Performance analysis of Mail Clients using SNORT

Mr. K Sreerama Murthy[1], Dr. S Pallam Setty[2], Dr. G S V P Raju[3]
Research scholar& CSSE, AU, Visakhapatnam, India

Abstract: *Intrusion detection system (IDS) monitor network traffic for mistrustful activity and alerts the system or network administrator, and may also take actions such as blocking the user or source IP address from accessing the network. SNORT acts as not only IDS but also can be configured as IPS for watching and interference of security attacks on networks.*

In our case, we used Snort for analysing performance of different mail clients by varying the text sizes from 50 KB to 2 MB and analysed the metrics (run time, analysed packets and total packets). From simulation scenario, we found that Hotmail is best for sending larger text and Yahoo should be less preferred for the same purpose.

Keywords — *IDS, SNORT, Mai, IPS.*

I. Introduction

An Intrusion Detection System is a device or software application that monitors network or system activities for malicious activities or policy violations and produces reports to a management station. Intrusion detection and prevention systems are primarily focused on identifying possible incidents, logging information about them, and reporting attempts. In addition, organizations use Intrusion Detection Prevention System (IDPSs) for other purposes such as identifying problems with security policies, documenting existing threats, and deter individuals from violating security policies.

All Intrusion Detection Systems use one of the two detection techniques:

Statistical anomaly-based Intrusion detection System:

A statistical anomaly-based IDS determines normal network activity like what sort of bandwidth is generally used, what protocols are used, what ports and devices generally connect to each other - and alert the administrator or user when traffic is detected which is anomalous(not normal).

Signature -based Intrusion Detection System:

Signature based IDS monitor packets in the network and compares with pre-configured and pre-determined attack patterns known as signatures.

NIDS:

Network Intrusion Detection Systems are placed at a strategic point or points within the network to monitor traffic to and from all devices on the network. Ideally one would scan all inbound and outbound traffic.

HIDS:

Host Intrusion Detection Systems run on individual hosts or devices on the network. A HIDS monitors the inbound and outbound packets from the device only and will alert the user or administrator of suspicious activity.

Snort is an open source network intrusion detection system, capable of performing real-time traffic analysis and packet logging on IP networks.

Literature Review

This [1] paper focuses on issues of information security and describes the security needs of an organization to protect their critical information from attacks. But for building new security strategies, huge effort is required, which is discussed in [2], [9] and [12].

According [5], Snort and source fire are best suitable for MNCs. Snort [5] provides high flexibility for users to self-configure and modify its source code by using Sourcefire.

The paper [8], provides differences between HIDS and NIDS systems. It describes about Promiscuous-mode and Network-mode. The main disadvantage is this IDS majorly takes care of only signature based attacks rather anomaly based attacks, so still human intervention is required [8].

Novel string matching technique [9] is an optimization of other matching algorithms. This algorithm is most efficient and ten times faster than the other existing systems and it consumes fewer resources, but the problem is its practical implementation and requires more memory, not suitable for anomaly based attacks as in [7] and [14].

Network security is a major issue and SNORT is the famous intrusion detection system in the field of open source software. In this paper, authors explained how SNORT implements the intrusion detection, which includes building the compiling environment and analysing the work flow and rule tree. This paper provides a valuable reference for the study of SNORT.

According to S. Mrdović and E. Zajko [10], discussed about distributed IDS for observing the network traffic behaviour. In this paper, they used Snort and MySQL as analysis engine and logging purpose [13].

Security of IDS is described in this paper[11]. It describes about misuse detection and anomaly detection. Three different approaches data mining, data fusion and immunological based approach used in IDS. The approaches that are discussed in [4], [9] & [14] are much sufficient for IDPS to detect and respond to anomalies in real time.

This paper [12] aimed at reducing false-positive rate by detecting multiple intrusions. Hidden Markov Model (HMM), statistical technique, decision tree technique have been used to gain the advantage of less false-positive rate that increases performance of detection[12]. If these IDS adopt the mechanism of protection that is discussed in [4] and [8] then the system can be secured in a better way.

INDRA (Intrusion Detection and Rapid Action) [13] majorly works for peer-to-peer network. If INDRA finds any interrupt then security agent cut off the effected packets, but the problem is it requires a large amount of memory to store all the collected information about intruder as discussed in [2] and [3].

This paper [14] has proposed virtual behaviour to HIDS. This technique is efficient for duplication of real operating system, invisibility and inaccessibility to intruders. Multiple virtual machines can run simultaneously on same hardware. The major benefit is cost effectiveness than other techniques discussed in [2] and [3].

Matt and Andrew in [15] investigate the IDPS and also IDS/IPS tools. SNORT is used to configure the log into the database directly. MySQL, TRIPWIRE software are used for this purpose. The major benefit of SNORT is that it can detect a large number of different attacks such as viruses, Denial of services, malware etc.

II. Problem Statement

Snort is a multimode packet analysis tool. In Snort, we mainly concentrated on sniffer mode. Snort will read the network traffic and print them to the screen. Snort is considered as superior Network Intrusion Detection System when compared to the most commercial systems. In this paper, we measured the performance of different mail clients by using Snort. In the simulation study, we selected three mail clients (Gmail, Yahoo, and Hotmail) by varying the text sizes from 50 KB to 2 MB for all the three mail clients.

III.SNORT OVERVIEW

Snort:

Snort is a free and open source Network Intrusion Detection System (NIDS) and Network Intrusion Prevention System (NIPS) and created by Martin Roesch in 1998. Snort's open source network-based intrusion detection system (NIDS) has the ability to perform real-time traffic analysis and packet logging on Internet Protocol (IP) networks. Snort performs protocol analysis, content searching, and content matching. The program can also be used to detect probes or attacks. Snort can be configured in three main modes: sniffer, packet logger, and network intrusion detection. In sniffer mode, the program will read network packets and display them on the console. In packet logger mode, the program will log packets to the disk. In intrusion detection mode, the program will monitor network traffic and analyse it against a rule set defined by the user.

Snort uses a flexible rules language to describe traffic that it should collect or pass, as well as a detection engine that uses a modular plug-in architecture. Snort has a real-time alerting capability as well, incorporating alerting mechanisms for syslog, a user-specified file, a UNIX socket, or WinPopup messages to Windows clients using Samba's smbclient. Snort has three primary uses. It can be used as a straight packet sniffer like tcpdump, a packet logger (useful for network traffic debugging and so), or as a full-blown network intrusion prevention system.

Uses:

Snort's open source network-based intrusion detection system (NIDS) has the ability to perform real-time traffic analysis and packet logging on Internet Protocol (IP) networks. Snort performs protocol analysis, content searching, and content matching. The program can also be used to detect probes or attacks, including, but not limited to, operating system fingerprinting attempts, common gateway interface, buffer overflows, server message block probes, and stealth port scans.

Email Clients:

An email client, email reader, or more formally mail user agent (MUA), is a computer program used to access and manage a user's email.

The term can refer to any system capable of accessing the user's email mailbox, regardless of it being a mail user agent, a relaying server, or a human typing on a terminal. In addition, a web application that provides message management, composition, and reception functions is sometimes also considered an email client, but more commonly referred to as webmail.

Popular web-based email clients:

Gmail, Yahoo! Mail, mail.com, Lycos mail, and Hotmail. In our case, we considered the following 3 mail clients.

Gmail:

Gmail is a free, advertising-supported email service provided by Google. Users may access Gmail as secure webmail via POP3 or IMAP4 protocols. Gmail initially started as an invitation-only beta release on

April 1, 2004 and it became available to the general public on February 7, 2007 though still in beta status at that time. The service was upgraded from beta status on July 7, 2009, along with the rest of the Google Apps suite.

Features:

- Storage
- Gmail labs
- Spam filter
- Social Networking
- Google Voice in Gmail chat
- Money Transfer

Yahoo:

Yahoo! Inc. is an American multinational internet corporation headquartered in Sunnyvale, California. It is widely known for its web portal, search engine Yahoo! Search, and related services, including Yahoo! Directory, Yahoo! Mail, Yahoo! News, Yahoo! Finance, Yahoo! Groups, Yahoo! Answers, advertising, online mapping, video sharing, fantasy sports and its social media website. It is one of the most popular sites in the United States.[

Products and services:

- Storing personal information and tracking usage
- Communication
- Content
- Mobile services
- Content
- Advertising

Hotmail:

Outlook.com (previously MSN Hotmail, Windows Live Hotmail and Hotmail) is a free web-based email service operated by Microsoft. Hotmail was one of the first web-based email services, it was founded by Sabeer Bhatia and Jack Smith and launched in July 1996 as "HoTMaiL". It was acquired by Microsoft in 1997 for an estimated $400 million, and shortly after, it was rebranded as "MSN Hotmail". The last version was released in 2011. In February 2013, it was renamed to Outlook.com as part of the rebranding of the Windows Live suite of products.

Features:

- Active view
- Conversation threading
- Skype Integration
- Instant actions

IV. SNORT OVERVIEW

Snort Modes:

Snort can run in three different modes:

Sniffer mode

Sniffer mode simply reads the packets off of the network and displays them in a continuous stream on the console.

Options:

./snort –v : Prints TCP/IP header onto screen (also for UDP/ICMP).

./snort –vd: Prints the application data too.

./snort –vde: Prints the data link layer contents as well.

Packet Logger mode

Packet logger mode logs the packets to the disk.

Options:

```
$sudo ./snort –dev –l ./log: or .C:\ ./snort –dev –l ./log:
```

Logs the packets to the directory specified.

```
$sudo  ./snort –l ./log –b      C:\ ./snort –l ./log –b
```

Binary log, binary file may be read back using –r switch

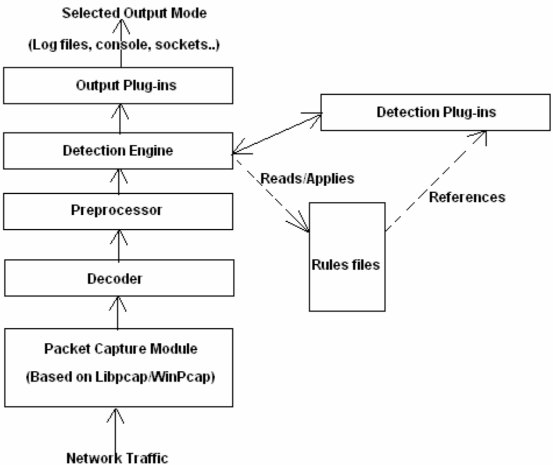
Network Intrusion Detection System

Network intrusion detection mode is the most complex and configurable mode allowing Snort to analyse network traffic for matches against a user defined rule set and perform several actions based upon what it sees.

Options:

```
./snort –A fast –c snort.conf
```

Architecture of Snort



Decoder:

It fits captured packets into data structures and identifies link level protocols. Then it takes the next level, decodes IP, and TCP/UDP to get information about port addresses. Snort alerts for malformed headers, unusual TCP option.

Preprocessors:

They are like filters, which identifies things that should be checked in Detection Engine module (like suspicious connection attempt to some TCP/UDP port or too many UDP packets received during a port-scan).

Rule files:

Text files with rule sets written with a known syntax.

Detection Plug-ins:

Those modules referenced from its definition in the rule files, and they are intended to identify patterns whenever a rule is evaluated.

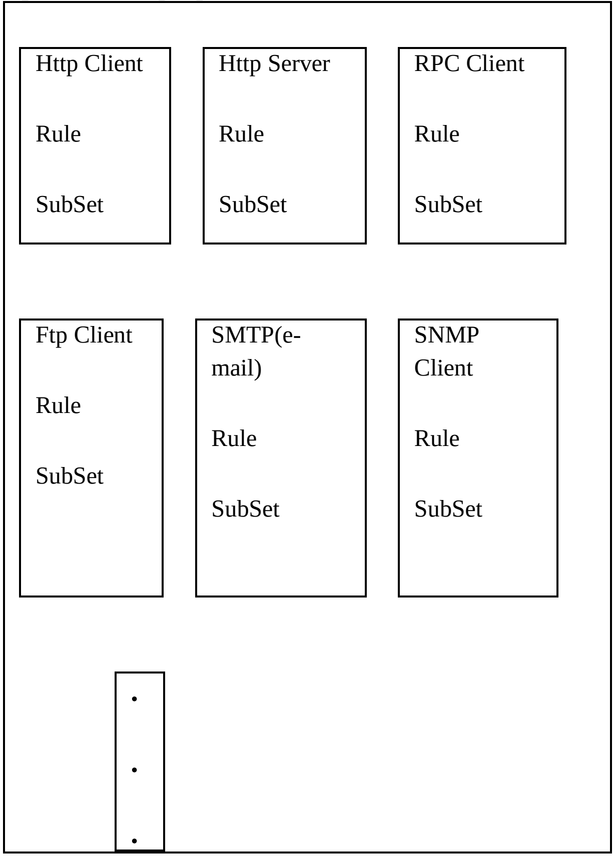
Detection engine:

Making use of detection plug-ins, it matches packets against rules previously charged into memory since snort initialization.

Output plug-ins: Alerts, logs, external files, databases.

Snort Rule Set:

The Rule classifier classifies all Snort rules into rule subsets. This is done prior to any packet streaming. Once this is over, each incoming packet is matched to a corresponding rule set based on the packet's unique parameters.



Previous Works Using Snort:

Snort As A Forensics Tool:

Much has been published regarding the open source intrusion detection system software known as Snort's What is less known is Snort's ability to read previously captured binary packet capture files from various network devices, process these files, and produce meaningful output for responders, analysts, investigators, and examiners. Snort users also have the ability to create customized rules and include within these rules any character-based or hexadecimal pattern of interest.

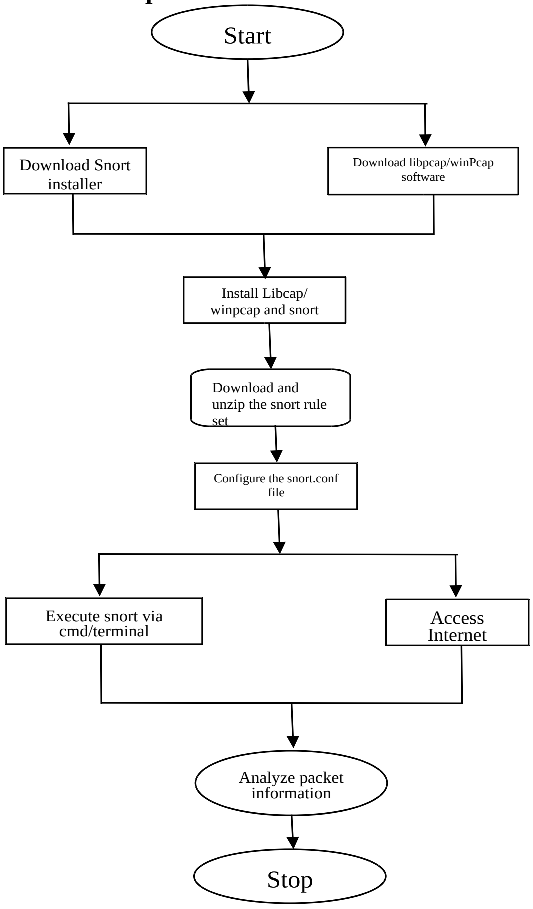
Snort by:

Unlike most network security monitoring applications, Snort by integrates with new and existing Solera DS Appliances, and Sera's Deep See installations to give analysts full packet and session data.

Sguil:

Sguil is built by network security analysts for network security analysts. Sguil's main component is an intuitive GUI that provides access to real time events, session data, and raw packet captures. Sguil facilitates the practice of network security monitoring and event driven analysis. The Sguil client is written in tcl/tk and can be run on any operating system that supports tcl/tk (including Linux, *BSD, Solaris, MacOS, and Win32).

Pictorial Representation


Start
Download Snort installer
Download libpcap/winPcap software
Install Libcap/ winpcap and snort
Download and unzip the snort rule set
Configure the snort.conf file
Execute snort via cmd/terminal
Access Internet
Analyze packet information
Stop

Snort Installation

Required Softwares:

- WinPcap/libpcap
- DAQ (Data acquisition)
- Snort Installer

WinPcap consists of:

Implementations of a lower-level library for the listed operating systems, to communicate with those drivers;

A port of libpcap that uses the API offered by the low-level library implementations

Libpcap:

Pcap (packet capture) consist of an application programming interface for capturing network traffic. Unix like systems implements pcap in the libpcap library.

Other Applications Using WinPcap/Libpcap:

- Snort
- Tcpdump
- Wireshark
- Ngrep

DAQ:

Data acquisition software is needed in order for the DAQ hardware to work with a PC. The device driver performs low-level register writes and reads on the hardware, while exposing a standard API for developing user applications. A standard API such as COMEDI allows the same user applications to run on different operating systems, e.g. a user application that runs on Windows will also run on Linux.

Snort Installer:

1. Install Snort

cd /usr/src

wget https://www.snort.org/downloads/snort/snort-2.9.7.0.tar.gz or zip

tar -zxf snort -2.9.7.6.tar.gz && cd snort -2.9.7.6

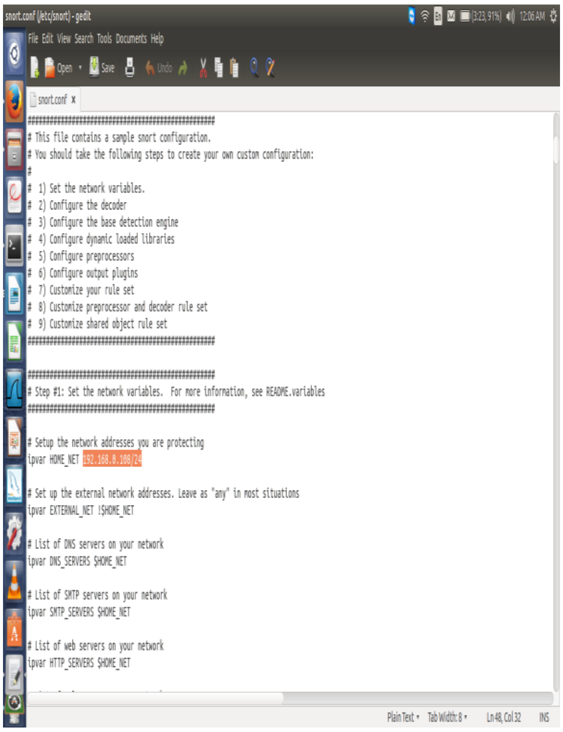
./configure --enable-sourcefire && make && make install

First, we will need to get the Snort Installer binary from Snort.org. For Snort to work properly, we will need to put our network interface card (NIC) into promiscuous mode where it can see all traffic flows to it.

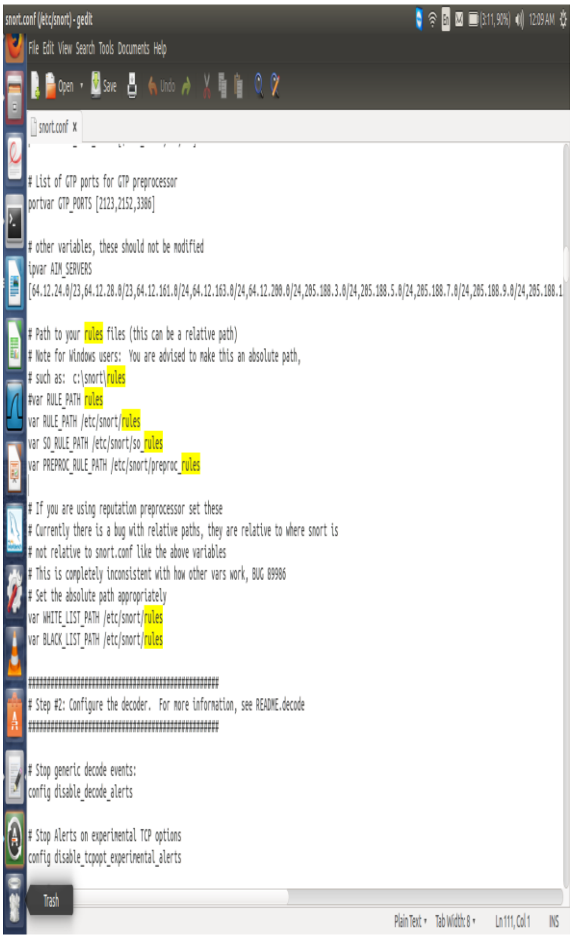
Snort's authorized web site is: http://www.snort.org. The site has links to the tools we will need to get snort up and running. A review of the tools available for Snort will reveal that many of them are designed to only run on *nix platforms. We will be installing version 2.9.7.6, which is the current stable version. The binary needed to install Snort can be found in the downloads section of the website,

As we were told by the Snort setup application, we will need to change a couple of parameters in the /usr/local/etc/snort/snort.conf

Open the snort.conf file and find the lines highlighted below:

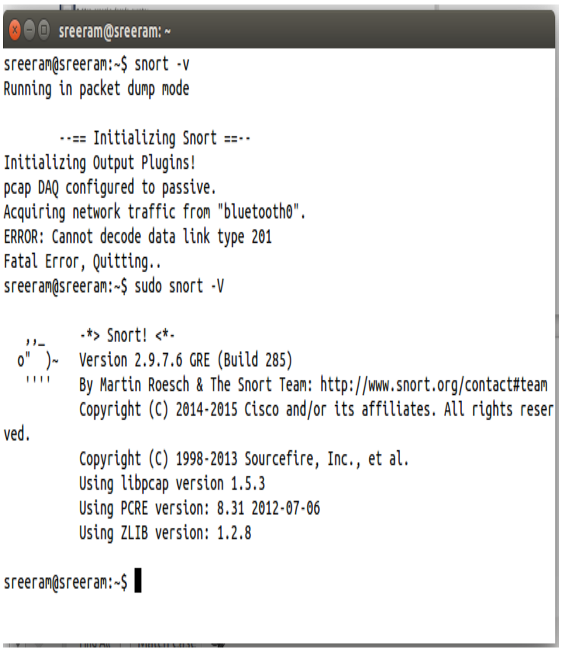

snort.conf (/etc/snort) - gedit
12:06 AM
File Edit View Search Tools Documents Help
Open
Save
Undo
snort.conf
Plain Text
Tab Width: 8
Ln 48, Col 32
INS

```
#################################################
# This file contains a sample snort configuration.
# You should take the following steps to create your own custom configuration:
#
#  1) Set the network variables.
#  2) Configure the decoder
#  3) Configure the base detection engine
#  4) Configure dynamic loaded libraries
#  5) Configure preprocessors
#  6) Configure output plugins
#  7) Customize your rule set
#  8) Customize preprocessor and decoder rule set
#  9) Customize shared object rule set
#################################################

#################################################
# Step #1: Set the network variables.  For more information, see README.variables
#################################################

# Setup the network addresses you are protecting
ipvar HOME_NET 192.168.0.100/24

# Set up the external network addresses. Leave as "any" in most situations
ipvar EXTERNAL_NET !$HOME_NET

# List of DNS servers on your network
ipvar DNS_SERVERS $HOME_NET

# List of SMTP servers on your network
ipvar SMTP_SERVERS $HOME_NET

# List of web servers on your network
ipvar HTTP_SERVERS $HOME_NET
```

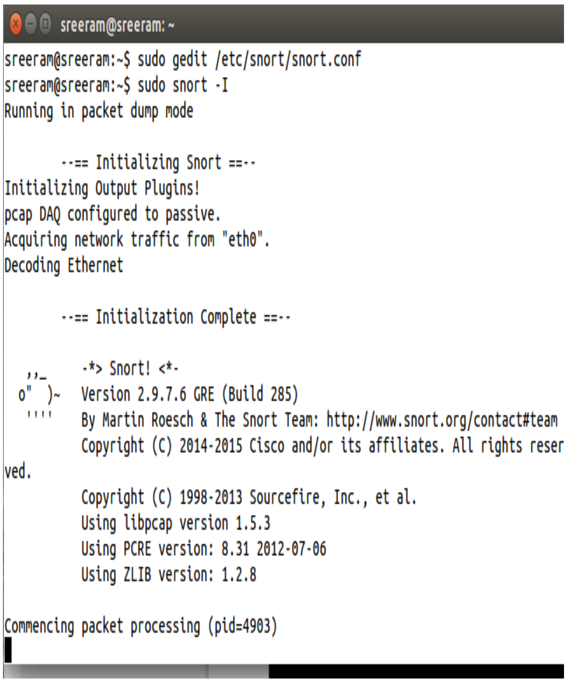

snort.conf (/etc/snort) - gedit
12:09 AM
File Edit View Search Tools Documents Help
Open
Save
Undo
snort.conf
Trash
Plain Text
Tab Width: 8
Ln 111, Col 1
INS

```
# List of GTP ports for GTP preprocessor
portvar GTP_PORTS [2123,2152,3386]

# other variables, these should not be modified
ipvar AIM_SERVERS
[64.12.24.0/23,64.12.28.0/23,64.12.161.0/24,64.12.163.0/24,64.12.200.0/24,205.188.3.0/24,205.188.5.0/24,205.188.7.0/24,205.188.9.0/24,205.188.1

# Path to your rules files (this can be a relative path)
# Note for Windows users:  You are advised to make this an absolute path,
# such as:  c:\snort\rules
#var RULE_PATH rules
var RULE_PATH /etc/snort/rules
var SO_RULE_PATH /etc/snort/so_rules
var PREPROC_RULE_PATH /etc/snort/preproc_rules

# If you are using reputation preprocessor set these
# Currently there is a bug with relative paths, they are relative to where snort is
# not relative to snort.conf like the above variables
# This is completely inconsistent with how other vars work, BUG 89986
# Set the absolute path appropriately
var WHITE_LIST_PATH /etc/snort/rules
var BLACK_LIST_PATH /etc/snort/rules

#################################################
# Step #2: Configure the decoder.  For more information, see README.decode
#################################################

# Stop generic decode events:
config disable_decode_alerts

# Stop Alerts on experimental TCP options
config disable_tcpopt_experimental_alerts
```

The following is output from the command on Ubuntu.



```
sreeram@sreeram:~$ snort -v
Running in packet dump mode

        --== Initializing Snort ==--
Initializing Output Plugins!
pcap DAQ configured to passive.
Acquiring network traffic from "bluetooth0".
ERROR: Cannot decode data link type 201
Fatal Error, Quitting..
sreeram@sreeram:~$ sudo snort -V

   ,,_     -*> Snort! <*-
  o"  )~   Version 2.9.7.6 GRE (Build 285)
   ''''    By Martin Roesch & The Snort Team: http://www.snort.org/contact#team
           Copyright (C) 2014-2015 Cisco and/or its affiliates. All rights reser
ved.
           Copyright (C) 1998-2013 Sourcefire, Inc., et al.
           Using libpcap version 1.5.3
           Using PCRE version: 8.31 2012-07-06
           Using ZLIB version: 1.2.8

sreeram@sreeram:~$ 
```

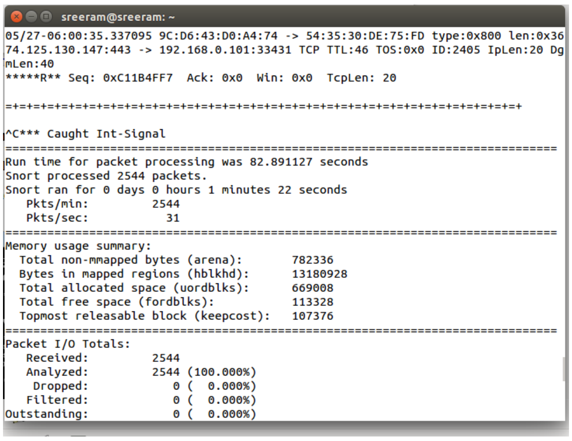


```
sreeram@sreeram:~$ sudo gedit /etc/snort/snort.conf
sreeram@sreeram:~$ sudo snort -I
Running in packet dump mode

        --== Initializing Snort ==--
Initializing Output Plugins!
pcap DAQ configured to passive.
Acquiring network traffic from "eth0".
Decoding Ethernet

        --== Initialization Complete ==--

   ,,_     -*> Snort! <*-
  o"  )~   Version 2.9.7.6 GRE (Build 285)
   ''''    By Martin Roesch & The Snort Team: http://www.snort.org/contact#team
           Copyright (C) 2014-2015 Cisco and/or its affiliates. All rights reser
ved.
           Copyright (C) 1998-2013 Sourcefire, Inc., et al.
           Using libpcap version 1.5.3
           Using PCRE version: 8.31 2012-07-06
           Using ZLIB version: 1.2.8

Commencing packet processing (pid=4903)
```

V. SNORT IN SNIFFER MODE:

To analyse the data produced by snort in sniffer mode, we have to follow certain steps. Initially, we have to run snort using the following command

sreeram@sreeram:~$ sudo snort -dev
or
c:\snort –dev



```
05/27-06:00:35.337095 9C:D6:43:D0:A4:74 -> 54:35:30:DE:75:FD type:0x800 len:0x36
74.125.130.147:443 -> 192.168.0.101:33431 TCP TTL:46 TOS:0x0 ID:2405 IpLen:20 Dg
mLen:40
*****R** Seq: 0xC11B4FF7  Ack: 0x0  Win: 0x0  TcpLen: 20

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

^C*** Caught Int-Signal
===============================================================================
Run time for packet processing was 82.891127 seconds
Snort processed 2544 packets.
Snort ran for 0 days 0 hours 1 minutes 22 seconds
   Pkts/min:         2544
   Pkts/sec:           31
===============================================================================
Memory usage summary:
  Total non-mmapped bytes (arena):       782336
  Bytes in mapped regions (hblkhd):      13180928
  Total allocated space (uordblks):      669008
  Total free space (fordblks):           113328
  Topmost releasable block (keepcost):   107376
===============================================================================
Packet I/O Totals:
   Received:         2544
   Analyzed:         2544 (100.000%)
    Dropped:            0 (  0.000%)
   Filtered:            0 (  0.000%)
Outstanding:            0 (  0.000%)
```

In the above screenshot, we can observe the packets analyzed by Snort while accessing the Internet.

VI. EXPERIMENTAL RESULTS

MAIL CLIENTS:

A) TOTAL RECEIVED PACKETS:

	50kb	100kb	500kb	1mb	2mb
Gmail	225	295	776	1870	2835
Yahoo	400	598	1041	2263	3241
Hotmail	190	258	807	1788	2821

Graph:

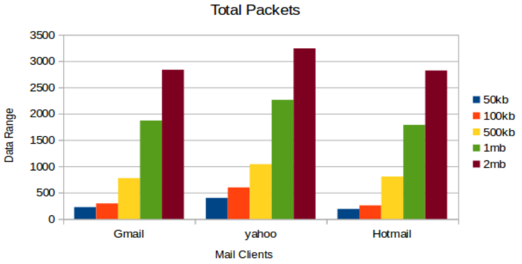


Fig: Total received packets for mail clients

B) ANALYZED PACKETS FOR MAIL CLIENTS:

	50kb	100kb	500kb	1mb	2mb
Gmail	225	295	776	1866	2612
Yahoo	397	598	1041	2257	3223
Hotmail	190	258	796	1788	2475

Graph:

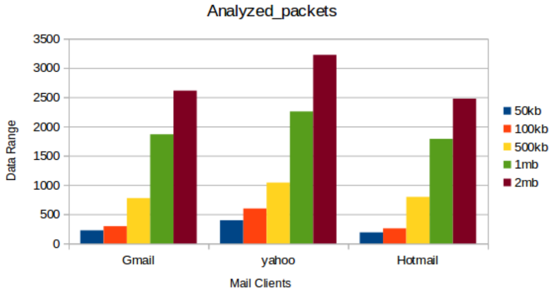


Fig: Analysed packets for mail clients

C) RUNTIME FOR MAIL CLIENTS:

	50kb	100kb	500kb	1mb	2mb
Gmail	24.238	25.167	35.827	56.544	65.23
Yahoo	29.568	31.95	42.153	56.66	71.36
Hotmail	28.48	22.8	31.15	55.52	60.93

Graph:

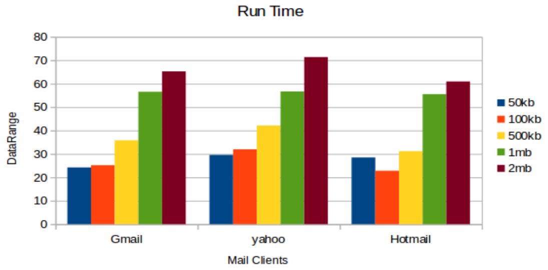


Fig: Runtime for mail clients

VII. CONCLUSIONS

Using Snort, we analysed the performance of different mail clients by varying the text sizes from 50 KB to 2 MB and analysed the metrics (run time, analysed packets and total packets).From simulation scenario, we found that Hotmail is best for sending larger text and Yahoo should be less preferred for the same purpose.

VIII. FUTURE ENHANCEMENT

In this paper, we analysed the performance of different mail clients by varying the data size using snort. As a future scope of work, we can apply snort for different encryption algorithms using compression techniques.

IX. REFERENCES

[1] Ahmed Patel, Qais Qassim, Christopher Wills. A survey of intrusion detection and prevention systems, Information Management & Computer Security Journal (2010).

[2] Oludele Awodele, Sunday Idowu, Omotola Anjorin, and Vincent J. Joshua, A Multi-Layered Approach to the Design of Intelligent Intrusion Detection and Prevention System (IIDPS), Babcock University, (Volume 6, 2009).

[3] Host Intrusion Prevention Systems and Beyond, SANS Institute (2008).

[4] Intrusion Detection and Prevention In-sourced or Out-sourced, SANS Institute (2008).

[5] Mario Guimaraes, Meg Murray. Overview of Intrusion Detection and Intrusion Prevention, Information security curriculum development Conference by ACM (2008).

[6] Muhammad Awais Shibli, Sead Muftic. Intrusion Detection and Prevention System using Secure Mobile Agents, IEEE International Conference on Security & Cryptography (2008).

[7] David Wagner, Paolo Soto. Mimicry Attacks on Host Based Intrusion Detection Systems, 9 th ACM Conference on Computer and Communications Security (2002).

[8] Harley Kozushko. Intrusion Detection: Host-Based and Network-Based Intrusion Detection Systems, (2003).

[9] Lin Tan, Timothy Sherwood. A High Throughput String Matching Architecture for Intrusion Detection and Prevention, Proceedings of the 32 nd Annual International Symposium on Computer Architecture (ISCA 2005).

[10] S. Mrdovic, E. Zajko. Secured Intrusion Detection System Infrastructure, University of Sarajevo/Faculty of Electrical Engineering, Sarajevo, Bosnia and Herzegovina (ICAT 2005).

[11] Yeubin Bai, Hidetsune Kobayashi. Intrusion Detection Systems: technology and Development, 17 th International Conference of Advanced Information Networking and Applications, (AINA 2003).

[12] Sang-Jun Han and Sung-Bae Cho. Combining Multiple Host-Based Detectors Using Decision Tree, Australian Joint Artificial Intelligence Conference, (AUSAI 2003).

[13] Ramaprabhu Janakiraman, Marcel Waldvogel, Qi Zhang. Indra: A peer-to-peer approach to network intrusion detection and prevention, Enabling Technologies: Infrastructure for Collaborative Enterprises, WET ICE 2003.

[14] M. Laureano, C. Maziero1, E. Jamhour. Protecting Host-Based Intrusion Detectors through Virtual Machines, The International Journal of Computer and Telecommunications Networking (2007).

[15] Matt Carlson and Andrew Scharlott. Intrusion detection and prevention systems, (2006).