MANAGING NETWORK SECURITY WITH SNORT OPEN SOURCE INTRUSION DETECTION TOOLS

Babatunde O. Lawal
Computer Science Department
Olabisi Onabanjo University Consult
Ibadan Centre
Ibadan, Nigeria
E-mail: lawal5@yahoo.com

Okesola, J. Olatunji
Central Bank of Nigeria
Lagos, Nigeria
tunji_okesola@yahoo.co.uk

ABSTRACT

Organizations both large and small are constantly looking to improve their posture on security. Hackers and intruders have made many successful attempts to bring down high-profile company networks and web services for lack of adequate security. Many methods have been developed to secure the network infrastructure and communication over the Internet such as the firewall and intrusion detection systems. While most organizations deploy security equipment, they still encounter the challenge of monitoring and reviewing the security events. There are various intrusion detection tools in the market for free. Also, there are multiple ways to detect these attacks and vulnerabilities from being exploited and leaking corporate data on the internet. One method involves using intrusion detection systems to detect the attack and block or alert the appropriate staff of the attack. Snort contains a suite of tools that aids the administrators in detecting these events. In this paper, Snort IDS was analysed on how it manages the network from installation to deployment with additional tools that helps to analyse the security data. The components and rules to operate Snort were also discussed. As with other IDS it has advantages and disadvantages.

Keywords: Intrusion detection, Network Security, Snort, Open Source Tools

1. INTRODUCTION

Security is a big issue for all networks in today's enterprise environment [1]. Hackers and intruders have made many successful attempts to bring down high-profile company networks and web services. Many methods have been developed to secure the network infrastructure and communication over the Internet, among them is the use of firewalls, encryption, and virtual private networks. They do not provide full protection and still need to be complimented by an intrusion detection system [1]. Intrusion detection is a relatively new addition to such techniques. Intrusion detection methods started appearing in the last few years. Using intrusion detection methods, administrators can collect and use information from known types of attacks and find out if someone is trying to attack the network or a particular host. The information collected this way can be used to harden the network security, as well as for legal purposes. Both commercial and open source products are now available for this purpose. Many vulnerability assessment tools are also available in the market that can be used to assess different types of security holes present in the network. A comprehensive security system consists of multiple tools, including:

- Firewalls that are used to block unwanted incoming as well as outgoing traffic of data.
- Intrusion detection systems (IDS) that are used to find out if someone has gotten into or is trying to get into your network.
- Vulnerability assessment tools that are used to find and plug security holes present in your network. Information collected from vulnerability assessment tools is used to set rules on firewalls so that these security holes are safeguarded from malicious Internet users. [1].

These tools can work together and exchange information with each other. Some products provide complete systems consisting of all of these products bundled together [2]. Security management plays an important role in today's network management tasks. Defensive information operations, and intrusion detection systems are primarily designed to protect the availability, confidentiality and integrity of critical network information systems [3]. The automated detection and immediate reporting of these events are required in order to provide a timely response to attacks [4]. A balance therefore exists between the use of resources and the accuracy and timeliness of intrusion detection information. Since most of the commercial intrusion detection systems are at typically expensive and they tend to represent a significant resource requirement in themselves, for small networks, use of such IDS is not feasible [5]. Therefore, the open source IDS comes to reality.

Intrusion detection is a set of techniques and methods that are used to detect suspicious activity both at the network and host levels. Intrusion detection systems fall into two basic categories: signature-based intrusion detection systems and anomaly detection systems. Intruders have signatures, like computer viruses, that can be detected using software. Network administrator tries to find data packets that contain any known intrusion-related signatures or anomalies related to Internet protocols. Based upon a set of signatures and rules, the detection system is able to find and log suspicious activity and generate alerts.
Anomaly-based intrusion detection usually depends on packet anomalies present in protocol header parts. In some cases these methods produce better results compared to signature-based IDS. Usually an intrusion detection system captures data from the network and applies its rules to that data or detects anomalies in it.

Intrusion Detection Tools
Any enterprise or organisation transacting over the network should require intrusion detection and preventing mechanism. Products vary from freeware (open source) to commercial products [8] [9]. Few of the most common open source IDS tools are Suricata, Security Onion, OSSEC, Prelude, Bro and Snort.

Table 1 Features of IDS Tools [13]

FEATURES / TOOLS	HIDS	NIDS	ATTACKS DETECTED / CONDUCTED	HUMAN COMPUTER INTERFACE	LICENCE	PLATFORMS SUPPORTED
SNORT [15]	No	Yes	DOS and GUI attacks, Intrusion attacks, Port Scans, SMB probes, Layer 3 and above attacks	GUI / Command Line	Open Source	Linux, Windows, FreeBSD, MAC OS
OSSEC [15]	Yes	No	Attempts to scan non-existent Files, Secure Shell attacks, FTP Scans, SQL Injections, File System attacks	GUI	Open Source	Linux, Windows, FreeBSD, MAC OS
SURICATA [10]	No	Yes	Intrusion attacks, File System attacks	Command Line	Open Source	Linux, Windows, FreeBSD, MAC OS
SECURITY ONION [11]	Yes	Yes	Intrusion attacks, Full packet capture	GUI	Open Source	Linux
PRELUDE [12]	Yes	Yes	Intrusion attacks,		Open Source	Linux, Windows, FreeBSD, MAC OS
BRO [23]	No	Yes	Network Intrusion attacks, File System attacks	Command Line	Open Source	Unix
FRAGROUTE [13]	No	Yes	Insertion, Evasion and Denial of Service	Command Line	Open Source	Linux, FreeBSD
METASPLOIT[13]	No	Yes	Vulnerability Exploitation	Command Line	Open Source	Linux, Windows, FreeBSD, MAC OS
TRIPWIRE [13], [14]	Yes	No	Root Kit Detection, File Integrity Check	Command Line	Open Source	Linux, Windows, FreeBSD, MAC OS

The rest of the paper is organised as follows. Section 2 discussed Snort, its components, architecture and other security features. Section 3 discussed Snort management from installation to deployment and how to configure it for maximum protection against attacks. In section 4, we discussed the different systems and networks to watch when configuring and administering the Snort IDS and in section 5, we briefly highlighted two other tools that Snort uses to enhance it effectiveness. Section 6 discussed both the advantages and disadvantages of Snort IDS while concluding remark given in section 7 and the references listed in section 8.

2. SNORT OPEN SOURCE INTRUSION DETECTION SYSTEM

Snort [16] is, without doubt, one of the most popular open source security tools. With millions of downloads, it is used by individuals as well as large corporations or government organizations. The first version was written in 1998 by Martin Roesch, who later founded Sourcefire. Since then, the product evolved both as features and as portability: currently Snort is available for most major platforms including Windows, BSD, Solaris or Mac OS X. It is worth mentioning that Snort has an excellent support from the user community. This can be considered as a big advantage since the availability of signatures for new attacks can be faster than for most commercial IDS tools [16].

2.1 Snort Preprocessors and Rules.
Working as IDS, Snort uses preprocessors and rules

2.1.1 *Snort* Preprocessors
It allow the functionality of Snort to be extended by allowing users and programmers add modular plug-ins. Preprocessor code is run before the detection engine is called, but after the packet has been decoded. Such preprocessors exist for IP defragmentation, TCP stream reassembly, HTTP, FTP, SMTP, SSH etc [17].

2.1.2 *Snort* rules
Like viruses, most intruder activity has some sort of signature. Information about these signatures is used to create Snort rules. Snort uses a simple, lightweight rules description language that is flexible and quite powerful. There are a number of simple guidelines to remember when developing Snort rules that will help safeguard sanity. Most Snort rules are written in a single line. This was required in versions prior to 1.8. In current versions of Snort, rules may span multiple lines by adding a backslash \ to the end of the line [18].

Snort rules are divided into two logical sections, the rule header and the rule options [17]. The rule header contains the rule's action, protocol, source and destination IP addresses and netmasks, and the source and destination ports information. The rule option section contains alert messages and information on which parts of the packet should be inspected to determine if the rule action should be taken [17].

2.2 Snort Architecture
The main components of the new architecture are briefly presented below.

i. The *Data Source* component encapsulates the common functionality needed by any network traffic before the analysis tasks and incorporates several modules:

- *The Data Acquisition Module (DAQ)* – gets the packets from the underlying hardware – Snort 3.0 can incorporate arbitrary packet interfaces like libpcap, IPQ etc;
- *The Decoder* – which performs the same tasks like in Snort 2.x: validate the packets, detect protocol anomalies and provide a referential structure;
- *The Flow Manager* – can help tracking the conversations on the network;

ii. The IP Defragmenter – can provide IPv4 and IPv6 services for putting the packets back together and for fragment reassembly.
iii. The *Action System* handles event queuing, notification and logging when the system fires events. The supported types are text (console), syslog and Unified 2 (a serialized binary stream format).

- *The TCP Stream Reassembler* – provide target-based services for reassembling TCP segments into normalized streams.

iv. The *Data Source API* is an interface between the *Data Source* component and the *Dispatcher*.
v. The *Dispatcher* coordinates the information flow between the different components of Snort 3.0.

From the case study carried out by [19] it was deduce that Snort IDS tool detected more attacks than others in study and evidencing the general attestations of Snort's effectiveness in the detecting and prevention of network attacks. Snort IDS tool is therefore worth further analysis to study its features, capabilities and potentials that it offers the security domain for usage considerations as well as how it manages the enterprise network.

2.3 Snort Modes
Snort is a single-threaded application, which can be configured to operate in four modes [20] in [21]:

a) Packet Sniffer Mode: Packet Sniffer mode simply reads the packets off of the network and displays them in a continuous stream on the console (screen).

b) Packet logger Mode: Packet Logger mode logs the packets to disk. To record the packets to the disk, specify a logging directory and Snort will automatically know how to go into packet logger mode. A directory named log in the current directory would be created. When Snort runs in this mode, it collects every packet it sees and places it in a directory hierarchy based upon the IP address of one of the hosts in the datagram.

c) Detection Mode: Network Intrusion Detection System (NIDS) mode allows Snort to analyze network traffic for matches against a user-defined rule set and performs several actions based upon what it sees.

d) Prevention Mode/ Inline Mode: It prevents the network threats. Snort Inline obtains packets from IP tables instead of *libpcap* and then uses new rule types to help IP tables pass or drop packets based on Snort rules. There are three rule types you can use when running Snort with Snort Inline:

- *drop* - The drop rule type will tell IP tables to drop the packet and log it via usual Snort means.
- *reject* - The reject rule type will tell IP tables to drop the packet, log it via usual Snort means, and send a TCP reset if the protocol is TCP or an ICMP port unreachable if the protocol is UDP.
- *sdrop* - The *sdrop* rule type will tell IP tables to drop the packet. Nothing is logged [21].

2.4 Snort Alert Modes

When Snort is running in the Network Intrusion Detection (NID) mode, it generates alerts when a captured packet matches a rule. Snort can send alerts in many modes. These modes are configurable through the command line as well as through snort.conf file. If the alert packets are logged, the administrator needs to decide how much detail he wants and in what format the alert data is wanted. The table below lists options that can be used from the command line using the -A switch. There are also the syslog, smb, and database output options, but these don't use the –A switch from the command line. They use separate output modules and offer a wider variety of output options. These must be configured at compile time with switches added to the configure statement [22].

- SMB sends the alerts to the Windows pop-up service, so alerts can visually pop up on the screen or the screen of a monitoring machine.
- Syslog sends the alerts to a UNIX Syslog server. Syslog is a service running on a machine that can capture and store various log files. This helps consolidate logs for the network in a single place, as well as making it more difficult for a hacker to erase logs of an intrusion.
- Snort directly supports four kinds of database output through its output modules. The supported formats are MySQL, PostgreSQL, Oracle, and unixODBC. This should meet the needs of most database users.
- Sending Alerts to SNMP: One very useful feature of Snort is SNMP traps. Administrator can configure an output plug-in to send messages in the form of SNMP traps to a network management system. Using this feature it can integrate the intrusion detection sensors into any centralized NMS like HP OpenView, OpenNMS, MRTG.
- Sending Alerts to Windows: Snort can send alerts to Microsoft Windows machines in the form of pop up windows. These pop-up windows are controlled by Windows Messenger Service [22].

Table 2: Snort Alert Mode Options [22]

Options	Descriptions
-A full	Full alert information including application data. This is the default alert mode and will be used when nothing is specified.
-A fast	Fast mode. Logs only the packet header information and the alert type. This is useful on very fast networks, but if you need more forensic information, you should use the full switch.
-A unsock	Sends the alert to a UNIX socket number that another program can be listening on.
-A none	Turns the alerts off.

2.5 Components of Snort

Snort is logically divided into multiple components. These components work together to detect particular attacks and to generate output in a required format from the detection system [17]. A Snort-based IDS consists of the following major components:

- Packet Decoder: takes packets from different types of network interfaces (Ethernet, SLIP,PPP…), prepare packets for processing
- Preprocessor: (1) prepare data for detection engine; (2) detect anomalies in packet headers; (3) packet defragmentation;(4) decode HTTP URI; (5) reassemble TCP streams.
- Detection Engine: the most important part, applies rules to packets
- Logging and Alerting System
- Output Modules: process alerts and logs and generate final output [17].

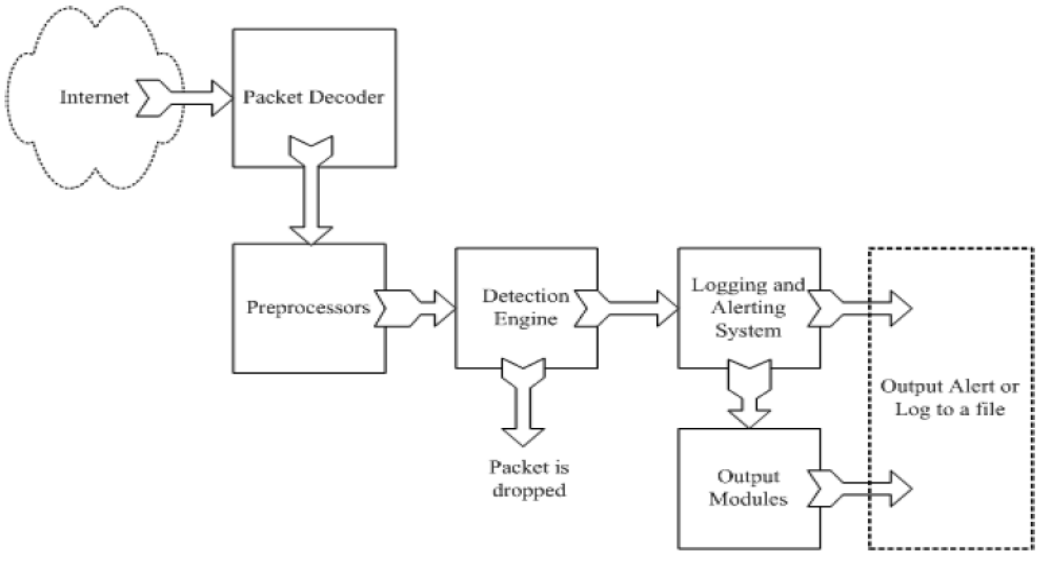

Figure 1 Components of Snort [17]

2.6 Snort Supported Platforms
Snort is supported on a number of hardware platforms and operating systems. Currently Snort is available for the following operating systems: Linux, OpenBSD, FreeBSD, NetBSD, Solaris (both Sparc and i386), HP-UX, AIX, IRIX, MacOS, and Windows. Thus Snort toolkit runs on any modern operating system and any old hardware one has. It helps to fix a number of network problems and intrusion detections [23].

2.7 How Snort Protect the IDS
Snort protects the intrusion detection system in two ways as mentioned below [17]:

2.7.1 Snort on Stealth Interface
Sometimes the administrator may want to run Snort in stealth mode. In stealth mode, other hosts are not able to detect the presence of the Snort machine. In other words, the Snort machine is not visible to intruders or other people. There are multiple ways to run Snort in stealth mode. One of these methods is to run Snort on a network interface where no IP address is assigned.

2.7.2 Snort with no IP Address Interface
Running Snort on a network interface without an IP address is feasible in the following two cases:
a. A stand-alone Snort sensor with only one network adapter.
b. A Snort sensor with two network adapters: one to access the sensor from an isolated network and the other one connected to the public network and running in stealth mode. This arrangement is shown in Figure below where network interface eth1 is connected to a private isolated network and eth0 is connected to a public network.

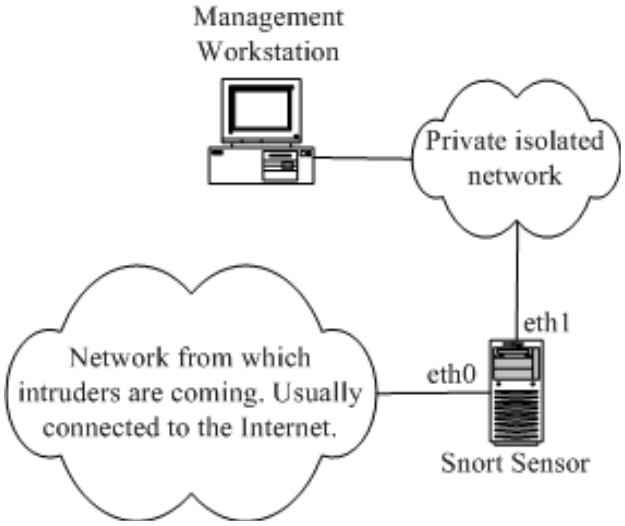

Figure 2 Running Snort in stealth mode on a system with two network adapters [17].

3. MANAGING SNORT NETWORK INTRUSION DETECTION

3.1 Installing Snort
The initial installation and configuration of Snort is fairly straightforward, and administrators can use the available support of the open source community surrounding Snort to aid in the ongoing maintenance and administration of the intrusion detection system (IDS) installation. There are two ways to install Snort: the Pre-compiled and the source code format. Installation of the pre-compiled RPM package is very easy and requires only a few steps. However, if Snort is in source code format, the installation process may take some time and understanding. At this time, the latest version is 2.9.6.0 and could be downloaded at the Snort website.

To install, run the following command to install Snort binaries:
rpm --install snort-2.9.6-0 snort.i386.rpm

This command will perform the following actions:
• Create a directory /etc/snort where all Snort rule files and configuration files are stored.
• Create a directory /var/log/snort where Snort log files will be stored.
• Create a directory /usr/share/doc/snort-1.9.0 and store Snort documentation files in that directory. You will see files like FAQ (Frequently Asked Questions), README and other files in this directory.
• Create a file snort-plain in /usr/sbin directory. This is the Snort daemon.
• Create a file /etc/rc.d/init.d/snortd file which is startup and shutdown script. On RedHat Linux, this is equivalent to /etc/init.d/snortd.

Basic installation is complete at this point and you can start using Snort. The version of Snort installed this way is not compiled with database support, so you can use it only for logging to files in the /var/log/snort directory.

Starting, Stopping and Restarting Snort

To run Snort manually, use the following command: /etc/init.d/snortd start
To stop Snort, use the following command: /etc/init.d/snortd stop
To restart Snort, use this command: /etc/init.d/snortd restart

3.2 Snort IDS Configuration on Windows

This section explains steps to configure Snort on Windows XP machine and how to use it for detection of attacks.

Steps:

1. Download Snort from "**http://www.snort.org/**" website.
2. Also download **Rules** from the same website. You need to sign up to get rules for registered users.
3. Click on the Snort_(version-number)_Installer.exe file to install it. By-default it will install snort in the "**C:\Snort**" directory.
4. Extract downloaded Rules file: snortrules-snapshot-(number).tar.gz
5. Copy all files from the "**rules**" directory of the extracted folder and paste them into "**C:\Snort\rules**" directory.
6. Copy "**snort.conf**" file from the "**etc**" directory of the extracted folder and paste it into "**C:\Snort\etc**" directory. Overwrite existing file if there is any.
7. Open command prompt (cmd.exe) and navigate to directory "**C:\Snort\bin**" directory.
8. To execute snort in sniffer mode use following command: **snort -dev -i 2**
   -i indicate interface number, **-dev** is used to run snort to capture packets.
   To check interface list use following command: **snort -W**
9. To execute snort in IDS mode, we need to configure a file "**snort.conf**" according to our network environment.
10. Set up network address to protect in snort.conf file. To do that look for "HOME_NET" and add your IP address: var HOME_NET 10.1.1.17/8
11. Set addresses or DNS_SERVERS, if there is any, otherwise go to the next step.
12. Change RULE_PATH variable with the path of rules directory: var RULE_PATH c:\snort\rules
13. Change the path of all libraries with the name and path on your system. or change path of snort_dynamicpreprocessor variable.
    sor file C:\Snort\lib\snort_dynamiccpreprocessor\sf_dcerpc.dll
14. Change path of the "**dynamicengine**" variable value in the "**snort.conf**" file with the path of your system. Such as: dynamicengine C:\Snort\lib\snort_dynamicengine\sf_engine.dll
15. Add complete path for "include classification.config" and "include reference.config" files.
    include c:\snort\etc\classification.config, include c:\snort\etc\reference.config
16. Remove the comment on the line to allow **ICMP** rules, if it is alredy commented.
    include $RULE_PATH/icmp.rules
17. Similary, remove the comment of **ICMP-info** rules comment, if it is already commented.
    include $RULE_PATH/icmp-info.rules
18. To add log file to store alerts generated by snort, search for "output log" test and add following line:
    output alert_fast: snort-alerts.ids
19. Comment whitelist $WHITE_LIST_PATH/white_list.rules and blacklist $BLACK_LIST_PATH/black_list.rules lines. Also ensure that you add change the line above $WHITE_LIST_PATH Change nested_ip inner , \ to nested_ip inner #, \
20. Comment following lines:
    #preprocessor normalize_ip4 #preprocessor normalize_tcp: ips ecn stream
    #preprocessor normalize_icmp4 #preprocessor normalize_ip6 #preprocessor normalize_icmp6
21. Save the "**snort.conf**" file and close it.
22. Go to the "**C:\Snort\log**" directory and create a file: snort-alerts.ids
23. To start snort in IDS mode, run following command: snort -c c:\snort\etc\snort.conf -l c:\snort\log -i 2

The above command will generate log file that will not be readable without using a tool. To read it use following command:

C:\Snort\Bin\> snort -r ..\log\log-filename

To generate Log files in ASCII mode use following command while running snort in IDS mode:
snort -A console -i2 -c c:\Snort\etc\snort.conf -l c:\Snort\log -K ascii

24. Scan the computer running snort from another computer using PING or launch attack. Then check **snort-alerts.ids** file the **log** folder [24].

4. STRATEGIES OF DEPLOYING SNORT

Deploying an NIDS presents an administrator with some real challenges. Installing and getting Snort up and running is just the beginning. Administrators need to figure out what you want to watch, how you can watch it, and how to get meaningful information out of all the efforts. The initial installation and configuration of Snort is fairly straightforward. Spending time and care on the installation, initial configuration, and placement of Snort will reduce false positives, improve performance, and ensure that you are watching what is important [26].

It is important to understand that running an NIDS is not as simple as just plugging it in and watching, there are other tasks involved. One of the challenges of using an open source application like Snort is that there are new versions coming out regularly that may have additional functionalities to use [26].

Below are brief points on how network administrators can manage their network using Snort open source IDS followed by other tools its uses to enhance its functionalities.

4.1 Initial Configuration

Administrator should take his time with the initial installation and configuration of the Snort system. A thorough understanding of the types of systems, their locations, and the services they provide will allow administrator make good decisions about how to configure the sensors. The administrator should ensure to deploy separate rules for different operating systems to avoid noise and unnecessary alerts [26].

4.2 Sensor Placement

Since the Snort sensor can only alert on what it sees, the placement of the sensor is very important. In many networks, putting the sensor in the wrong spot can cause it to miss an entire network's traffic.

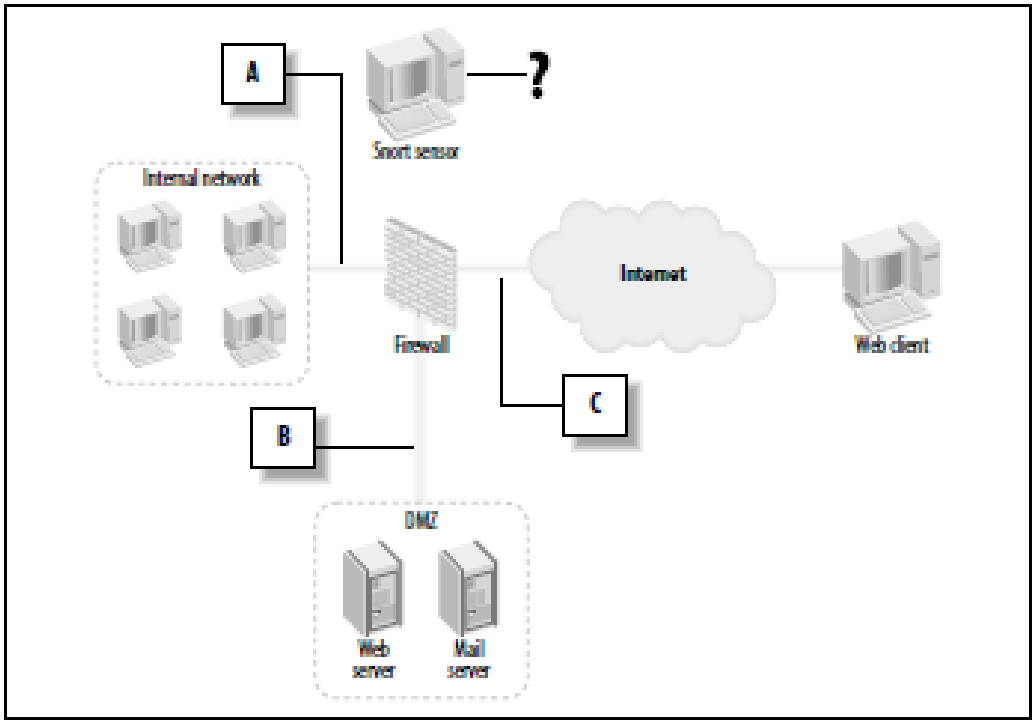

Figure 3 Sensor placement

If the Snort sensor is placed at point A, it will be able to see all traffic between the internal network and the Internet. It will not be able to see the traffic between the DMZ (containing a web server and mail server) and the Internet. In this case, an attack on the web server would go unnoticed.

If the sensor is at point B, it will see all traffic between the systems in the DMZ and the Internet. In this case, it will not see the traffic between the internal network and the Internet.

Locating a sensor at point C will allow it to see all traffic travelling to and from both networks (DMZ and internal) and the Internet. Putting the sensor at point C still leaves a potential blind spot: traffic between the DMZ and the internal network will not be watched.

Connecting sensors to the network is a very important decision. Below are some aspects of this decision.

i. Systems and Networks to Watch
It is not realistic to expect to watch all traffic between all systems on the network effectively with Snort IDS. Prioritization of systems and networks is the most necessary. Here is a list of points and connections to watch.

Internet Service: Systems that provide services to the Internet are a good first choice. These systems are more at risk than systems on the internal network. They also may be providing services to organisation's customers or business partners that are very important to the goals of the organizations [26].

Servers: There are a group of servers that provide services to people sitting at the desks such as print servers, file servers, authentication and directory servers, mail servers, intranet servers, and databases. It is also of value to watch the database storing the ERP solution or the accounting system.

Internal Network: All the workstations, laptops, and other citizens of the internal network must not be ignored. It is suggested that traffic between these systems and the Internet be watched by an NIDS. If there is a WAN connection to organisation's business partners or branch offices, a sensor watching traffic across these links is advised.
Network Connections: The exact placement of sensors is made easier by looking for natural bottlenecks (i.e. the connections between networks). The points that the network connects to the Internet is an easy choice.

Sensors: Some administrators are in doubt of where to place the sensors on the network. Sensor could be placed either inside, outside or behind the firewall seeing different traffic in different location

ii. Creating Connection Points
To create connection points, the administrator can plug a small hub into the path of the traffic they want to watch, but small hubs are very often not as reliable as the enterprise-class switches. To protect the information travelling on the network, administrator can use Cisco switches (an enterprise-class switches) [26].

iii. Encrypted Traffic
It is mostly impossible for the Snort sensor to watch the content of an encrypted packet because the traffic is always encrypted using SSL encryption to make the transaction much more secure. The solution that allows the web traffic to remain encrypted but also allows Snort to watch for signs of intrusion is an *SSL proxy*. This device sits between the client and the server and handles the task of encrypting the traffic. The traffic from the web server to the SSL proxy is not encrypted, but the traffic between the proxy and the web client *is* encrypted. Plugging the Snort sensor between the web server and the proxy allows the traffic to be monitored [26].

iv. Securing the Sensor
It is important to protect the integrity of the systems responsible for monitoring and maintaining the security of the network. The integrity of syslog servers, authentication servers, monitoring, and management tools also needs protection as well as the NIDS systems. Employing and controlling the *management network* is very necessary.

v. Choosing an Operating System for Snort
There are several things to consider when choosing an operating system on which Snort would reside. These are supportability, performance, stability, and security topping the list. Network administrator need to weight these criteria against different operating systems to choose the most appropriate for the network.

vi. Configure Interfaces
Snort sensors should be configured with at least a pair of interfaces. One of these interfaces will be on the management network; all alerting and management traffic will use this interface, keeping it away from prying eyes. Snort will use the other interface as a monitoring point. This interface will not be configured with an IP address, so it will be invisible to hosts on that network. This is commonly referred to as a *stealth interface*. Keeping the listening interface invisible to the other systems on the network makes keeping the sensor secure much easier [26].

vii. Disable Unnecessary Services
If a service is not needed for the business function of a server, it should not be installed or enabled. The fewer services running on a system, the fewer potential issues need to be secured and kept up-to-date. This is an essential step to making a system secure.

viii. Apply Patches and Updates
Commonly now, there are always updates and patches that need to be applied to the operating system and services—especially on a freshly installed system. As time goes by, it is important that administrators keep abreast of newly discovered vulnerabilities in their operating system and services. Administrator could create a maintenance schedule for when updates will be made to your systems [26].

ix. Utilize Robust Authentication
Where possible, use stronger authentication than just a simple username and password. There are weaknesses discovered in passwords and can be easily attacked through dictionary attacks. Use passwords along with mechanisms that enforce passwords of a certain length and complexity. Force the passwords to be changed periodically (e.g. 30 or 60 days). Most importantly, configure it to lock out the account after a certain number of consecutive failed password guesses. Employ a stronger mechanism for authenticating users. Smart cards (and PKI), one-time password generators, or biometric mechanisms are excellent choices.

x. Monitor System Logs
It is very important that the system is configured to generate logs and that those logs are reviewed regularly for signs of system, hardware, or configuration problems (including signs of intrusion). Auditing authentication, system function, and hardware operation is a good place to start. If possible, send the logs to a central syslog server (hopefully located on the controlled management network). This makes it much easier to review the logs and establish some correlation of events across multiple systems and networks [26].

5. OTHER TOOLS USED BY SNORT FOR EFFECTIVENESS

5.1 MySQL
All systems need some type of efficient logging feature, usually using a database at the backend. Snort can be made to work with MySQL, Oracle or any other Open Database Connectivity (ODBC) compliant database [27]. Logs and alerts can be saved to a database. Logging to a database is very useful for maintaining history data, generating reports and analyzing information—including details about the packet that triggered the alert. Since MySQL is a freely available database and works perfectly well on Windows, Linux and other operating systems, Snort is a good choice [27].

MySQL database can be used with Snort in different scenarios:
• It can be installed and run the MySQL database server on the same machine where Snort is running
• The MySQL server can be installed on a different machine and configure Snort to log to the database
• By having multiple Snort sensors to log to a centralized database server running MySQL server.

5.2 ACID – (Analysis Console for Intrusion Database)
Once the data is in the database, administrators need to choose some tools that can present the data in a way that makes managing the alerts and the sensors quick and easy such as ACID. ACID is a tool used to analyze and present Snort data using a web interface. It is written in (Pretty Home Page) PHP. It works with Snort and databases like MySQL, and makes information available in the database to the user through a web server. In addition to Snort, the tool can be used with other security-related products like firewalls and networking monitoring [26].

5.3 SnortSnarf
SnortSnarf is another tool to display Snort data using a web interface. It is available from its web site at http://www.silicondefense.com/software/snortsnarf/index.htm. Basically it is a Perl script and can run after downloading without going through any compilation process. It can parse Snort log files as well as extract data from MySQL database. The following command parses /var/log/snort/alert file and places the newly generated HTML files in the /var/www/html/snortsnarf directory where they can be viewed later using a web browser:
snortsnarf.pl /var/log/snort/alert -d /var/www/html/snortsnarf

To get data from a database, administrator have to define the following parameters on the command line: Database user name, Password, Database name, Host where database server is running and Port number for the database server. By default the port number is 3306 and this parameter is optional. The general format of defining these parameters is: user:passwd@dbname@host:port [26].

5.4 SnortSam
SnortSam is a tool used to make Snort work with most commonly used firewalls. It is used to create a Firewall/IDS combined solution. It can configure the firewall automatically to block offending data and addresses from entering the system when intruder activity is detected. It is available from http://www.snortsam.net/. The tool consists of two parts:
1. A Snort output plug-in that is installed on the Snort sensor.
2. An agent that is installed on a machine close to Firewall or Firewall itself. Snort communicates to the agent using the output plug-in in a secure way [26].

5.5 IDS Policy Manager

IDS policy manager is a Microsoft Windows based GUI. It is used to manage the Snort configuration file and Snort rules on a sensor. It is available from its web site http://activeworx.com/idspm/. The software can be downloaded and installed using normal Windows installation procedures. It has three tabs at the bottom: The "Sensor Manager" tab, the "Policy Manager" tab and the "Logging" tab [26].

5.6 Easy IDS

Easy IDS is an integrated system available from http://www.argusnetsec.com for the Linux operating system. It has all of the necessary components to build complete IDS quickly. These components are precompiled and configured for easy installation. The package includes Snort, Apache Web server, MySQL server, ACID, PHPLOT and ADODB [26].

6. ADVANTAGES AND DISADVANTAGES OF SNORT

6.1 Advantages

Snort is a very flexible application. Due to the modular design and ability to add or break in specialized software components Snort can be a powerful tool in a defence/security in-depth implementation. This design allows anyone capable of programming to build and implement their own preprocessor modules to customize Snort‟s operation to their specific environment. Customization can also be accomplished through specialized configurations of the existing pre-processor modules, as well as alert output operations [28].

Snort also has a large following and according to the Snort website snort.org, Snort is the effectively standard in intrusion detection systems. There are many commercialized systems available, but many organizations use Snort because it is an effective intrusion detection system, and free cost. Snort is a signature based detection system and with the large user base new signatures are constantly being added. This large user and support base has led to what is described as a highly effective and efficient detection engine [29].

6.2 Disadvantages

Snort does have some limited shortfalls when it comes to anomaly detection. The system was not designed for this type of operation, but some pre-processor modules attempt to add this functionality [Lippmann R., Haines J. W, Fried D. J., Korba J., and Das K., "The 1999 DARPA Off-Line Intrusion Detection Evaluation," *Lincoln Laboratory MIT*, 2000]. Currently these modules are not considered effective in detection. There is also concern about how efficient the detection engine actually is in terms of processing performance. The base engine is considered quite efficient, but there is speculation as to how efficient the system becomes when used with the pre-processor modules [29].

7. CONCLUSION

There are several IDS systems available in the market and some of them are open-source (free of charge). Snort is a free of charge IDS system available for download at its website by everyone. Snort is used to secure computer network by monitoring, detecting and analysing the network traffic and gives all the information related to a particular attack signature and its address by generating alerts. Snort helps the network administrators to make the network environment secure as it focuses on performance and simplicity which makes it best choice to be run on any operating system. It is one of the best known lightweight IDS that can easily be deployed on any node of a network, with minimal disruption to operations. When deploying Snort, it is important to make sure that the used rules are relevant and up to date otherwise the system will be much less efficient. Snort's main disadvantage is that its performance becomes down during heavy network traffic.

We analysed the management features and capabilities of Snort IDS, discussed the rule sets that should be used by administrators for maximum efficiency and some other administrative tools (MySQL, ACID) that help Snort manages its security data. We also recommended the strategies of deploying and managing snort ids and the systems and network that administrators need to watch which may be vulnerable to attacks and intrusions.

8. REFERENCES

[1] Harley Kozushko. Intrusion Detection: Host-Based and Network-Based Intrusion Detection Systems. Available at http://infohost.nmt.edu/~sfs/Students/HarleyKozushko/Papers/IntrusionDetectionPaper.pdf
[2] SnortSam at http://www.snortsam.net/
[3] Kayacik G., Zincir-Heywood A. N., "Evaluation of the Cisco IOS Firewall with Darpa 99 Dataset", Technical Report, Faculty of Computer Science, Dalhousie University, November 2002. Available at http://www.cs.dal.ca/~kayacik/download/report.pdf
[4] Bass T., "Intrusion Detection Systems and Multisensor Data Fusion", Communications of the ACM, Vol. 43, No. 4, pp 99-105, April, 2000.
[5] A case study of three open source security management tools by hilmi gunes kayacik, a. nur zincir-heywood. *Dalhousie University, Faculty of Computer Science, Canada*
[6] Mukherjee, B., Heberlein, L.T. and Levitt, K.N.; "Network intrusion detection", Proceedings of IEEE International Conference on "Network", vol. 8, Issue: 3, pp: 26 – 41, 1994.
[7] Ahmed, M.; Pal, R.; Hossain, M.; Hasan, K. and Bikas, A.N.; "A Comparative Study on the Currently Existing Intrusion Detection Systems", Proceedings of IEEE International Conference on "Computer Science and Technology", pp: 151 – 154, 2009.
[8] http://www.iana.org/assignments/ethernetnumbers
[9] O.B. Lawal, A. Ibitola & O.B. Longe (2013). Analysis and Evaluation of Network-Based Intrusion Detection and Prevention System in an Enterprise Network Using Snort Freeware. Afr J. of Comp & ICTs. Vol 6, No. 1. Pp 169-184.
[10] www.suricata-ids.org
[11] http://code.google.com/p/security-onion/wiki/IntroductionToSecurityOnion
[12] Curt Yasm. Prelude as a Hybrid IDS Framework. GCIA Gold Certification. SANS Institute 2009
[13] M. Sharma, K. Jindal, B. K Sharma. Analysis of IDS Tools & Techniques. International Journal of Emerging Technology and Advanced Engineering Website: www.ijetae.com (ISSN 2250-2459 (Online), Volume 4, Special Issue 1, February 2014
[14] James Cannady and Jay Harrell. A Comparative Analysis of Current Intrusion Detection Technologies. Georgia Tech Research Institute, Georgia Institute of Technology, Atlanta, Georgia 30332-0800
[15] Jennifer Albornoz Mulligan, security researcher at Forrester Research
[16] Jack TIMOFTE. Intrusion Detection using Open Source Tools. Revista Informatica Economică nr.2(46)/2008
[17] Rafeeq Ur Rehman. Intrusion Detection Systems with Snort. Advanced IDS Techniques Using Snort, Apache, MySQL, PHP, and ACID. Available at http://ptgmedia.pearsoncmg.com/images/0131407333/downrloads/0131407333.pdf
[18] http://manual.snort.org/node28.html
[19] A case study of three open source security management tools by hilmi gunes kayacik, a. nur zincir-heywood. *Dalhousie University, Faculty of Computer Science, Canada*
[20] Brian Caswell and Jeremy Hewlett. Snort Users Manual (http://www.snort.org/docs/)
[21] Suman Rani, Vikram Singh. Snort: An Open Source Network Security Tool for Intrusion Detection in Campus Network Environment. International Journal of Computer Technology and Electronics Engineering (IJCTEE) Volume 2, Issue 1
[22] Tony Howlett. Open Source Security Tools Practical Applications for Security. Prentice Hall, Professional Technical Reference, Upper Saddle River, NJ 07458, www.phptr.com
[23] Pritika Mehra. A brief study and comparison of Snort and Bro Open Source Network Intrusion Detection Systems. International Journal of Advanced Research in Computer and Communication Engineering. Vol. 1, Issue 6, ISSN : 2278 – 1021. August 2012
[24] http://kailaspatil.blogspot.com/2013/07/tutorial-to-configure-and-use-snort-ids.html
[25] http://searchsecurity.techtarget.com/tip/Top-five-free-enterprise-network-intrusion-detection-tools
[26] Kerry J. Cox and Christopher Gerg. Managing Security with Snort and IDS Tools. Available at www.oreilly.com/catalog/**snortids**/chapter/ch06.pdf
[27] http://www.odbc.org
[28] Baker A. R. and Esler J., *Snort IDS and IPS Toolkit*. Burlington: Syngress Publishing, Inc., 2007.
[29] Suhad Abbas Yasir. Overhead Evaluation in Real-Time Network Intrusion Detection System Using Snort. Technical Institute / shattra, available at *www.iasj.net/iasj?func=fulltext&aId=51074*.pdf

AUTHOR'S BIOGRAPHY

Babatunde O. Lawal is a lecturer in Computer Science at the Olabisi Onabanjo University Consult, Ibadan, Nigeria. He received his Master of Computer Systems (MCS) degree from University of Ibadan, Nigeria. He has worked for Trans International Bank and Spring Bank Plc as IT Support Officer and Database Administrator. His research interests are Network Security, Database Management, Data Mining and Information Systems Management. He could be reached at lawal5@yahoo.com.