen: 20
[Xref => bugtraq 318][Xref => cve CVE-1999-0678]
```

•
•
•

```
[**] [1:895:5] WEB-CGI redirect access [**]
[Classification: Attempted Information Leak] [Priority: 2]
05/26-15:15:26.063891 202.185.107.204:2891 -> 208.185.211.71:80
TCP TTL:127 TOS:0x0 ID:41213 IpLen:20 DgmLen:446 DF
***AP*** Seq: 0x6C74FBE1  Ack: 0xC295E29E  Win: 0xFFFF  TcpLen:
20 [Xref => cve CVE-2000-0382][Xref => bugtraq 1179]

[**] [1:862:6] WEB-CGI csh access [**]
[Classification: Attempted Information Leak] [Priority: 2]
05/26-15:15:40.715339 202.185.109.30:1953 -> 203.166.138.124:80
TCP TTL:127 TOS:0x0 ID:18757 IpLen:20 DgmLen:492 DF
***AP*** Seq: 0x4D3193C  Ack: 0xC9658E8B  Win: 0x40B0  TcpLen: 20
[Xref => cve CAN-1999-0509][Xref => url
www.cert.org/advisories/CA-1996-11.html]
#
```

From the above details of alerts, the first of an alert show the name of the signature that triggers an alert. The second line of an alert indicates the classification of signature and the priority set. The third line of an alert shows the date and time of alerts generated. The source IP and destination IP as well service ports are also specified. The fourth and fifth line of an alert presents the packet header information. The last line of an alert shows the references such as CVE, SID and Bugtraq to facilitate understanding of the signature and attacks.

Log files include the packet information. The contents of `202.185.107.204` log file are as follows (# is the prompt sign):

```
# cd 202.185.107.204
# ls
TCP:2795-80  TCP:2810-80  TCP:2843-80  TCP:2891-80
TCP:2801-80  TCP:2842-80  TCP:2868-80
# more  TCP:2891-80
[**] WEB-CGI redirect access [**]
05/26-15:15:26.063891 202.185.107.204:2891 -> 208.185.211.71:80
TCP TTL:127 TOS:0x0 ID:41213 IpLen:20 DgmLen:446 DF
***AP*** Seq: 0x6C74FBE1  Ack: 0xC295E29E  Win: 0xFFFF  TcpLen:
20
47 45 54 20 2F 70 6F 70 73 2F 65 5A 75 6C 61 50  GET /pops/eZulaP
6F 70 53 63 72 69 70 74 32 2E 61 73 70 3F 44 53  opScript2.asp?DS
5F 49 44 3D 33 30 36 30 36 33 26 44 6F 6D 61 69  _ID=306063&Domai
6E 3D 77 77 77 2E 70 61 6E 69 73 6C 61 6D 69 63  n=www.panislamic
2E 63 6F 6D 26 50 6F 73 3D 4C 69 73 74 31 38 26  .com&Pos=List18&
57 3D 32 37 35 26 48 3D 33 33 30 26 55 31 3D 68  W=275&H=330&U1=h
74 74 70 3A 2F 2F 77 77 77 2E 65 7A 75 6C 61 2E  ttp://www.ezula.
63 6F 6D 2F 65 5A 75 6C 61 2F 72 65 64 69 72 65  com/eZula/redire
63 74 2F 72 65 64 69 72 65 63 74 2E 61 73 70 3F  ct/redirect.asp?
44 53 5F 49 44 3D 33 30 36 30 36 33 26 55 32 3D  DS_ID=306063&U2=
68 74 74 70 3A 2F 2F 77 77 77 2E 65 7A 75 6C 61  http://www.ezula
2E 63 6F 6D 2F 70 6F 70 73 2F 69 6D 61 67 65 73  .com/pops/images
2F 32 37 35 78 33 33 30 42 6C 75 65 53 74 72 69  /275x330BlueStri
70 65 2E 67 69 66 20 48 54 54 50 2F 31 2E 31 0D  pe.gif HTTP/1.1.
0A 41 63 63 65 70 74 3A 20 2A 2F 2A 0D 0A 41 63  .Accept: */*..Ac
63 65 70 74 2D 4C 61 6E 67 75 61 67 65 3A 20 65  cept-Language: e
6E 2D 75 73 0D 0A 41 63 63 65 70 74 2D 45 6E 63  n-us..Accept-Enc
6F 64 69 6E 67 3A 20 67 7A 69 70 2C 20 64 65 66  oding: gzip, def
6C 61 74 65 0D 0A 55 73 65 72 2D 41 67 65 6E 74  late..User-Agent
3A 20 4D 6F 7A 69 6C 6C 61 2F 34 2E 30 20 28 63  : Mozilla/4.0 (c
6F 6D 70 61 74 69 62 6C 65 3B 20 4D 53 49 45 20  ompatible; MSIE
35 2E 30 31 3B 20 57 69 6E 64 6F 77 73 20 4E 54  5.01; Windows NT
20 35 2E 30 29 0D 0A 48 6F 73 74 3A 20 77 77 77   5.0)..Host: www
2E 65 7A 75 6C 61 2E 63 6F 6D 0D 0A 43 6F 6E 6E  .ezula.com..Conn
65 63 74 69 6F 6E 3A 20 4B 65 65 70 2D 41 6C 69  ection: Keep-Ali
76 65 0D 0A 0D 0A                                ve....
```

4.4 Summary

This chapter studies the implementation of Snort in FCSIT networks. The study details the steps of installation and the configurations of hardware and software as well as network devices involved to effect the intrusion detection. Some of the results of the implementation of Snort in FCSIT networks are also included and explained.

Chapter 5 Analysis of results

The chapter presents the analysis of results for the implementations of Snort in FCSIT networks. Based on the analysis, Snort is fine tuned to minimize false alarms and improve detection of intrusions.

5.1 Analysis of results – FCSIT network monitoring

SnortSnarf provides analysis of Snort log files and alert file by signatures, source IPs and destination IPs. Snort was implemented on Location A as shown in Figure 8. 4,782 alerts were captured on March 25, 2003 between 10:34 and 11:15. These alerts originated from 332 source IPs targeted 645 destination IPs. A summary of full details of these signatures, top 20 source IPs and destination IPs is tabled in Appendix A. SnortSnarf has provided with the references such as Snort Signature Database (SID), CERT's CVE and SecurityFocus' Bugtraq to understand and analyze the alerts.

Snort signatures are classified and prioritized to facilitate analysis. The severity of the signature's priority by default as set by Snort is 1, 2 or 3 and this priority can be edited in the `classification.conf`. The meaning of the priority setting is as follows:

- 1 - alerts about traffic that is definitely hostile
- 2 - alerts about benign normal network traffic
- 3 - alerts that fall in the middle, or might be hostile sometimes, benign others.

4,782 alerts are made up of 46 signatures. For each of the signature, there is a link to the summary, which indicates all the source IPs and destination IPs relating to that signature.

Table 12 : Analysis of all Snort signatures

Priority	No. of signatures	No. of alerts	% of alerts
1	14	706	14.76
2	27	3,219	67.31
3	4	764	15.98
n/a	1	93	1.94
	46	4,782	100.00

5.1.1 Analysis of Snort alerts with priority 1

There are 706 alerts make up of 14 signatures with priority 1. The top 3 signatures made up of 94% of the alerts of priority 1. These 3 signatures are WEB-IIS ISAPI .ida attempt (48%), WEB-IIS cmd.exe access (43%) and POP 3 PASS overflow attempt (3%). The balance of 11 signatures contributes to 6% of the alerts with priority 1. A summary of alerts for signatures with priority 1 is depicted in Figure 11.

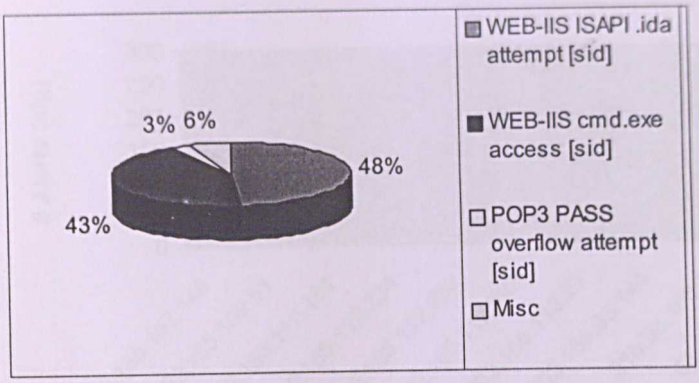


Figure 11 : Alerts due to signature with priority 1

5.1.1.1 WEB-IIS ISAPI .ida attempt

This signature is related to CodeRed attacks. Microsoft IIS 4.0 and IIS 5.0 allow a remote attacker to obtain the pathname of the document root by requesting non-existent files with .ida or .idq extensions. This vulnerability existed in the indexing service used by Microsoft IIS 4.0 and IIS 5.0 running on Windows NT,Windows 2000, and beta versions of Windows XP.

There are 346 alerts issued arising from the WEB-IIS ISAPI .ida attempt. 25 source IPs and 266 destination IPs are involved. Out of the 346 alerts, 261 alerts are attributable to 202.185.107.146. The top 10 source IPs causing the alerts is shown in Figure 12.

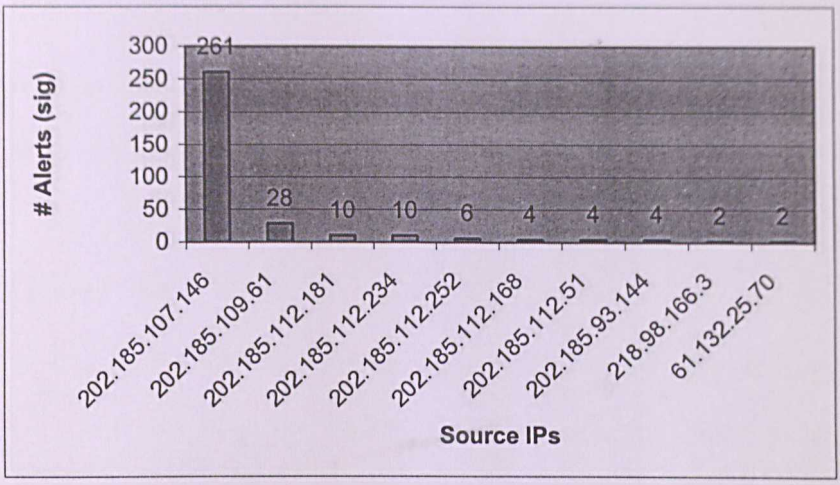


Figure 12 : Top 10 source IPs causing alert on WEB-IIS ISAPI .ida attempt

5.1.1.2 WEB-IIS cmd.exe access

This signature is closely related to CodeRed attacks. In CodeRed attacks, a backdoor is installed into affected systems by copying the standard Windows NT/2000 command interpreter "cmd.exe" into web server's "scripts" directory. This backdoor will allow any web surfer to execute commands on the infected web site by typing related URLs to the web location.

There are 302 alerts issued arising from the WEB-IIS cmd.exe access. 26 source IPs and 262 destination IPs are involved. Out of the 346 alerts, 213 alerts are attributable to 202.185.107.146. The top 10 source IPs causing the alerts is shown in Figure 13.

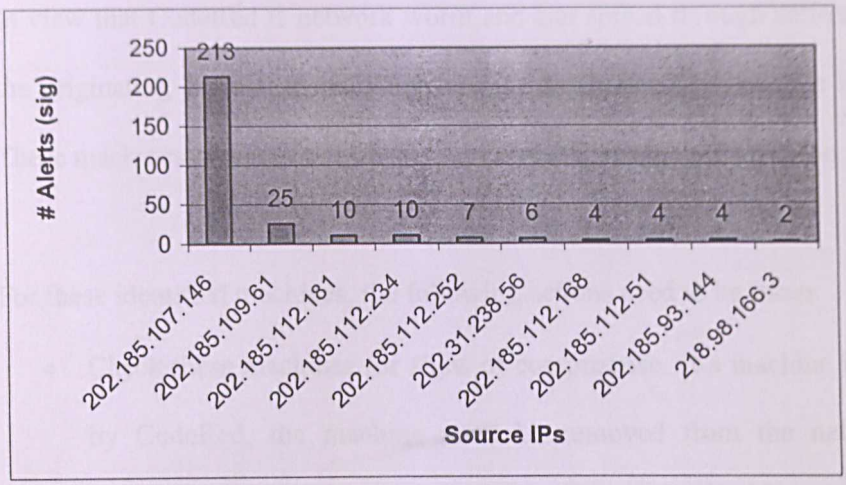


Figure 13 : Top 10 source IPs causing alert on WEB-IIS cmd.exe access

5.1.1.3 WEB-IIS CodeRed v2 root.exe access

WEB-IIS CodeRed v2 root.exe access involves attempts to access the root.exe executable on a webserver. Normally, root.exe access is detected as part of an attempted infection by a CodeRed infected machine.

There is one source IP and destination IP that are involved i.e., the source IP is 202.31.238.56 and the destination IP is 202.185.109.22. Destination IP 202.185.109.22 also suffers from WEB-IIS ISAPI .ida attempt and WEB-IIS cmd.exe access.

5.1.1.4 Corrective actions

From the analysis of the source IPs, both the WEB-IIS ISAPI .ida attempt and WEB-IIS cmd.exe access are related to CodeRed attacks. More so, the source IPs for the causing alerts on WEB-IIS ISAPI .ida attempt and WEB-IIS cmd.exe access are identical. Hence, the attacks are likely CodeRed attacks.

In view that CodeRed is network worm and can spread through self-replication, the originating and target machines need to be checked for possible infections. These machines are identified based on both the source and destination IPs.

For these identified machines, the following actions need to be taken:

- Check these machines for signs of compromise. If a machine is infected by CodeRed, the machine must be removed from the network and cleaned.
- Rectify the configuration error to avoid WEB-IIS ISAPI .ida attempt by not serving .ida or .idq files from a network share.
- Check and ensure that the IIS implementation is fully patched.
- Check and ensure that the underlying operating system is fully patched.

5.1.2 Analysis of Snort alerts with priority 2

There are 3,219 alerts make up of 27 signatures with priority 2. The top 3 signatures contribute to 90% of the alerts of priority 2. These signatures are ICMP redirect net (55%), ICMP redirect host (26%) and ICMP PING NMAP (9%). A summary of alerts for signatures with priority 2 is depicted in Figure 14.

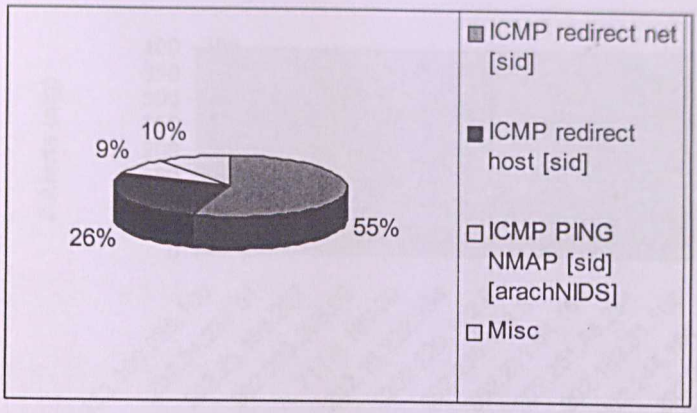


Figure 14 : Alerts due to signature with priority 2

5.1.2.1 ICMP Redirect Network

ICMP Redirect Network packets may be sent in certain normal situations to save network administrators from having to keep extensive routing tables on hosts. However, an attacker may send corrupted ICMP Redirect Net message (Type 5, Code 0) to a network trying to crash a system. This, nevertheless, may also generate false alarms since any ICMP Redirect Network will trigger alerts.

There are 1,780 alerts issued arising from the ICMP redirect net. 14 source IPs and 2 destination IPs are involved. None of the source IPs are FCSIT's while the 2 destination IPs are 202.185.107.146 and 202.185.109.61. The distribution of the alerts is shown in Figure 15 and 16.

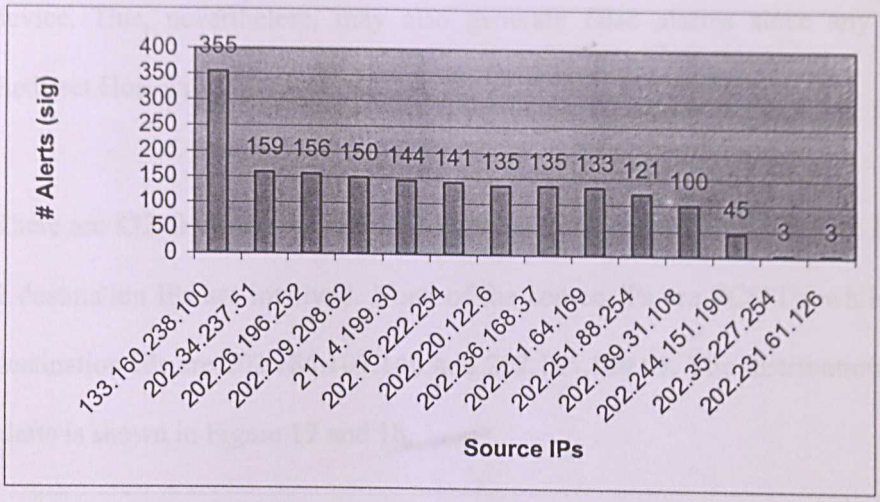


Figure 15 : Source IPs that trigger alerts of ICMP Redirect Network

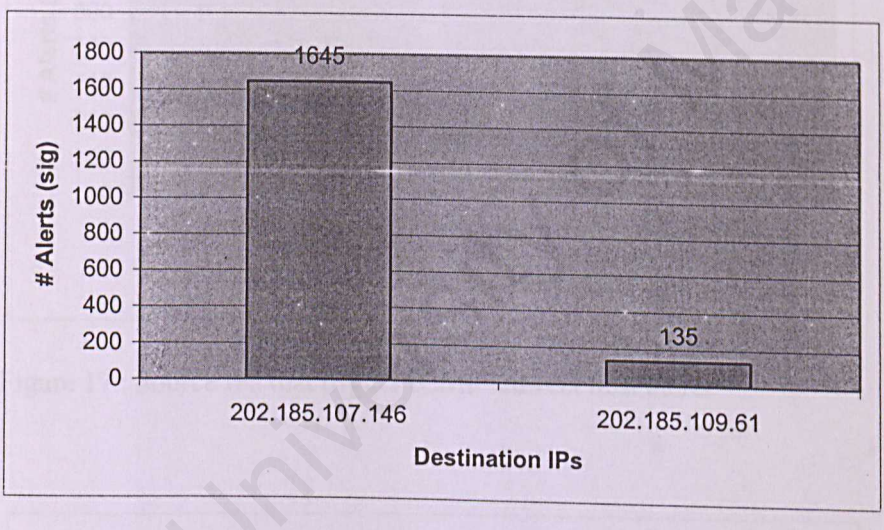


Figure 16 : Destination IPs that received the ICMP Redirect Network message

5.1.2.2 ICMP Redirect Host

Normally, gateway devices generate an ICMP Redirect Host datagram if there is a shorter route to a particular destination exists. The ICMP Redirect message is sent back to the host that originally generated the traffic. An attacker may ICMP Redirect Host message to force hosts to use a compromised gateway

device. This, nevertheless, may also generate false alarms since any ICMP Redirect Host will trigger alerts.

There are 833 alerts issued arising from the ICMP redirect host. 3 source IPs and 2 destination IPs are involved. None of the source IPs are FCSIT's while the 2 destination IPs are 202.185.107.146 and 202.185.109.61. The distribution of the alerts is shown in Figure 17 and 18.

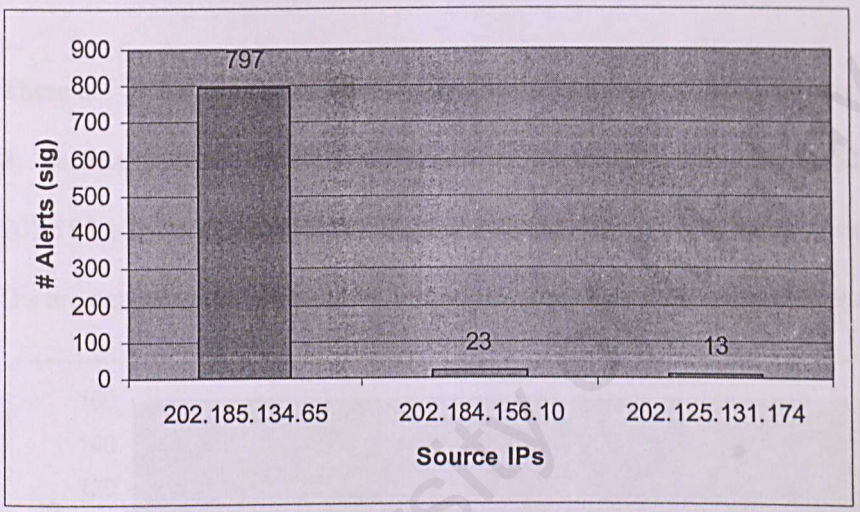


Figure 17 : Source IPs that trigger ICMP redirect host alerts

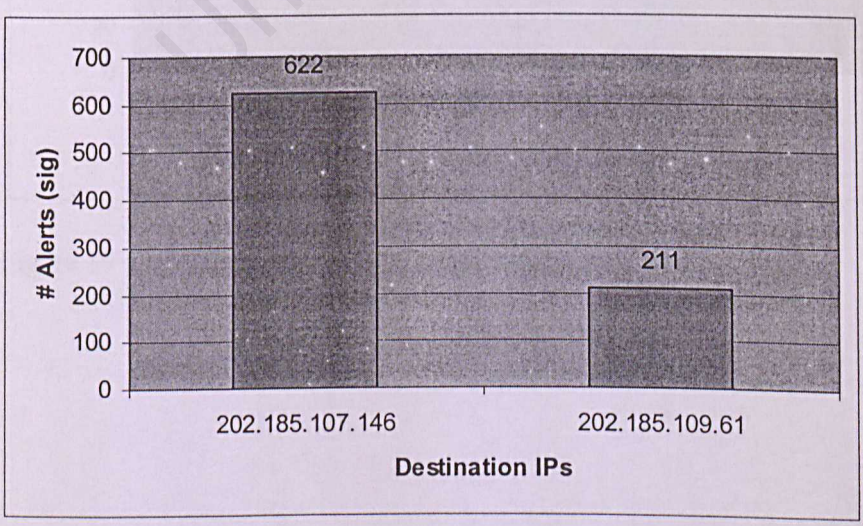


Figure 18 : Destination IPs that received ICMP redirect host message

5.1.2.3 ICMP PING NMAP

A user with root access may send ICMP ping through NMAP to ascertain whether the host is alive. NMAP's ICMP ping, by default, sends zero data as part of the ping. This ICMP ping may be part of an attacker's information gathering process. This, nevertheless, may also generate false alarms since any NMAP's ICMP ping will trigger alerts.

There are 278 alerts issued arising from the ICMP redirect host. 3 source IPs and 2 destination IPs are involved. The 3 source IPs are FCSIT's i.e., 202.185.109.169, 202.185.109.12 and 202.185.109.223 while the 2 destination IPs are external IPs. The distribution of the alerts is shown in Figure 19 and 20.

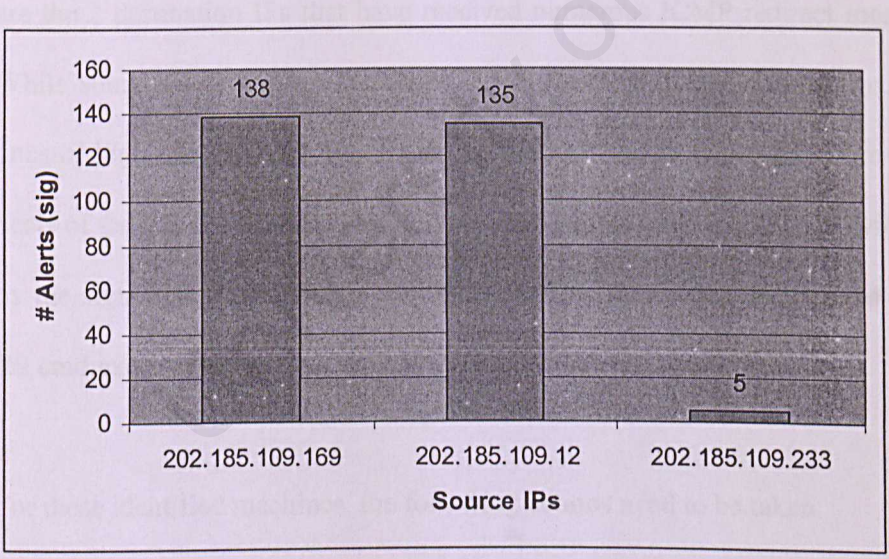


Figure 19 : Source IPs that trigger ICMP ping

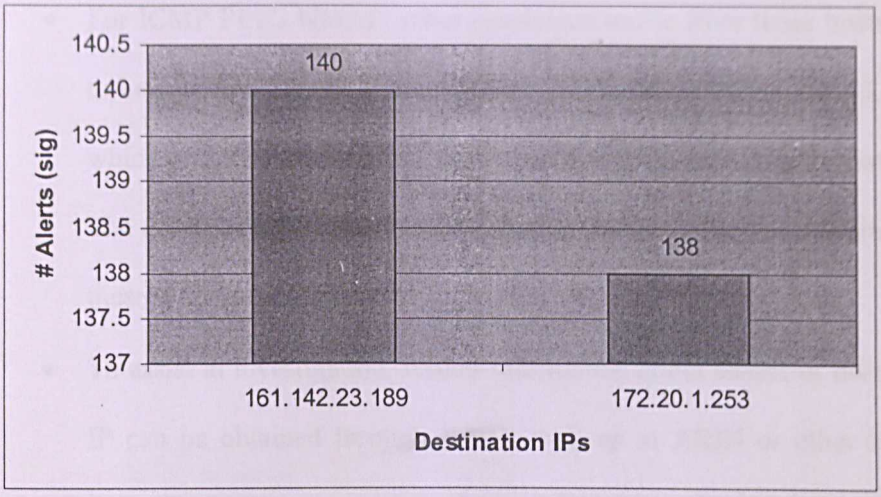


Figure 20 : Destination IPs that received ICMP ping

5.1.2.4 Corrective actions

From the analysis of the destination IPs, 202.185.107.146 and 202.185.109.61 are the 2 destination IPs that have received numerous ICMP redirect messages. While some ICMP redirect messages may be valid in certain situations, these unusual high numbers of messages tend to suggest malicious traffic. More so, the hosts of these 2 IPs are also likely to be infected by CodeRed. This is evidenced by the significant higher incidences of WEB-IIS ISAPI .ida attempt and WEB-IIS cmd.exe access.

For these identified machines, the following actions need to be taken:

- For ICMP Redirect Network, SP4 patches for Microsoft Windows NT 4.0 need to be updated on the affected machines.
- Ingress filtering should be utilized to deny incoming ICMP Type 5 datagrams.

- For ICMP PING NMAP, other suspicious traffic from these hosts needs to be assessed for possible malicious attacks. Except for 202.185.109.12 which could be affected by CodeRed and need to be further investigated, The traffic of other 2 source IPs seems normal and the alerts arising from these 2 IPs may be ignored.
- To assist in investigation, further information about source or destination IP can be obtained through Whois look up at ARIN or other look up links. Figure 21 is a snapshot of the ARIN look up for 133.160.238.100.

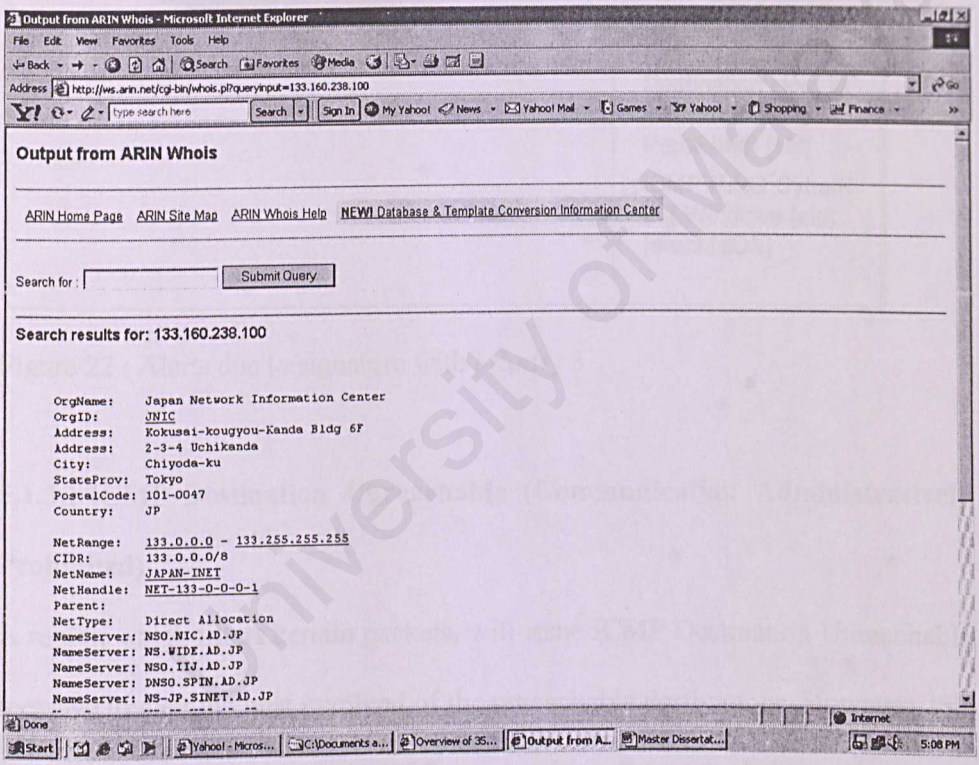


Figure 21 : Snapshot of ARIN look up on 133.160.238.100

5.1.3 Analysis of Snort alerts with priority 3

There are 764 alerts make up of 4 signatures with priority 3. The top 2 signatures made up of 99.08% of the alerts of priority 3. These signatures are ICMP destination unreachable (communication administratively prohibited) (92%) and

ICMP PING speedera (7%). A summary of alerts for signatures with priority 2 is depicted in Figure 22.

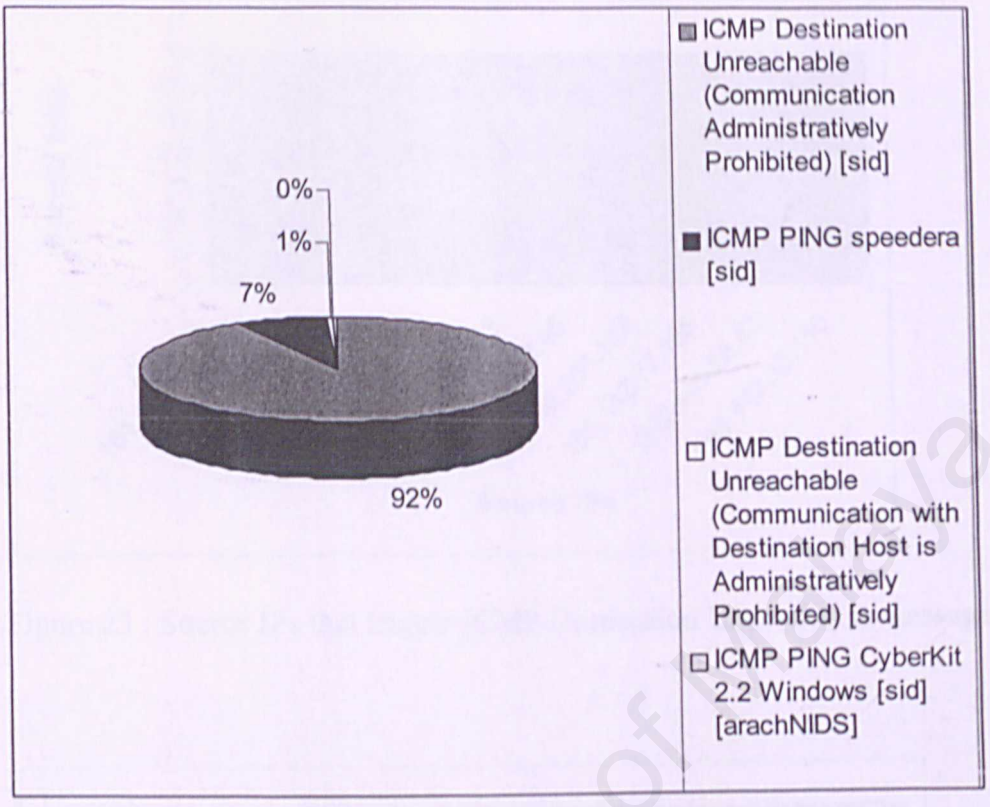


Figure 22 : Alerts due to signature with priority 3

5.1.3.1 ICMP Destination Unreachable (Communication Administratively Prohibited)

A router, which filters certain packets, will issue ICMP Destination Unreachable message to alert the host involved of the unreachable destinations. However, this message may be attributable to DOS attack arising from spoofed source address. Also similar situation may occur when a large portscan is launched to mask the true source of the scan. This, nevertheless, may generate false alarms.

There are 701 alerts issued arising from the ICMP Destination Unreachable (Communication Administratively Prohibited). 157 source IPs and 2 destination IPs are involved. None of the source IPs are FCSIT's while the 2 destination IPs

are 202.185.107.146 and 202.185.109.61. The distribution of the alerts is shown in Figure 23 and 24.

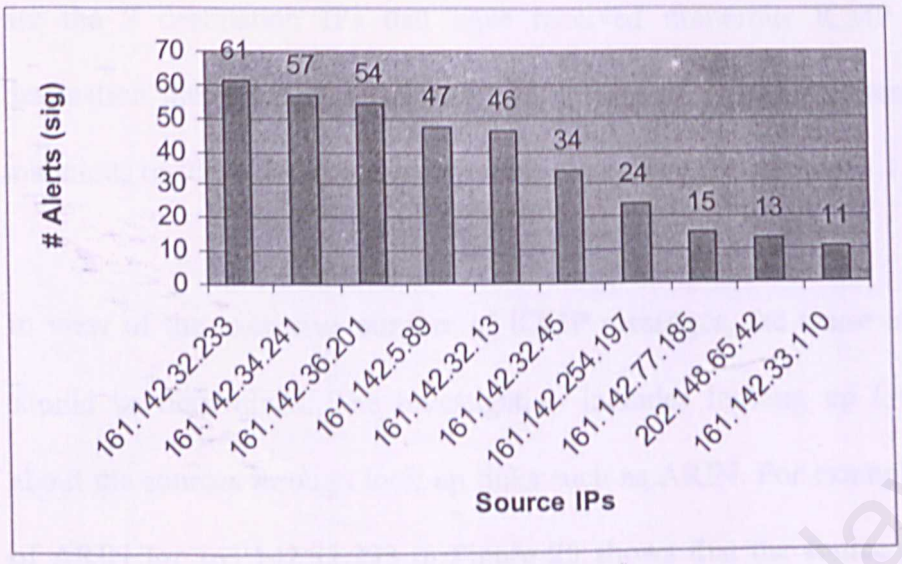


Figure 23 : Source IPs that trigger ICMP Destination Unreachable message

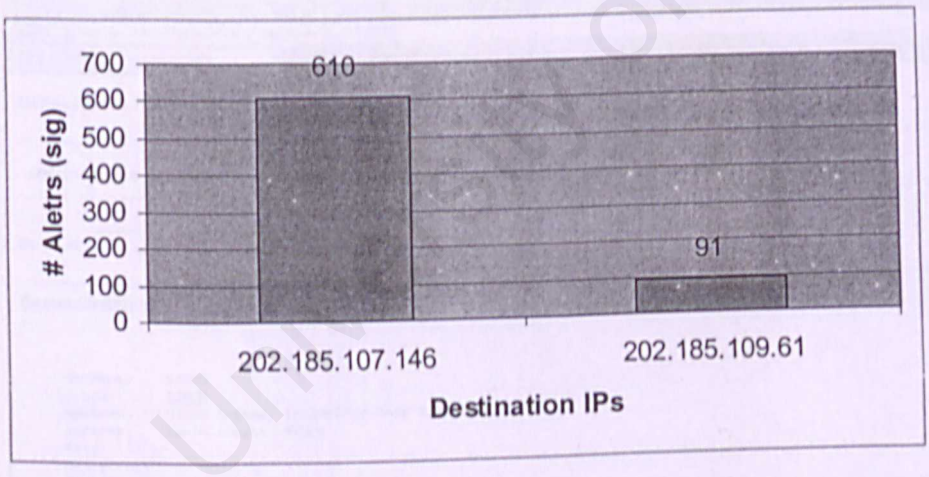


Figure 24 : Destination IPs that received ICMP Destination Unreachable message

5.1.3.2 ICMP PING speedera

SpeedEra.net uses ICMP ping to find closest cache to certain hosts. This is normally not an attack. However, an attacker could disguise hostile pings as speedera pings. This normally generates false alarms.

5.1.3.3 Corrective actions

From the analysis of the destination IPs, 202.185.107.146 and 202.185.109.61 are the 2 destination IPs that have received numerous ICMP unreachable destination messages. The unusual high numbers of messages tend to suggest malicious traffic as infected by CodeRed as explained in Section 5.1.2.4.

In view of the excessive number of ICMP messages, the cause of such traffic should be determined. The investigation includes looking up for information about the sources through look up links such as ARIN. For example, a snapshot of ARIN for 161.142.32.233 in Figure 25 shows that the source IP belongs to Mimos. In fact, most of the top source IPs are attributable to Mimos.

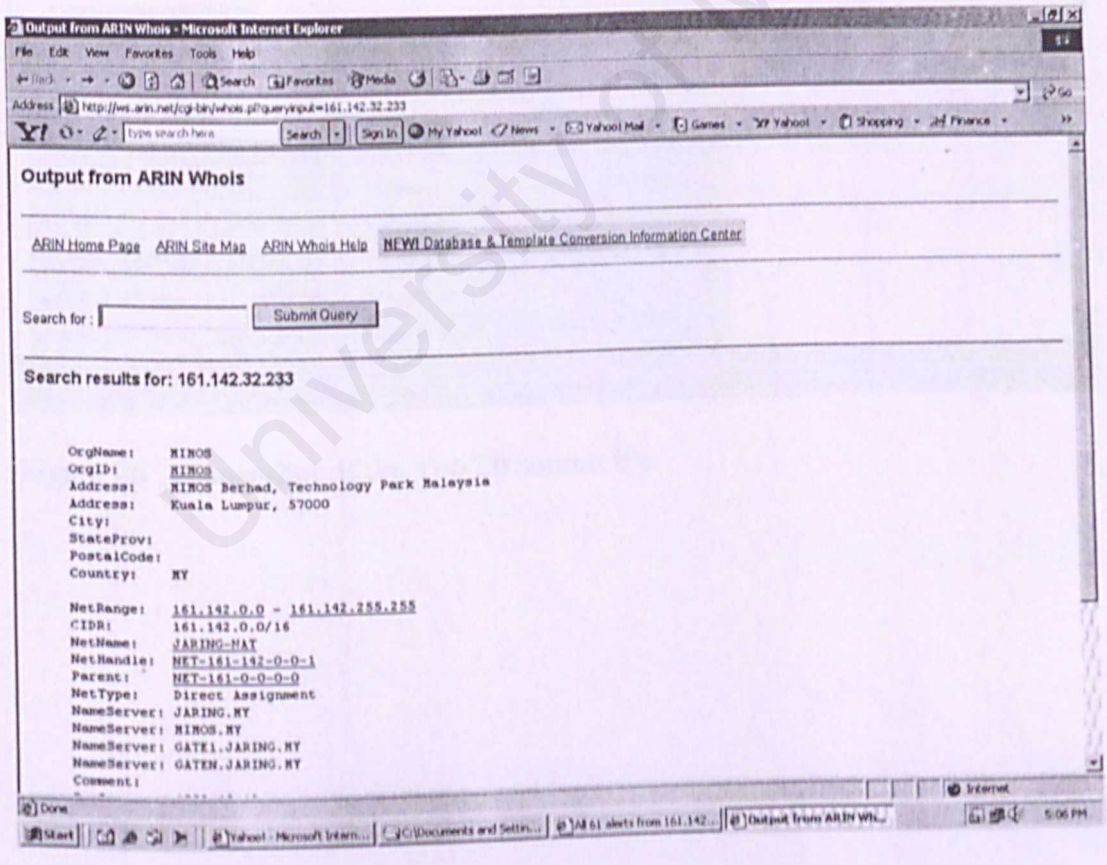


Figure 25 : Snapshot of ARIN look up on 161.142.132.233

5.2 Analysis of results – VLAN monitoring

Based on the Top 20 source IPs for Snort implementation at location A, there is an IP which is related to NAT i.e., 202.185.108.225 as shown in Figure 26. The Top 20 destination IPs for Snort implementation at location A also shows that 202.185.108.225 is one of the top ranked destination IPs as shown in Figure 27.

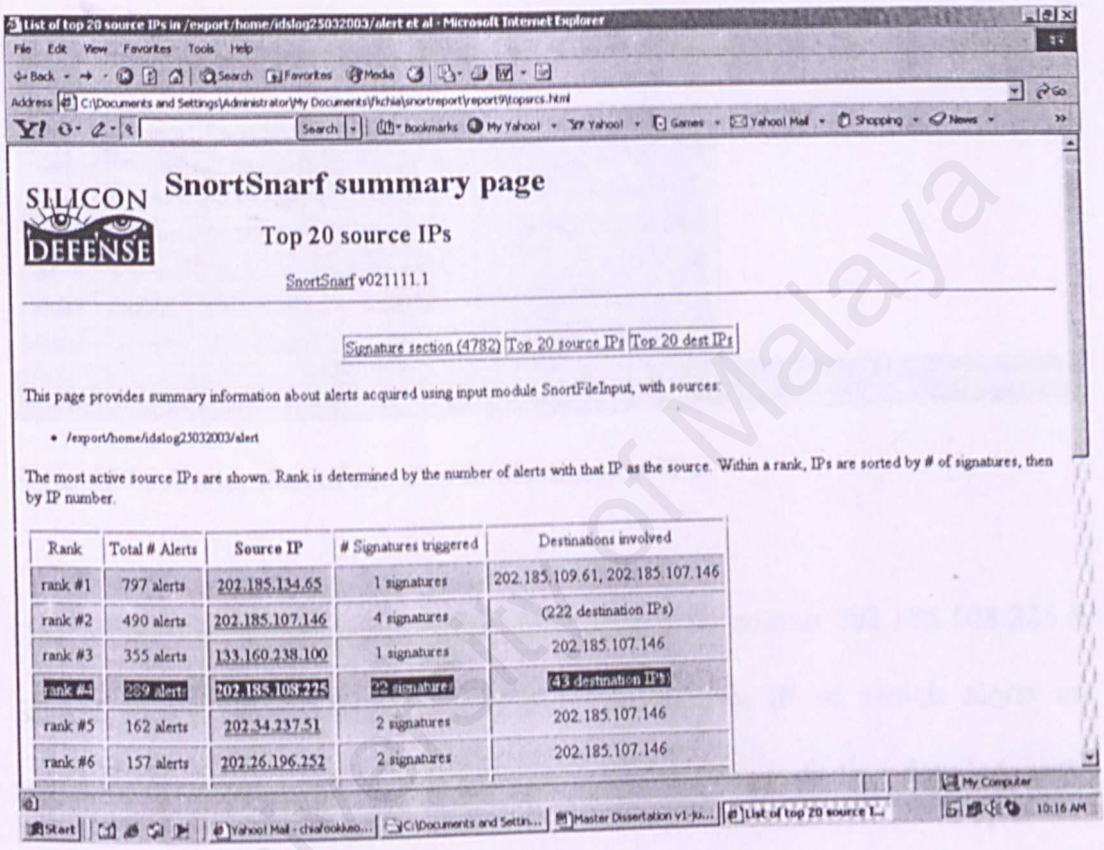
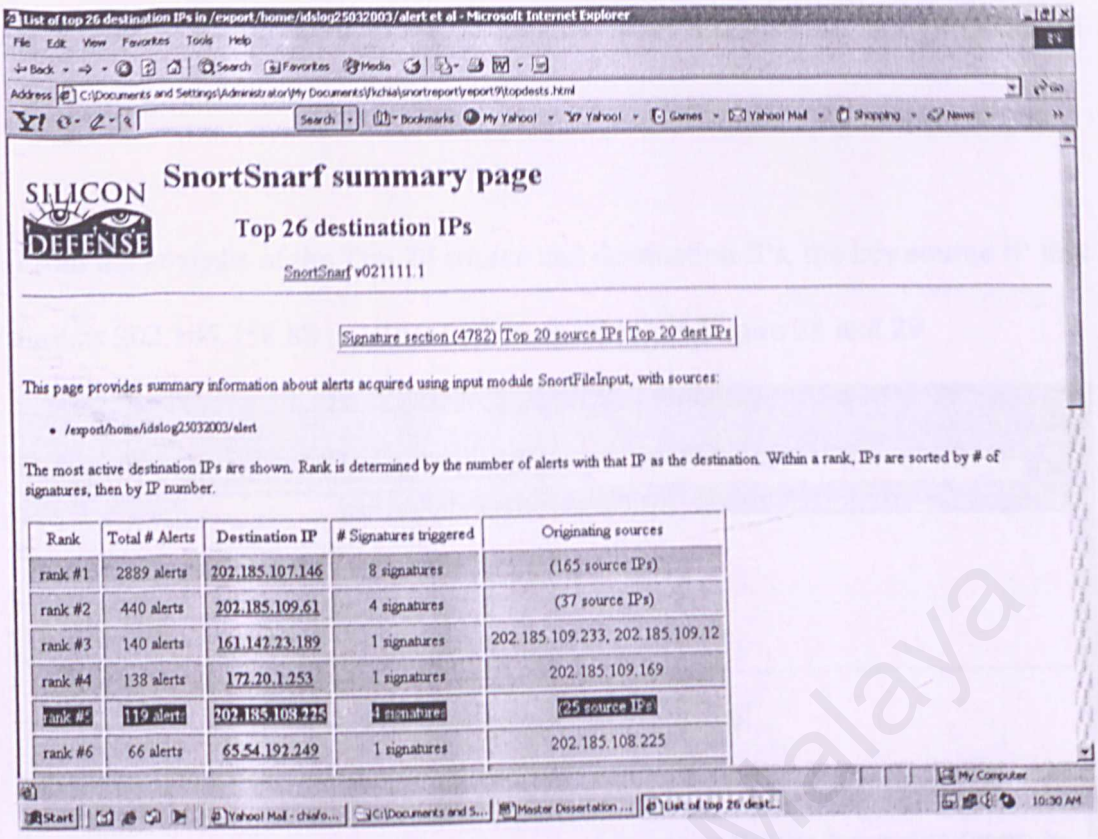


Figure 26 : A snapshot of the Top 20 source IPs



List of top 26 destination IPs in /export/home/idslog25032003/alert et al - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Address C:\Documents and Settings\Administrator\My Documents\fkchai\snortreport\report9\topdests.html

SILICON DEFENSE

SnortSnarf summary page

Top 26 destination IPs

SnortSnarf v021111.1

Signature section (4782) | Top 20 source IPs | Top 20 dest IPs

This page provides summary information about alerts acquired using input module SnortFileInput, with sources:

- /export/home/idslog25032003/alert

The most active destination IPs are shown. Rank is determined by the number of alerts with that IP as the destination. Within a rank, IPs are sorted by # of signatures, then by IP number.

Rank	Total # Alerts	Destination IP	# Signatures triggered	Originating sources
rank #1	2889 alerts	202.185.107.146	8 signatures	(165 source IPs)
rank #2	440 alerts	202.185.109.61	4 signatures	(37 source IPs)
rank #3	140 alerts	161.142.23.189	1 signatures	202.185.109.233, 202.185.109.12
rank #4	138 alerts	172.20.1.253	1 signatures	202.185.109.169
rank #5	119 alerts	202.185.108.225	4 signatures	(25 source IPs)
rank #6	66 alerts	65.54.192.249	1 signatures	202.185.108.225

Figure 27 : A snapshot of the Top 20 destination IPs

One of the key signatures detected as triggered by source 202.185.108.225 is SCAN Proxy (8080) attempt. The only destination IP of which alerts are triggered is 202.185.158.88 and the port is 8080. This is further detailed in the Snort alert as below:

```
[**] [1:620:2] SCAN Proxy (8080) attempt [**]
[Classification: Attempted Information Leak] [Priority: 2]
03/25-10:34:55.088840 202.185.108.225:2045 -> 202.185.158.88:8080
TCP TTL:127 TOS:0x0 ID:35084 IpLen:20 DgmLen:48 DF
******S* Seq: 0x81FA97 Ack: 0x0 Win: 0x2000 TcpLen: 28
TCP Options (4) => MSS: 1460 NOP NOP SackOK
```

Since this IP represent the global IP for outbound traffics from certain hosts with private IPs, more study on the VLAN traffics is required to identify the real source. NAT is only performed for internal traffics routed through Secondary core switch. Hence, Snort was implemented to monitor the VLANs that route to Secondary core switch. 375 alerts were captured on March 25, 2003 between

11:15 and 11:24. These alerts originated from 122 source IPs targeted 84 destination

From the analysis of the Top 20 source and destination IPs, the key source IP that targets 202.185.158.88 is 10.100.1.176 as shown in Figure 28 and 29.

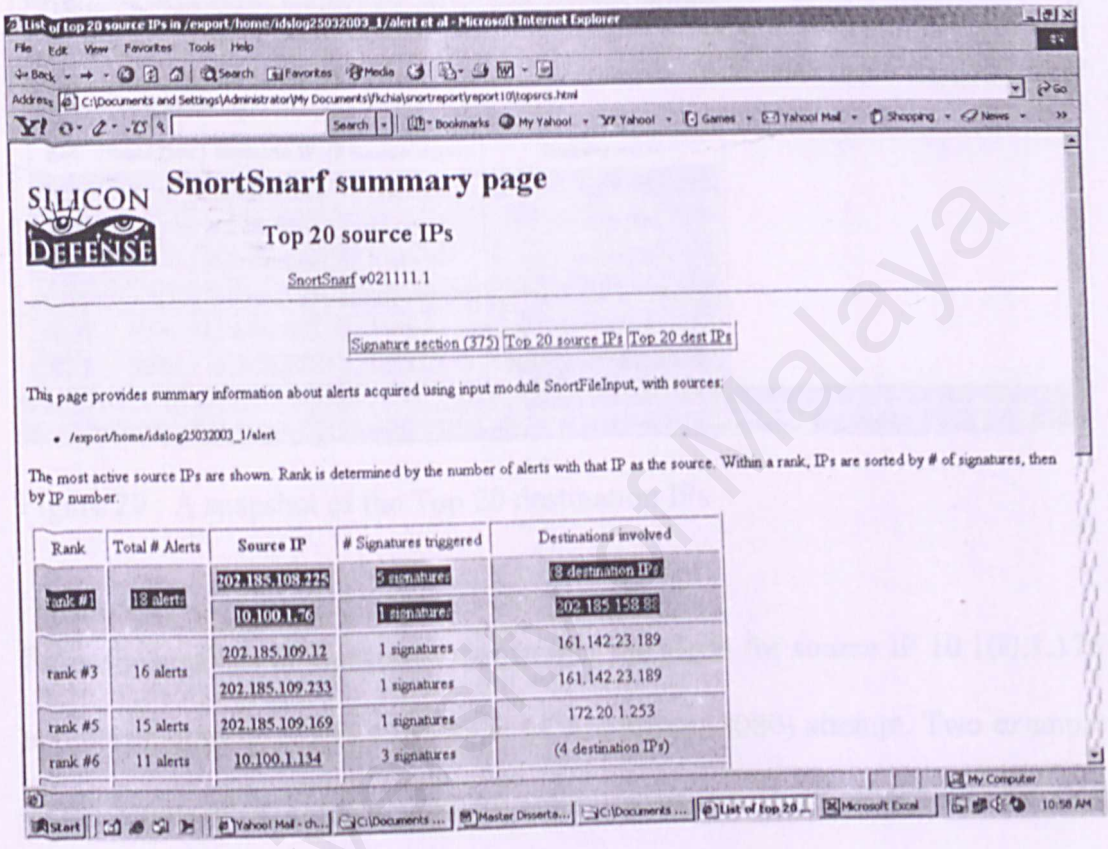
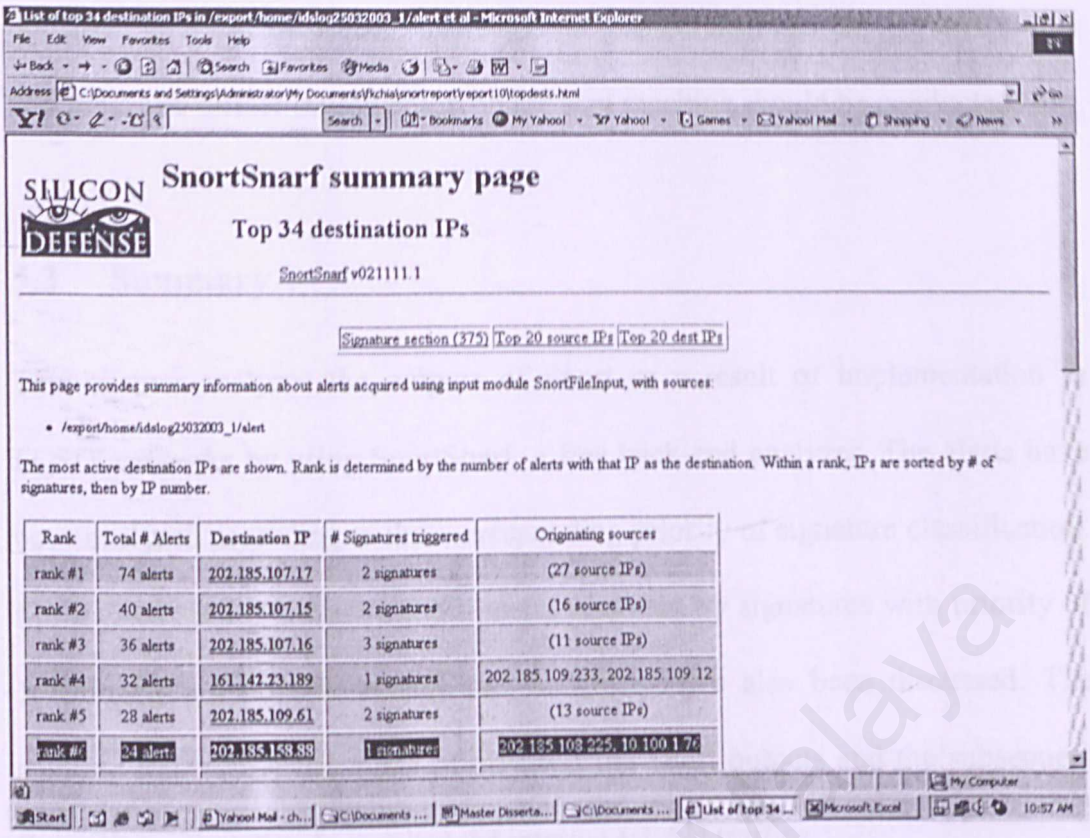


Figure 28 : A snapshot of the Top 20 source IPs



SnortSnarf summary page

Top 34 destination IPs

SnortSnarf v021111.1

Signature section (375) | Top 20 source IPs | Top 20 dest IPs

This page provides summary information about alerts acquired using input module SnortFileInput, with sources:

- /export/home/idslog25032003_1/alert

The most active destination IPs are shown. Rank is determined by the number of alerts with that IP as the destination. Within a rank, IPs are sorted by # of signatures, then by IP number.

Rank	Total # Alerts	Destination IP	# Signatures triggered	Originating sources
rank #1	74 alerts	202.185.107.17	2 signatures	(27 source IPs)
rank #2	40 alerts	202.185.107.15	2 signatures	(16 source IPs)
rank #3	36 alerts	202.185.107.16	3 signatures	(11 source IPs)
rank #4	32 alerts	161.142.23.189	1 signatures	202.185.109.233, 202.185.109.12
rank #5	28 alerts	202.185.109.61	2 signatures	(13 source IPs)
rank #6	24 alerts	202.185.158.88	1 signatures	202.185.108.225, 10.100.1.76

Figure 29 : A snapshot of the Top 20 destination IPs

From the analysis of signatures that trigger the alerts for source IP 10.100.1.176 and destination IP 202.185.158.88 is SCAN Proxy (8080) attempt. Two example of the details of SCAN Proxy (8080) attempt alerts are as shown below:

```
[**] [1:620:2] SCAN Proxy (8080) attempt [**]
[Classification: Attempted Information Leak] [Priority: 2]
03/25-11:16:01.667642 202.185.108.225:2702 -> 202.185.158.88:8080
TCP TTL:127 TOS:0x0 ID:34762 IpLen:20 DgmLen:48 DF
******S* Seq: 0x574A89E1 Ack: 0x0 Win: 0xFAF0 TcpLen: 28
TCP Options (4) => MSS: 1460 NOP NOP SackOK

[**] [1:620:2] SCAN Proxy (8080) attempt [**]
[Classification: Attempted Information Leak] [Priority: 2]
03/25-11:24:58.038674 10.100.1.76:2746 -> 202.185.158.88:8080
TCP TTL:128 TOS:0x0 ID:35252 IpLen:20 DgmLen:48 DF
******S* Seq: 0x5F5FF02A Ack: 0x0 Win: 0xFAF0 TcpLen: 28
TCP Options (4) => MSS: 1460 NOP NOP SackOK
```

The above details indicated that the destination is 202.185.158.88:8080. On the other hand, the ports of source IP are 2702 and 2746 respectively. Normally, the

port used is 1 – 1024. Hence, this suggests that the behavior of 10.100.1.176 is abnormal and further investigation on the host machine should be conducted.

5.3 Summary

This chapter analyzes the outputs of Snort as a result of implementation in FCSIT networks by using SnortSnarf, a free back-end analyzer. The alerts have been analyzed according to the corresponding priority of signature classification. More discussion is made on those alerts triggered by signatures with priority of 1. Corrective actions to rectify the situations have also been discussed. The analysis also covers the effect of NAT on the Snort outputs and the subsequent Snort implementation to monitor the internal VLAN traffics.

Chapter 6 Conclusions and future works

6.1 Conclusions

This project focuses on the implementation issues of IDS. The IDS chosen is Snort and the testing site for implementation is in FCSIT networks. The study involves detailed discussion of IDS, analysis of FCSIT networks and Snort, implementation of Snort. The Snort results are analyzed and the Snort implementation design is fine-tuned.

The study began by discussing the current network security threats and network security measures. The outcome inevitably highlights the importance of IDS as the second line of defense. Next, the selection and evaluation of IDS were explored given the various commercial and open source IDS. This has led to the selection of Snort, a free, open source, lightweight, multi-platform and customizable software for this project.

In Chapter 3, the implementation issues of Snort were scrutinized. The FCSIT network environment and protocols run were analyzed. The impacts of the VLANs, switches, HSRP and NAT on Snort implementation have been considered. Snort has also been studied to ensure proper Snort configuration. The potential locations for implementation were identified. Snort and the dependencies were subsequently implemented and the steps were detailed in Chapter 4. The dependencies includes SnortSnarf as the back-end analyzer and Apache as the web server.

Upon implementing Snort in FCSIT, alerts and log files have been successfully captured and analyzed by using SnortSnarf. The analysis of results was detailed in Chapter 5. With the assistance of reference built into Snort alerts, like SID, CVE, Bugtraq and ArachNID, the attacks or problems are studied. In addition, corresponding corrective actions have been made. The corrective actions include investigation to determine the possible infected machines, applying security patches and changes to the Snort configurations. In short, the study has met the objective and covered the scope set in Chapter 1.

6.2 Suggestions for future works

Many improvements can be made to this implementation of Snort, which includes the stealth capabilities, real time monitoring, IDS maintenance and incident handling procedures. Logically, IDS will be the target for attacks or be fooled by attackers. Hence, the IDS should have the stealth capability to avoid becoming the attacker's target. IDS outputs should also be analyzed together with other network information gathered, such as through network traffic analysis. This should provide a more complete picture of malicious activity.

Alerts are generated and stored as and when any traffic that match the rule set. However, an administrator will not be aware of these alerts until the alerts are analyzed. In order to provide a timely monitoring, the alerts generated should be sent to the administrator on real time basis. Nevertheless, the administrator should not be overwhelmed by all types of alerts which tend to cause frustration and overlooks.

The IDS implemented should be maintained continuously. The maintenance includes the continuous fine-tuning to minimize false alarms and updates of latest attack signatures. Also, the IDS version and configurations should be reviewed in relation to changes in the network environment and IDS technology.

Finally, yet importantly, there should be appropriate incident-handling procedures. These procedures are essential to ensure that appropriate corrective actions are taken against the security incidents. The security incidents include viruses, insider abuse of systems, and attacks.

References

Abhijit Sarmah (2001). Intrusion Detection Systems: Definition, Need and Challenges [Online] Available from: < http://216.239.33.100/search?q=cache:xRLVVaYoVlgJ:https://www.sans.org/rr/intrusion/IDS_definition.php+vlan+intrusion+detection+system+monitoring&hl=en&ie=UTF-8> [Accessed on 10 May, 2003]

Allen, J. Christie, A. William, F. McHugh, J. Pickel, J. Stoner, E. (2000), State of the Practice of Intrusion Detection Technologies, Carnegie Mellon Software Engineering Institute. [Online] Available from: < http://www.sei.cmu.edu/publications/documents/99.reports/99tr028/99tr028app-d.html > [Accessed on 14 November, 2002]

American Society for Industrial Security and PricewaterhouseCoopers (1999). "Trends in Proprietary Information Loss" Survey Report

Ant Allan (2002), Intrusion Detection Systems (IDSs): Perspective, Gartner Research. [Online] Available from: <http://www.tarrani.net/linda/95367.pdf> [Accessed on 15 April, 2003]

Chenxi Wang, John C. Knight (2000). Department of Computer Science, University of Virginia. [Online] Available from: <www.cert.org/research/isw/isw2000/papers/38.pdf> [Accessed on 10 May, 2003]

Chris Calabrese (2001). How to place IDS sensor in redundant networks? [Online] Available from: <http://www.sans.org/resources/idfaq/ids.redun.php> [Accessed on 10 May, 2003]

Cisco (2003). Configuration Technical Guides. [Online] Available from: <http://www.cisco.com/> [Accessed on 10 May, 2003]

Computer Economics (2001). Computer Costs Skyrocket. [Online] Available from: <http://www.cs.nmt.edu/~cs491_02/IA/viruscost.htm> [Accessed on 10 May, 2003]

CSI/FBI Computer Crime and Security Survey, 2002. [Online] Available from: <http://www.gocsi.com/press/20020407.html> [Accessed on 10 May, 2003]

Danny Rozenblum (2001), Understanding Intrusion Detection Systems. [Online] Available from: <http:// http://www.sans.org/rr/paper.php?id=337> [Accessed on 14 November, 2002]

Dorothy Denning (1987). An Intrusion Detection Model, IEEE Trans. Software Eng., Vol. SE-13, No. 2, Feb. 1987, pp. 222–232.

E. Amoroso and R. Kwapniewski (1998). A Selection Criteria for Intrusion Detection systems. Proc. 14th Ann. Computer Security Applications Conf., IEEE

Computer Soc. Press, Los Alamitos, Calif., 1998, pp. 280–288.

Eric Cole (2001). Hackers Beware. 1st Edition, New Riders Publishing, 2002, pp 64 – 101.

G. A. Fink, B. L. Chappell, T. G. Turner, and K. F. O'Donoghue (2002). A Metrics-Based Approach to Intrusion Detection System Evaluation for Distributed Real-Time Systems

IDC (2002). To Protect and Serve : Making IT Security Meet Your Business Needs. . [Online] Available from: <http://www.nokia.com/ipsecurity/emea> [Accessed on 14 November, 2002]

John McHugh, Alan Christie, and Julia Allen (2000). Software Engineering Institute, CERT Coordination Center, IEEE Software September/October 2000, pp 42 – 51.

Joy Ghosh (2001), Role of a strong intrusion detection mechanism. [Online] Available from: <http://www.sans.org/rr/paper.php?id=337> [Accessed on 14 November, 2002]

J.P. Anderson (1980). Computer Security Threat Monitoring and Surveillance, technical report, James P. Anderson Co., Fort Washington, Pa, 1980.

Mandy Andreas (2001). Surviving Security: How to Integrate People, Process and Technology. 1st Edition, Sams Publishing, 2001, pp 14 – 16.

Marcus Ranum (2001). Experiences Benchmarking Intrusion Detection Systems, [Online] Available from: <http://www.nfr.com/forum/white-papers/Benchmarking-IDS-NFR.pdf> [Accessed on 10 May, 2003]

Michael Wilkison (2001). How to evaluating Network Intrusion Detection Systems? [Online] Available from: < http://www.sans.org/resources/idfaq/eval_ids.php> [Accessed on 10 May, 2003]

Robert Graham (2000), FAQ: Network Intrusion Detection Systems. [Online] Available from: <http://www.robertgraham.com/pubs/network-intrusion-detection.html> [Accessed on 10 May, 2003]

Sandro Poppi (2002). Snort-setup for Statistic HOWTO – V1.01 Feb 23, 2002. [Online] Available from: <http://www.lugburghausen.org/dienste/projekte?Snort-Statistics/Snort-statistics-HOWTO.pdf> [Accessed on 10 May, 2003]

The NSS Group (2001). Intrusion Detection System Comparisons, Intrusion Detection Systems Group Test (edition 2). [Online] Available from: <http://www.nss.co.uk> [Accessed on 10 May, 2003]

The NSS Group (2002). Intrusion Detection System Comparisons, Intrusion Detection Systems Group Test (edition 2). [Online] Available from:

<http://www.nss.co.uk> [Accessed on 10 May, 2003]

William S. (2000). Network security essential. Prentice Hall, 2000, pp 280-287.

Appendix A

A1 All Snort Signatures

Signature section (4782)	Top 20 source IPs	Top 20 dest IPs

4782 alerts found using input module SnortFileInput, with sources:

- /export/home/idslog25032003/alert

Earliest alert at **10:34:34**.800094 *on 03/25/2003*
Latest alert at **11:15:11**.161459 *on 03/25/2003*

Priority	Signature (click for sig info)	# Alerts	# Sources	# Dests	Detail link
N/A	(spp_portscan2) Portscan detected	93	31	47	Summary
3	ICMP PING CyberKit 2.2 Windows [sid] [arachNIDS]	2	1	1	Summary
3	ICMP Destination Unreachable (Communication with Destination Host is Administratively Prohibited) [sid]	5	2	2	Summary
3	ICMP PING speedera [sid]	56	10	3	Summary
3	ICMP Destination Unreachable (Communication Administratively Prohibited) [sid]	701	157	2	Summary
2	WEB-MISC robots.txt access [sid]	1	1	1	Summary
2	WEB-MISC nc.exe attempt [sid]	1	1	1	Summary
2	WEB-CGI rsh access [sid]	1	1	1	Summary
2	WEB-CGI finger access [sid]	1	1	1	Summary
2	ICMP Source Quench [sid]	1	1	1	Summary
2	WEB-MISC ftp attempt [sid]	2	1	1	Summary
2	WEB-MISC /doc/ access [sid]	2	1	1	Summary
2	WEB-CGI calendar access [sid]	2	2	2	Summary
2	ATTACK RESPONSES Invalid URL [sid]	2	1	1	Summary
2	WEB-IIS fpcount access [sid] [BUGTRAQ]	2	1	1	Summary
2	WEB-CGI scriptalias access [sid]	4	1	1	Summary
2	ICMP Large ICMP Packet [sid] [arachNIDS]	4	1	1	Summary
2	WEB-CGI search.cgi access [sid]	4	1	1	Summary
2	WEB-MISC http directory traversal [sid]	5	1	2	Summary

	[arachNIDS]				
2	WEB-CGI htsearch access [sid]	6	1	1	Summary
2	WEB-CGI swc access [sid]	6	1	1	Summary
2	WEB-IIS showcode.asp access [sid]	8	1	1	Summary
2	ICMP L3retriever Ping [sid] [arachNIDS]	8	1	1	Summary
2	WEB-IIS view source via translate header [sid]	28	1	1	Summary
2	WEB-MISC login.htm access [sid]	30	1	3	Summary
2	WEB-MISC ?open access [sid]	32	1	2	Summary
2	SCAN Proxy (8080) attempt [sid]	46	3	1	Summary
2	ATTACK RESPONSES 403 Forbidden [sid]	51	12	6	Summary
2	WEB-CGI redirect access [sid]	81	2	6	Summary
2	ICMP PING NMAP [sid] [arachNIDS]	278	3	2	Summary
2	ICMP redirect host [sid]	833	3	2	Summary
2	ICMP redirect net [sid]	1780	14	2	Summary
1	WEB-MISC cross site scripting attempt [sid]	1	1	1	Summary
1	WEB-MISC Transfer-Encoding: chunked [sid]	2	2	2	Summary
1	WEB-IIS fpcount attempt [sid] [BUGTRAQ]	2	1	1	Summary
1	WEB-MISC showcode access [sid]	2	1	1	Summary
1	WEB-IIS CodeRed v2 root.exe access [sid]	2	1	1	Summary
1	WEB-PHP external include path [sid]	2	1	2	Summary
1	WEB-CLIENT Outlook EML access [sid]	4	1	1	Summary
1	WEB-MISC jigsaw dos attempt [sid]	4	1	1	Summary
1	WEB-CLIENT javascript URL host spoofing attempt [sid] [BUGTRAQ]	6	2	2	Summary
1	WEB-IIS unicode directory traversal attempt [sid]	7	1	1	Summary
1	WEB-PHP content-disposition [sid] [BUGTRAQ]	8	1	3	Summary
1	POP3 PASS overflow attempt [sid]	18	8	1	Summary
1	WEB-IIS cmd.exe access [sid]	302	26	262	Summary
1	WEB-IIS ISAPI .ida attempt [sid]	346	25	266	Summary

A2 Top 20 Source IPs

Signature section (4782)	Top 20 source IPs	Top 20 dest IPs

This page provides summary information about alerts acquired using input module SnortFileInput, with sources:

- /export/home/idslog25032003/alert

The most active source IPs are shown. Rank is determined by the number of alerts with that IP as the source. Within a rank, IPs are sorted by # of signatures, then by IP number.

Rank	Total # Alerts	Source IP	# Signatures triggered	Destinations involved
rank #1	797 alerts	202.185.134.65	1 signatures	202.185.109.61, 202.185.107.146
rank #2	490 alerts	202.185.107.146	4 signatures	(222 destination IPs)
rank #3	355 alerts	133.160.238.100	1 signatures	202.185.107.146
rank #4	289 alerts	202.185.108.225	22 signatures	(43 destination IPs)
rank #5	162 alerts	202.34.237.51	2 signatures	202.185.107.146
rank #6	157 alerts	202.26.196.252	2 signatures	202.185.107.146
rank #7	150 alerts	202.209.208.62	1 signatures	202.185.107.146
rank #8	144 alerts	211.4.199.30	1 signatures	202.185.107.146
rank #9	141 alerts	202.16.222.254	1 signatures	202.185.107.146
rank #10	139 alerts	202.185.109.12	2 signatures	192.168.1.85, 161.142.23.189
rank #11	138 alerts	202.185.109.169	1 signatures	172.20.1.253
rank #12	135 alerts	202.220.122.6	1 signatures	202.185.109.61
		202.236.168.37	1 signatures	202.185.107.146
rank #14	133 alerts	202.211.64.169	1 signatures	202.185.107.146
rank #15	121 alerts	202.251.88.254	1 signatures	202.185.107.146
rank #16	100 alerts	202.189.31.108	1 signatures	202.185.107.146
rank #17	64 alerts	202.185.109.61	5 signatures	(32 destination IPs)
rank #18	61 alerts	161.142.32.233	1 signatures	202.185.107.146, 202.185.109.61
rank #19	57 alerts	161.142.34.241	1 signatures	202.185.107.146, 202.185.109.61

rank #20	54 alerts	161.142.36.201	1 signatures	202.185.107.146, 202.185.109.61

A3 Top 20 Destination IPs

Signature section (4782)	Top 20 source IPs	Top 20 dest IPs

This page provides summary information about alerts acquired using input module SnortFileInput, with sources:
/export/home/idslog25032003/alert
The most active destination IPs are shown. Rank is determined by the number of alerts with that IP as the destination. Within a rank, IPs are sorted by # of signatures, then by IP number.

Rank	Total # Alerts	**Destination IP**	# Signatures triggered	Originating sources
rank #1	2889 alerts	**202.185.107.146**	8 signatures	(165 source IPs)
rank #2	440 alerts	**202.185.109.61**	4 signatures	(37 source IPs)
rank #3	140 alerts	**161.142.23.189**	1 signatures	202.185.109.233, 202.185.109.12
rank #4	138 alerts	**172.20.1.253**	1 signatures	202.185.109.169
rank #5	119 alerts	**202.185.108.225**	4 signatures	(25 source IPs)
rank #6	66 alerts	**65.54.192.249**	1 signatures	202.185.108.225
rank #7	46 alerts	**202.185.158.88**	1 signatures	(3 source IPs)
rank #8	36 alerts	**202.185.111.134**	2 signatures	202.185.108.225
rank #9	24 alerts	**202.185.112.3**	1 signatures	202.185.108.225
rank #10	22 alerts	**202.185.111.131**	3 signatures	(9 source IPs)
rank #11	15 alerts	**202.185.109.22**	3 signatures	202.31.238.56
rank #12	14 alerts	**216.239.57.100**	1 signatures	202.185.108.225
rank #13	11 alerts	**202.185.109.169**	1 signatures	202.185.112.132
rank #14	10 alerts	**206.104.8.56**	1 signatures	202.185.108.225
rank #15	9 alerts	**202.185.107.1**	1 signatures	(9 source IPs)
rank #16	8 alerts	**202.185.109.46**	2 signatures	(3 source IPs)
		202.185.111.83	1 signatures	202.185.108.225
		203.92.129.7	1 signatures	202.185.108.225
		216.5.163.42	1 signatures	202.185.108.225
rank	7 alerts	**202.95.236.192**	2 signatures	202.185.107.146

#20		202.157.149.111	2 signatures	202.185.107.146
		202.157.154.120	2 signatures	202.185.107.146
		202.157.164.2	2 signatures	202.185.109.61
		202.184.16.9	2 signatures	202.185.107.146
		202.185.201.68	2 signatures	202.185.107.146
		202.220.217.146	2 signatures	202.185.107.146