****

**STUDY & ANALYSIS OF SNORT**

*A report submitted*

*To*

**NATIONAL INSTITITE OF TECHNOLOGY, KURUKSHETRA**

*For Partial Fulfillment of the Requirement for the*

*Award of the Degree*

*Of*

**BACHELOR OF ENGINEERING**

*In*

**INFORMATION SECURITY**

*By*

**LAKHSAY HURRIA**

**Roll No: 1140444**

**JAMOLIDDIN FASIKHIDDINOV**

**Roll No: 2140029**

**MUJAHID OMER**

**Roll No: 2140003**

**Abstract**

Malicious network traffic (such as worms, hacking attempts, etc.) has certain patterns to it.You could monitor your network traffic with a sniffer and look for this malicious traffic manually, but that would be an impossible task. IDS (Intrusion Detection System) software which automates the process of sniffing, examining, and upon finding something suspicious, alerting.

IDS have been called the burglar alarm of computer networks and are an important part of network perimeter security. Without IDS you have no idea if someone is probing or attacking your servers (unless the attack is so overwhelming that it results in a denial of service). Having this information can let you know if you need to make some firewall changes or harden the OS on a particular server a bit more.   
  
You may see the term IPS for Intrusion *Prevention* Systems which takes things one step further, having the IDS adjust the firewall when it discovers something. Smart people disagree on the use of IPSs as it, in effect, gives an attacker some control of your firewall.   
  
**Snort** ([www.snort.org](http://www.snort.org/)) is the most widely-used IDS software application and it's open source and included with Debian. There are two flavors of IDSs, host-based and network-based. Snort is a network-based IDS that can monitor all of the traffic on a network link to look for suspicious traffic. Typically, a network-based IDS is set up to monitor a DMZ or the internal network right behind the firewall so it alerts to any possible threats that your firewall didn't catch.   
  
There is a Web interface that works with Snort called BASE (Basic Analysis and Security Engine) which is based on ACID (Analysis Console for Intrusion Databases) which we'll set up. BASE uses what's commonly referred to as a LAMP server (Linux, Apache, MySQL, PHP) so we'll need to install those applications as well.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ABSTRACT** | | | | | | **1** |
|  | | | | | | |
| **Chapter 1** | | | | | **INTRODUCTION** | **3** |
|  | 1.1 | | Intrusion Detection Systems with Snort | | | 3 |
|  | 1.2 | | Motivation | | | 6 |
|  | 1.3 | | Objective | | | 7 |
|  | | | | | | |
| **Chapter 2** | | | | | **INSTALLTION** | **8** |
|  | 2.1 | | | Softwares Installed Before Snort | | 8 |
|  | 2.2 | | | Installing Snort Rules And Configuration File | | 9 |
|  | 2.3 | | | Configure Snort System File | | 10 |
|  | 2.4 | | | Setting Up The Database In MySQL | | 10 |
|  | 2.5 | | | Softwares Installed After Snort | | 11 |
|  | 2.6 | | | Check to see If Everything is Working | | 11 |
|  | | | | | | |
| **Chapter 3** | | | | | **COMPONENTS OF SNORT** | **13** |
|  | | 3.1 | | Packet Decoder | | 14 |
|  | | 3.2 | | Preprocessors | | 14 |
|  | | 3.3 | | The Detection Engine | | 15 |
|  | | 3.4 | | Logging and Alerting System | | 16 |
|  | | 3.5 | | Output Modules | | 16 |
|  | | | | | | |
| **Chapter 4** | | | | | **SNORT MODES** | **18** |
|  | 4.1 | | | Network Sniffer Mode | | 18 |
|  | 4.2 | | | Logging Snort Data in Text Format | | 21 |
|  | 4.3 | | | Network Intrusion Detection Mode | | 22 |
|  | | | | | | |
| **Chapter 5** | | | | | **WORKING WITH RULES** | **24** |
|  | 5.1 | | | Structure of a Rule | | 24 |
|  | 5.2 | | | Using ACID With Snort | | 25 |
|  | | | | | | |
| **CONCLUSION** | | | | | | **32** |
| **REFERENCES** | | | | | | **33** |

**CHAPTER 1: INTRODUCTION**



Intrusion Detection Systems with Snort

Security is a big issue for all networks in today's enterprise environment. Many methods have been developed to secure the network infrastructure and communication over the Internet. One relatively new method is intrusion detection methods, which started appearing in the last few years. Using intrusion detection methods, you can collect and use information from known types of attacks and find out if someone is trying to attack your network or particular hosts.

A comprehensive security system consists of multiple tools, including:

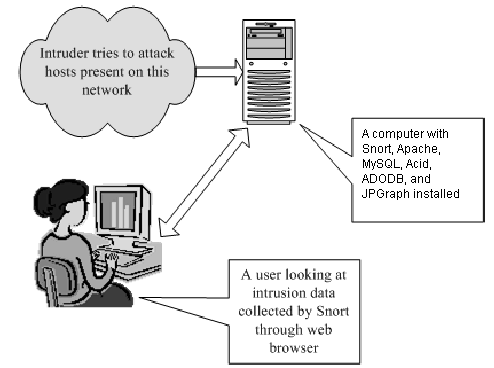
* Firewalls that are used to block unwanted incoming as well as outgoing traffic of data.
* Intrusion detection systems (IDS) that are used to find out if someone have gotten into or are trying to get into your network.
* Vulnerability assessment tools that are used to find and plug security holes present in your network.

These tools can work together and exchange information with each other. Intrusion Detection System (IDS) is software, hardware or combination of both used to detect intruder activity. There are two types of IDS. One is Network Intrusion Detection System (NIDS) that capture data packets traveling on the network media and match them to a database of signatures. The other is Host-based Intrusion Detection System (HIDS) that are installed as agents on a host. This system can look into system and application log files to detect any intruder activity. Snort is an open source NIDS which is available free of cost.

In this report, I will discuss installation procedure for Snort as well as other products that work with Snort, components of Snort, most frequently used functions and testing of Snort/ACID.

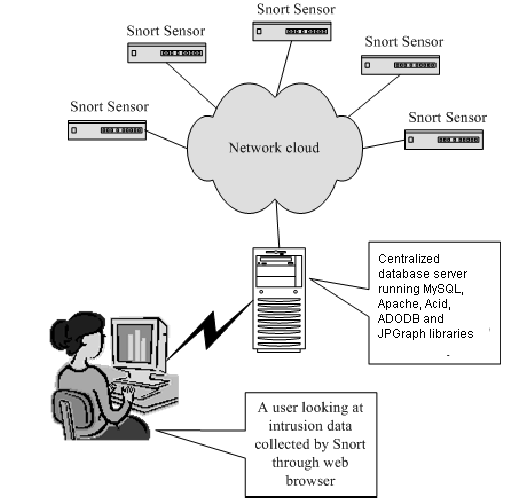
In a Snort based Intrusion Detection System, first Snort captured and analyze data. Then, it stores this data in the MySQL database using the database output plug-in. Apache web server takes help from ACID, PHP, ADODB and JPGraph packages to display the data in a browser window when a user connects to Apache. A user can then make various type of operations to analyze, archive, graph and delete data.

In essence, an IDS building on a single computer is shown in Figure 1



**Figure1:** **A network intrusion detection system in a single machine**

In the enterprise environment, multiple Snort sensors are used behind every router or firewall. In that case, a single centralized database is used to collect data from all of the sensors. Figure2 shows the architecture used in such a system.



**Figure2: Multiple Snort sensors logging to a centralized database server**

**1.2 MOTIVATION**

Intrusion detection and prevention is necessary for security of any network. Initially, firewall was considered essential to provide security for the network but now Intrusion Detection System (IDS) and Intrusion Prevention System (IPS) are the mainstream devices along with firewall. Nowadays intrusion prevention is one of the fundamental requirements of any big organization.

**1.3 OBJECTIVE**

To capture SNORT & analyze data which then stores this data in MySQL Database. We then use ACID to display the data in a browser when a user connects to apache.

## **CHAPTER 2: Installation**

Snort can be downloaded from it’s official website at <http://www.snort.org>. Snort is supported on a number of hardware platforms and operating systems. These include:

* Linux
* OpenBSD
* FreeBSD
* NetBSD
* Solaris (both Sparc and i386)
* HP-UX
* AIX
* IRIX
* MacOS
* Windows

In this report, we downloaded the source file package and compiled from scratch, however, Snort website also provide binary version for linux and windows platforms.

**2.1 Softwares Installed Before Snort**

Before compile and install snort, we have to first install the related software and libraries, this includes:

* Zlib1.2.1 (<http://www.gzip.org/zlib/>)
* LibPcap0.7.2 (<http://www.tcpdump.org/release/libpcap-0.7.2.tar.gz>)
* MySQL4.0.15 (<http://downloads.mysql.com/archives.php?p=mysql-4.0&o=linux->)
* Apache2.0.52 (<http://httpd.apache.org/download.cgi>)
* PHP4.3.3 (<http://www.php.net/distributions/php-4.3.3.tar.gz>)

For detailed description about the installation of these softwares, please read the article: ”Snort, Apache, PHP, MySQL and Acid Install on RH9.0” (<http://www.infosecwriters.com/text_resources/pdf/snort_acid_rh9.pdf>).

The following is the installation procedures on RedHat9.0 linux operating system:

shell> tar –xzvf snort-2.2.0 .tar.gz

shell> cd snort-2.2.0

shell> ./configure –with-mysql=/usr/local/mysql

shell> make

shell> make install

**2.2 Installing Snort Rules And Configuration File**

After successfully installed Snort, we then need to install the rules and configure file used for Snort. This can be down like this:

shell> mkdir /etc/snort

shell> mkdir /var/log/snort

shell> cd rules

shell> cp \* /etc/snort

shell> cd ../etc

shell> cp snort.conf /etc/snort

shell> cp \*.config /etc/snort

**2.3 Configure Snort System File**

The Snort system file snort.conf is now located in /etc/snort, we need to make the following changes to make it works well on our system.

var HOME\_NET 192.168.0.0/24 (make this whatever your internal network is)

var RULE\_PATH /etc/snort/ (this is the location where the snort rules are stored)

output database: log, mysql, user=snort password=xxx dbname=snort host=localhost (this is to tell Snort to log to the mysql database)

**2.4 Setting Up The Database In MySQL**

Open MySQL client, type the following commands:

mysql> set password for root@localhost=password(‘xxx’);

mysql> create database snort;

mysql > grant insert, select on root,.\* to snort@localhost;

mysql> set password for snort@localhost=password(‘xxx’);

mysql> grant create, insert, select, delete, update on snort.\* to snort@localhost;

mysql> grant create, insert, delete, select, update on snort.\* to snort;

mysql> exit

From the Snort source directory execute the following command:

shell> /usr/local/mysql/bin/mysql –u root –p < ./contrib./create\_mysql snort

Enter password: xxx (which is the password set above for root account)

Finally, install the extra DB tables using the following command from the contrib directory:

shell> zcat snortdb-extra.gz | /usr/local/mysql/bin/mysql –p snort

Enter password: xxx

**2.5 Softwares Installed After Snort**

In order to display alert massages generated by Snort when capturing prospective intrusion actions, we need to install Analysis Console for Intrusion Detection (Acid) software package (<http://acidlab.sourceforge.net/acid-0.9.6b23.tar.gz>) as well as related libraries JPGraph (<http://www.aditus.nu/jpgraph/downloads/jpgraph-1.1.3.ar.gz>) and ADODB (<http://phplens.com/lens/dl/adodb390.tgz>) and copy them to the htdocs directory of Apache web server. For detailed description about the installation of these softwares, please read the article: ”Snort, Apache, PHP, MySQL and Acid Install on RH9.0” (<http://www.infosecwriters.com/text_resources/pdf/snort_acid_rh9.pdf>).

**1.6 Check to See If Everything Is Working**

To check if you have successfully installed the above softwares and build a Snort-based IDS, you can execute the following commands:

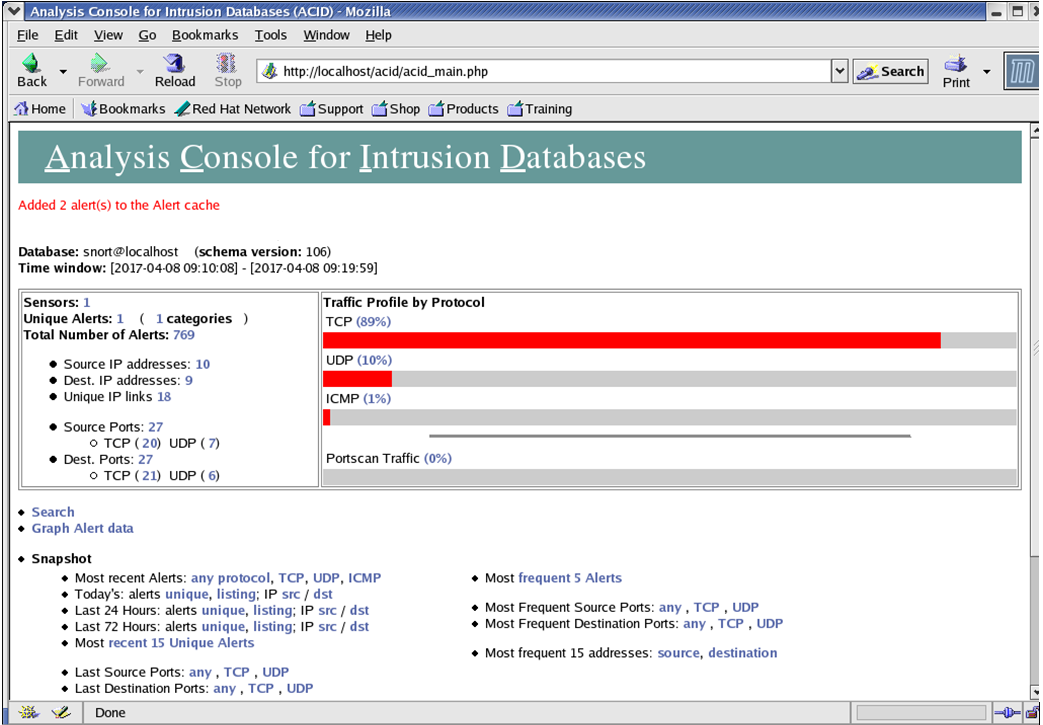
shell> /usr/local/apache/bin/apachectl start (start Apache web server)

shell> /usr/local/mysql/bin/mysqld\_safe & (start MySQL database in the background)

shell> /usr/local/bin/snort –c /etc/snort/snort.conf –D (star Snort in the background)

shell> ping yahoo.com

Then, type the following address in your favorite web browser: <http://192.168.0.2/acid/>. If you can see the main page of Acid, and it is displaying the alerts from MySQL database generated by Snort, then congratulations, you have successfully installed a Snort-based IDS on a single machine.



### **Figure3: Acid main page displaying the alerts from MySQL database**

# CHAPTER 3: Components of Snort

A Snort-based IDS contains the following components:

* Packet Decoder
* Preprocessors
* Detection Engine
* Logging and Alerting System
* Output Modules

Figure4 shows how these components work together to detect particular attacks and to generate output. Any data packet coming from the Internet enters the packet decoder. On its way towards the output modules, it is either dropped, logged or an alert is generated.

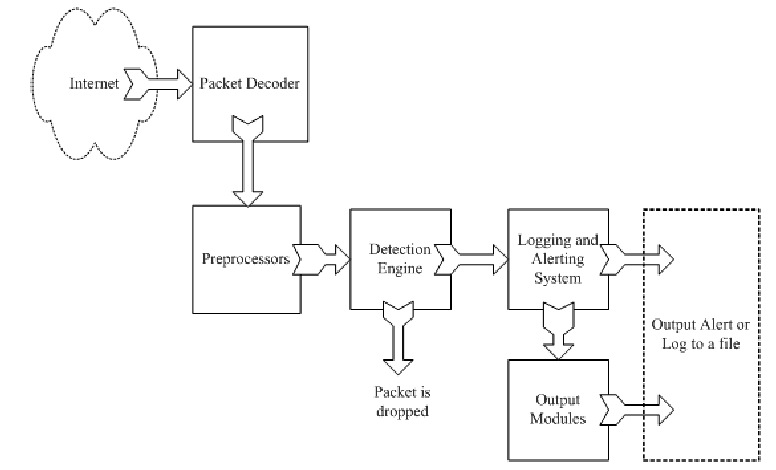


Figure4: Components of Snort

**3.1 Packet Decoder**

In an organization’s network, the network interface can be Ethernet, SLIP, or PPP and so on. The packet decoder takes packets from different types of network interfaces and prepares the packets to be preprocessed or to be sent to the detection engine.

**3.2 Preprocessors**

Preprocessors are components that can be used to arrange or modify data packets before the detection engine does some operation to find out if the packet is being used by an intruder. Hackers use different techniques to fool an IDS in different way, these may include:

* **Exact match:** For examples, you may have created a rule to find a signature “httpd/conf” in HTTP packets. If you are matching this string exactly, a hacker can easily fool you by modifying the string as “httpd/./conf” or “httpd../httpd/conf”. In addition, since the web servers usually understand hexadecimal characters or Unicode characters, hackers can also inset these kinds of characters into the web URI. This also makes the exact match rule void. A preprocessor can rearrange the string so that it is detectable by the IDS.
* **Packets fragmentation:** When a large data chunk is transferred to a host, the packet is usually fragmented into multiple data packets so that each packet fragment satisfies the requirement of default maximum length of some particular network interface. Later, the receiving systems need to reassemble these smaller units to form the original data packets. Hackers can use fragmentation to hide a signature into several small units to fool the IDS. A Preprocessor can reassemble these small units first and send the whole packet to the detection engine for signature testing.

**3.3 The Detection Engine**

The detection engine is the most important part of Snort. Its responsibility is to detect if any intrusion activity exists in a packet. The detection engine employs Snort rules for this purpose. The rules are read into internal data structures or chains where they are matched against all packets. If a packet matches any rule, Snort will send alerts to system logs or output plug-ins; otherwise the packet is dropped. The detection engine is the time-critical part of Snort. The load on it depends on the following factors:

* Number of rules
* Power of the machine on which Snort is running
* Speed of internal bus used in the Snort machine
* Load on the network

Note that the detection engine can dissert a packet and apply rules on different parts of the packet. These parts may be:

* The IP header of the packet
* The Transport layer header: This header includes TCP, UDP or other transport layer headers. It may also work on the ICMP header.
* The application layer level header: Application layer headers include, but are not limited to, DNS header, FTP header, SNMP header, and SMTP header. You may have to use some indirect methods for application layer headers, like offset of data to be looked for.
* Packet payload. This means that you can create a rule that is used by the detection engine to find a string inside the data that is present inside the packet.

**3.4 Logging and Alerting System**

Depending on what the detection engine finds inside a packet, the packet may be used to log the activity or generate an alert. Logs are kept in simple text files, tcpdump-style files or some other form. All of the log files are stored under /var/log/snort folder by default. You can use –l command line options to modify the location of generating logs and alerts. Many command line options discussed in the next chapter can modify the type and detail of information that is logged by the logging and alerting system.

**3.5 Output Modules**

Output modules or plug-ins can do different operations depending on how you want to save output generated by the logging and alerting system of Snort. Basically these modules control the type of output generated by the logging and alerting system. Depending on the configuration, output modules can do things like the following:

* Simply logging to /var/log/snort/alerts file or some other file
* Sending SNMP traps
* Sending messages to syslog facility
* Logging to a database like MySQL or Oracle.
* Generating eXtensible Markup Language (XML) output
* Modifying configuration on routers and firewalls
* Sending Server Message Block (SMB) messages to Microsoft Windows-based machines

Other tools can also be used to send alerts in other formats such as e-mail messages or viewing alerts using a web interface.

#### **CHAPTER 4: Snort Modes**

Snort operates in two basic modes:

* Sniffer mode
* NIDS mode

It can be used as a packet sniffer, like tcpdump or snoop. When sniffing packets, Snort can also log these packets to a log file. The file can be viewed later on using Snort or tcpdump. No intrusion detection activity is done by Snort in this mode of operation. Using Snort for this purpose is not very useful as there are many other tools available for packet logging. For example, all Linux distributions come with the tcpdump program which is very efficient.

When you use Snort in network intrusion detection (NIDS) mode, it uses its rules to find out if there is any network intrusion detection activity.

**4.1 Network Sniffer Mode**

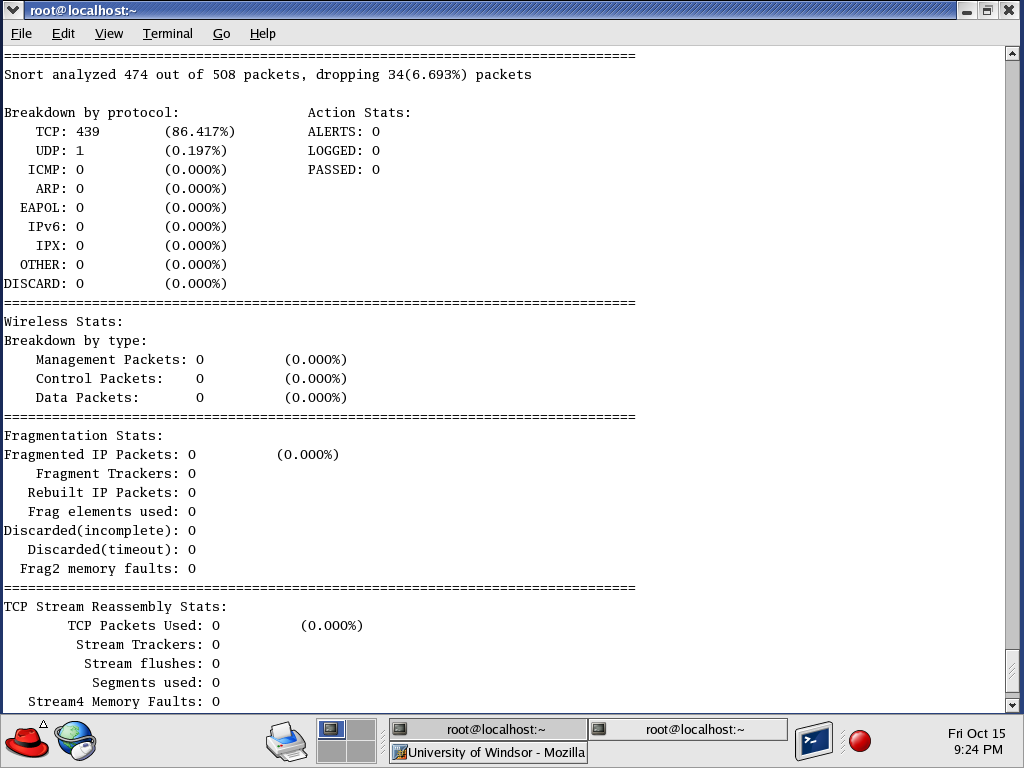
In the network sniffer mode, Snort acts like the commonly used program tcpdump. You don’t need a configuration file to run Snort in the sniffing mode. With the capacity of parameters, you can capture and display network packets with different levels of detail.

Figure5 is a network sniffing mode test. In this test, I start Snort server that is running as a network sniffer by typing the following command:

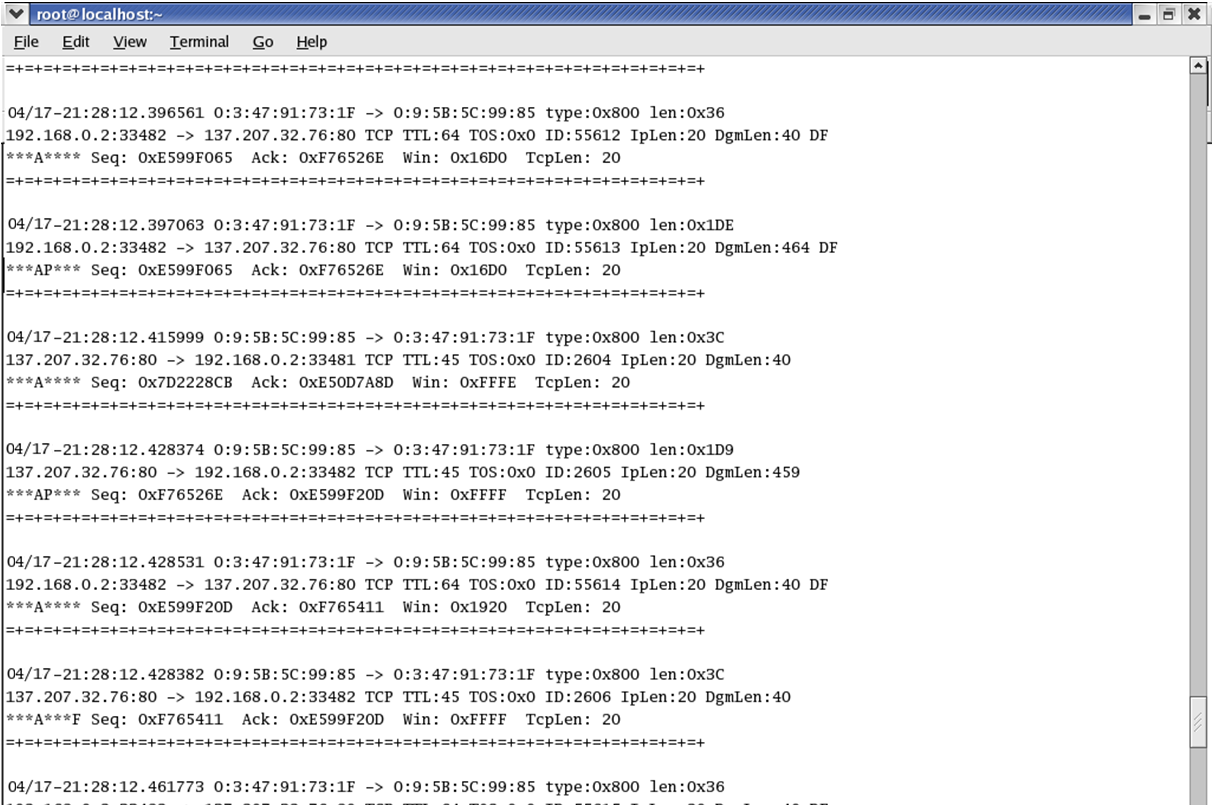
Shell> snort –v

Shell> ping www.yahoo.com

Snort will continue to display captured packets on the screen until you break using Ctrl-C. At the time Snort terminates, it will display statistical information.

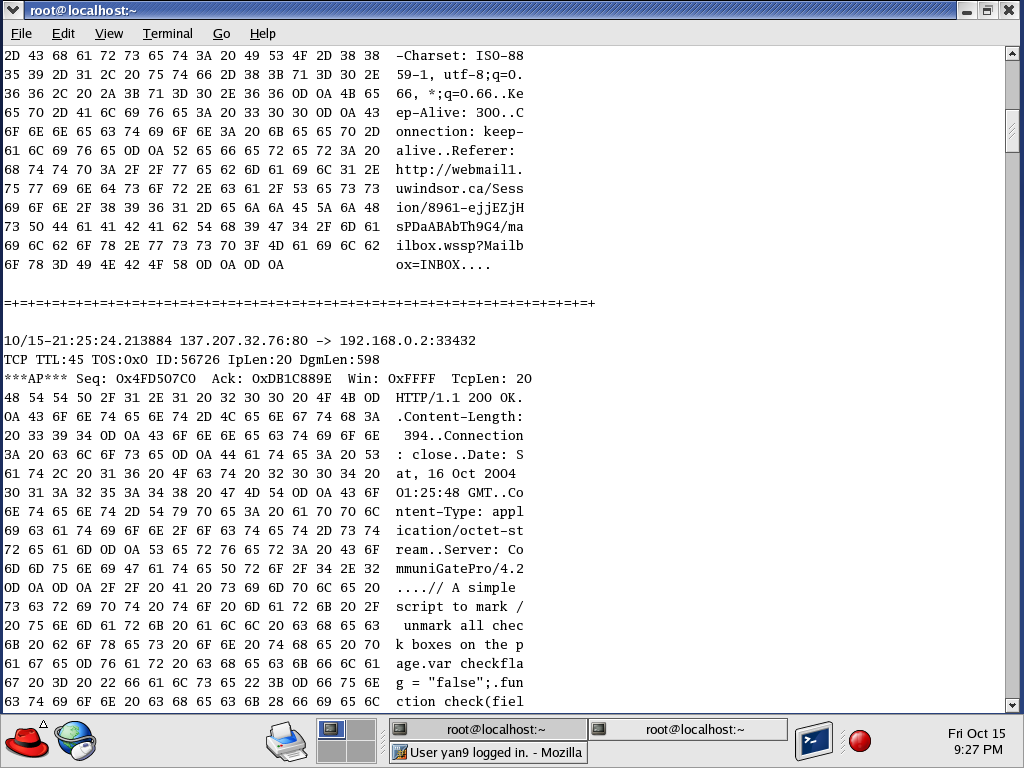
**Figure6: statistical information generated when terminating Snort**

To get the layer 2 information as well as the headers, you can start Snort by using parameter dv. The output is like this:



**Figure7: packets captured by Snort using sniffing mode with parameter dv**

Note the above information is not all the packets information on the network. In order to capture all of them, showing the data link layer headers and application layer payload, you have to use parameter dev. This command displays captured data in hexadecimal as well as ASCII format.

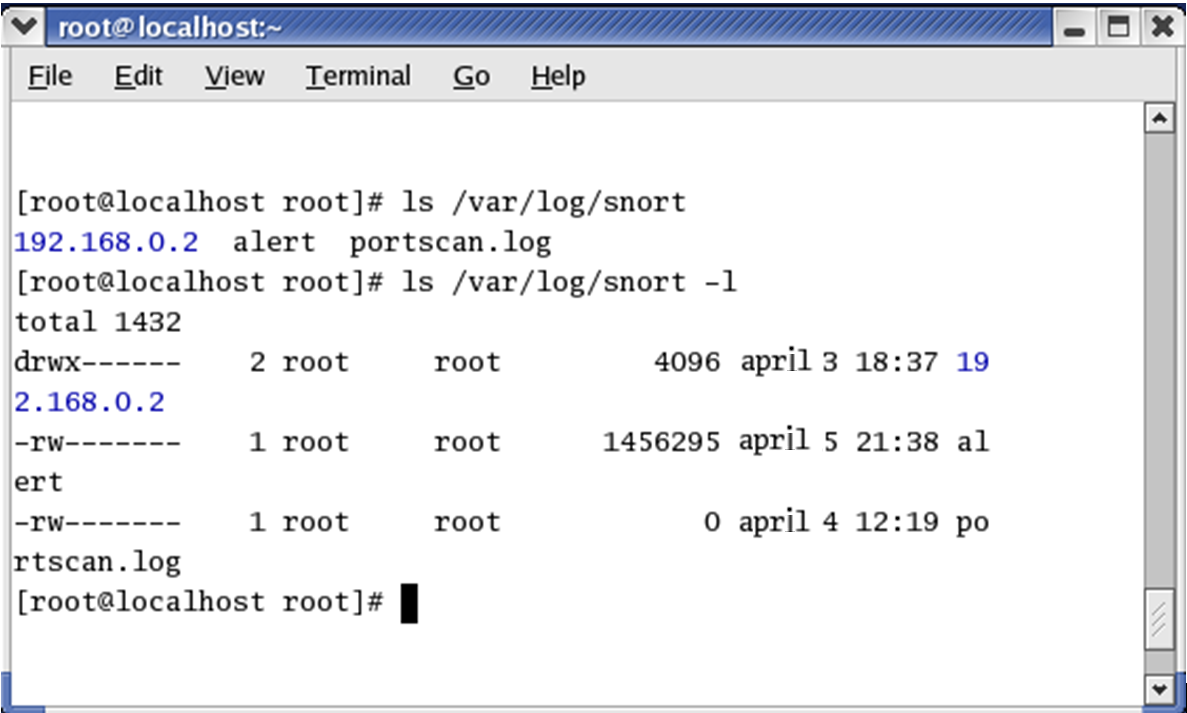
**Figure8: packets captured by Snort using sniffing mode with parameter dev**

**4.2 Logging Snort Data in Text Format**

You can log Snort data in text mode by adding -l <directory name> on the command line. The following command logs all Snort data in /var/log/snort directory in addition to displaying it on the console.

snort -dev -l /var/log/snort

When you go to the /var/log/snort directory, you will find multiple directories under it. Each of these directories corresponds to one host and contains multiple files. The name of the directory is usually the same as the IP address of host. These files contain logs for different connections and different types of network data. For example, files containing TCP data will start with TCP. A typical name for a file containing TCP data is TCP:2489-23. A typical file containing ICMP data may be ICMP\_ECHO. The format of data logged in these files is the same as the data displayed on the screen when you run Snort in the network sniffer mode.



# Figure9: log files generated inside the /var/log/snort directory

**4.3 Network Intrusion Detection Mode**

In intrusion detection mode, Snort does not log each captured packet as it does in the network sniffer mode. Instead, it applies rules on all captured packets. A packet is captured only if it matches a rule. If it is not match any rule, the packet is dropped silently and no log entry is created. The following command starts Snort in the NID mode:

Shell> snort –c /etc/snort/snort.conf

When you start this command, Snort will read the configuration file /etc/snort/snort.conf and all other files included in this file. Typically these files contain Snort rules and configuration data. After reading this files, all captured packets will be matched against these rules and appropriate action will be taken. These include sending alerts to database, Syslog, SNMP traps and MS Windows etc.

#### **CHAPTER 5: Working with Rules**

Most intruder activity will leave some kind of footprint, this footprint is called signature in the Snort jargon. Snort rules are created based on these signatures. These signatures may be present in the header parts of a packet or in the payload. Snort rules can be used to check various parts of a data packet including both network layer protocols and application layer protocols.

**5.1 Structure of a Rule**

A Snort rule is divided into two parts: rule header and rule options. The rule header contains information about what action a rule takes. It also contains criteria for matching a rule against data packets. The general structure of a Snort rule header is shown in Figure10.



**Figure10: structure of Snort rule header**

The options part contains additional criteria for matching a rule against data packets. A rule may detect one type or multiple types of intrusion activity. Intelligent rules should be able to apply to multiple intrusion signatures.

Rule options are enclosed inside a pair of parentheses. There may be one or many and the options are separated with a semicolon. In general, an option may have two parts: a keyword and an argument. Arguments are separated from the option keyword by a colon.

**4.2 Using ACID With Snort**

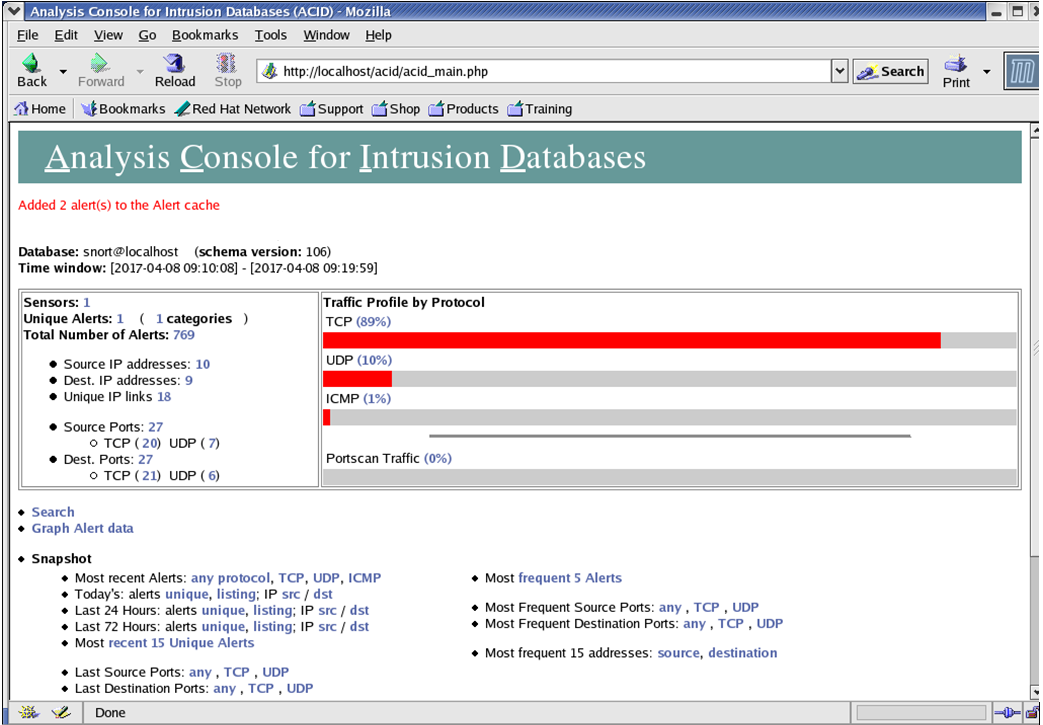
Analysis Console for Intrusion Database (ACID) is a tool used to analyze and present Snort data using a web interface. It is written in PHP. It works with Snort and databases like MySQL. A user will use a web browser to interact with ACID.

The following test is to show you how user customized rules are used by Snort and how we use ACID to analysis Snort alerts from MySQL database. The first test is as follow:

First you need to delete all the build-in rules used by Snort in the snort.conf configuration file, and then add the following rule to the snort.conf file rules part:

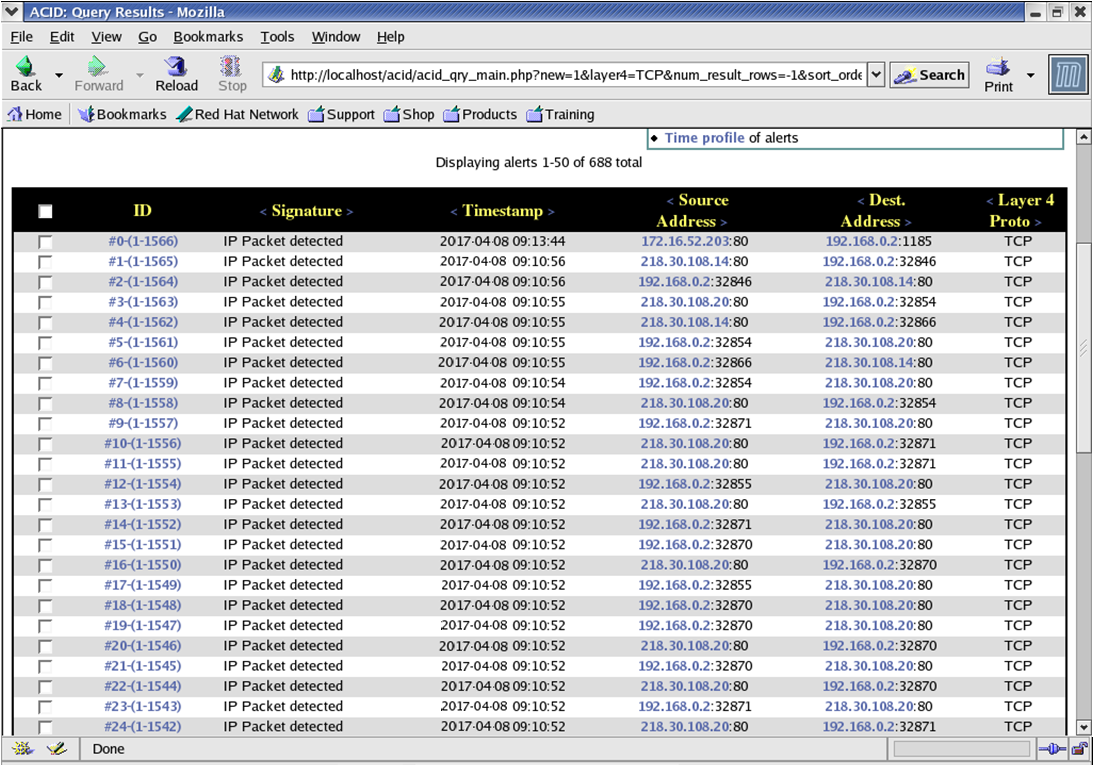
alert ip any any -> any any (msg: “IP Packet detected”;)

This rule will generate an alert message for every captured IP packet. The ACID main page provides an overview of currently available data. It has different sections to display information in groups. You can view traffic profiles by different protocols, get a snapshot of sensors, search data and so on.



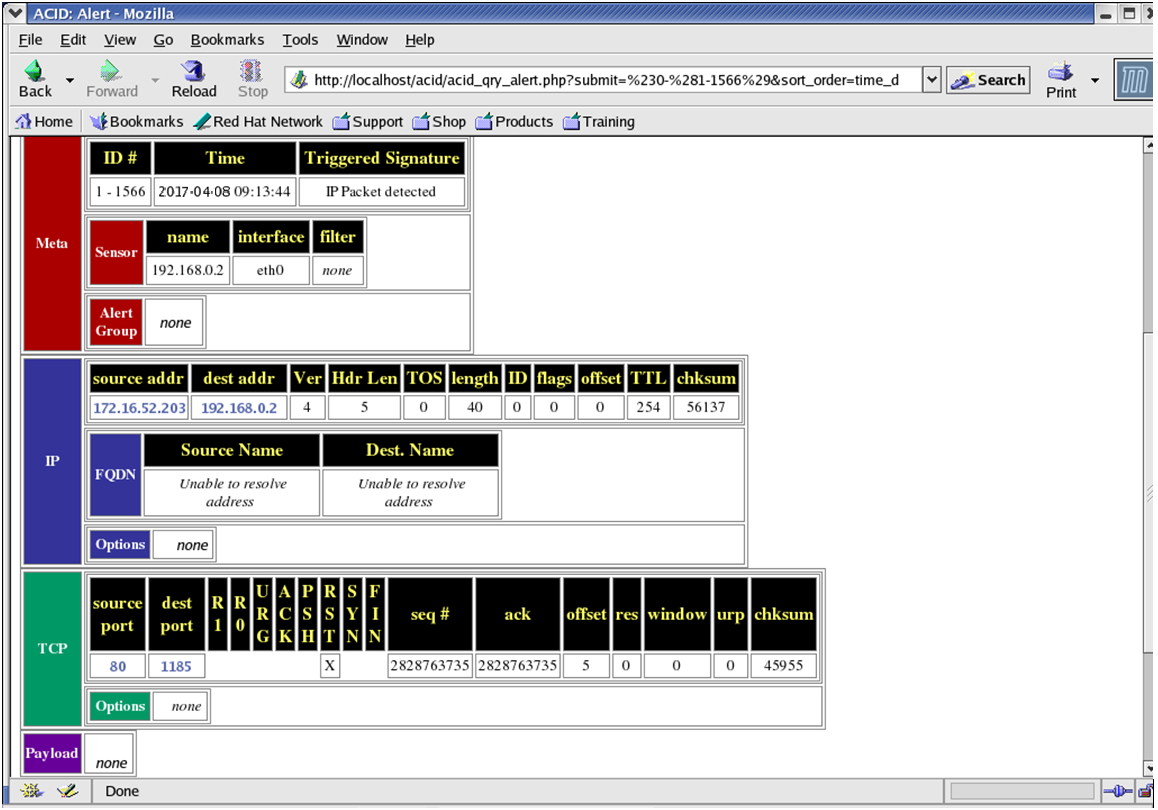
# Figure11: ACID main page when using user customized rule

From the main page, you can click on a protocol to get information about packets logged fro that particular protocol. Figure12 shows a screen shot for TCP protocol. The bottom part of the screen shows the last fifteen individual packets that have been logged into the database.



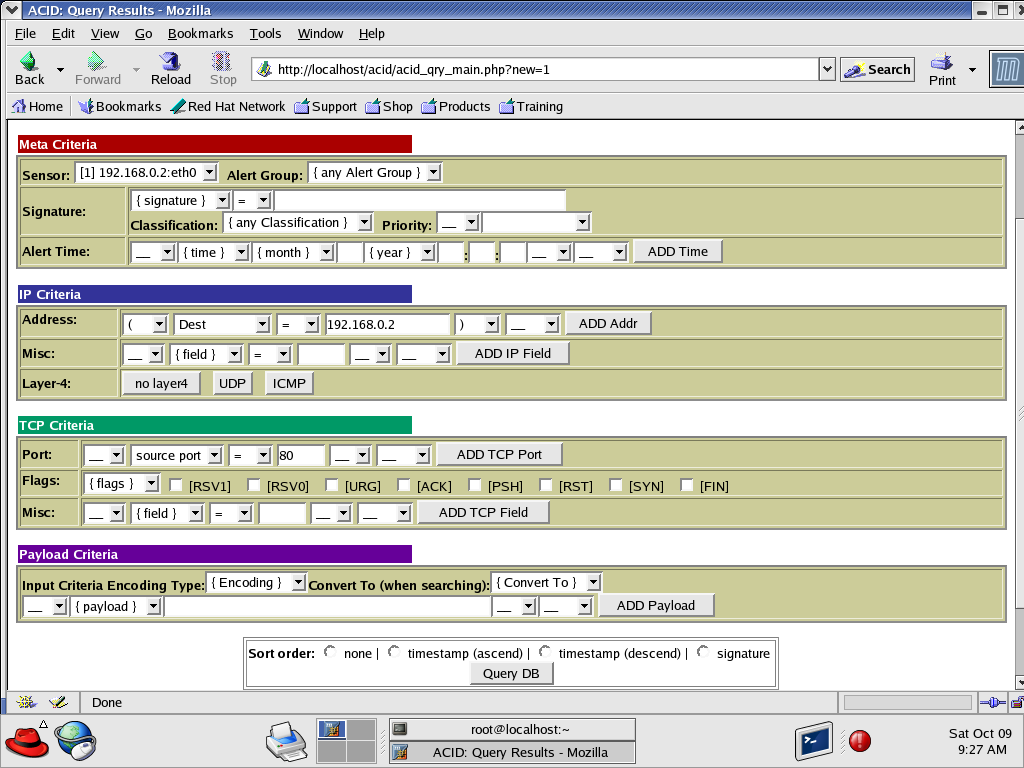
**Figure12: TCP protocol data**

Figrue13 shows details about a particular TCP packet.

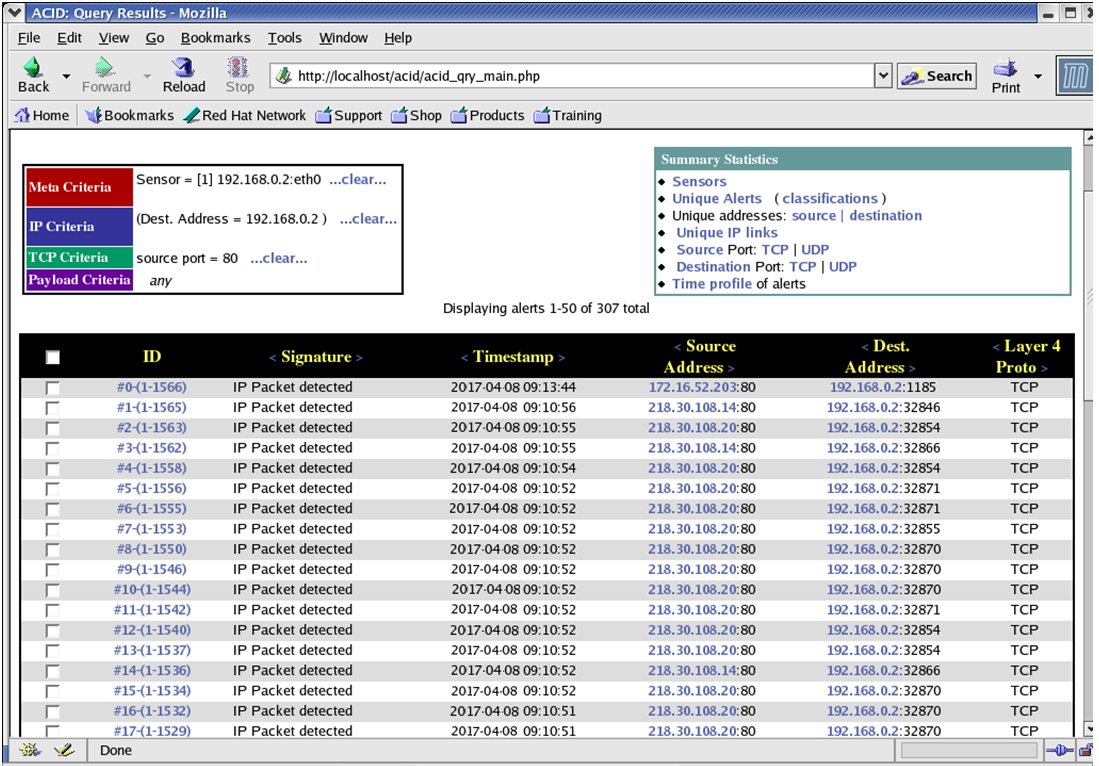


# Figure13: TCP alert detail

One important feature of ACID is that it can be used to search the captured alert data. Figure14 is the screen shot for some searching procedure. All the criteria that you specify in this screen are translated to a SQL statement that is passed to the MySQL database server. Figure15 is the result of this query.

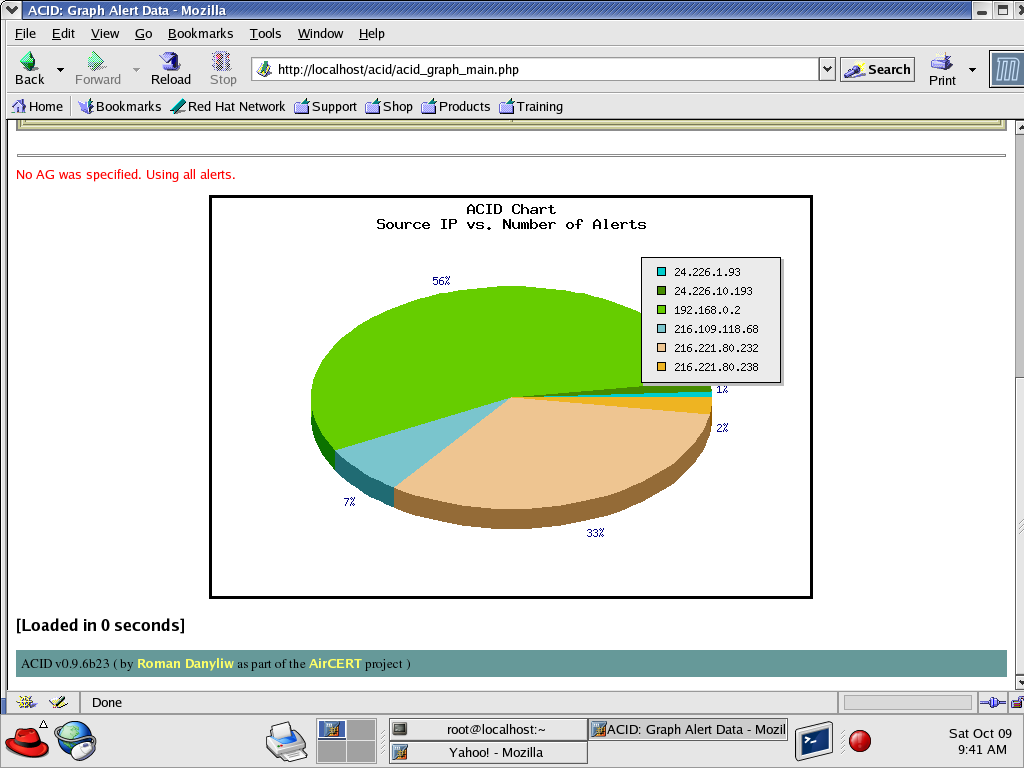


# Figure14: Searching database using ACID

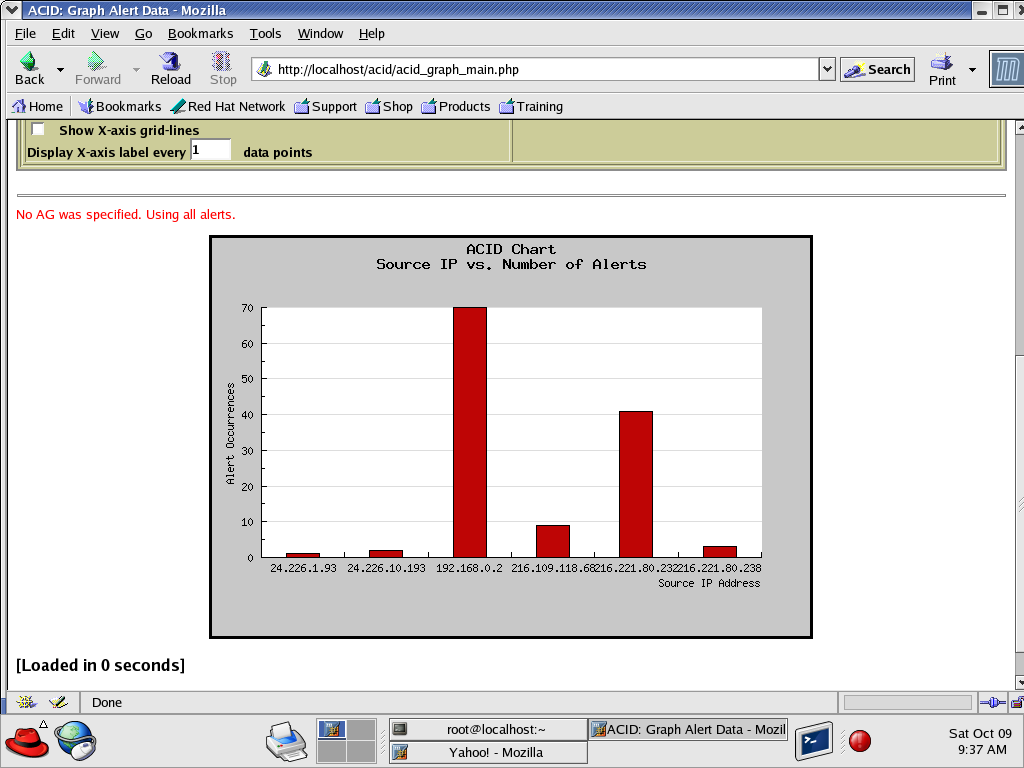


# Figure15: Searching result

Finally, you can go to the ACID main page where a link is provided to generate graphs. When generating graph, you can select data and type of graph. Figure13 and 14 are the line graph and bar graph generated for the IP…



# Figure16: ACID Pie graph



# Figure17: ACID Bar graph

#### **Conclusion**

Intrusion Detection System look for attack signatures, which are specific patterns that usually include malicious or suspicious intent. Network-based Intrusion Detection Systems (IDS) use raw network packets as the data source. Snort is a light weight IDS, which uses a bunch of build-in rules and user customized rules to prevent prospective intruder from intruding in a network system. This report discussed installation procedure for Snort as well as other products that work with Snort, components of Snort, and most frequently used functions. Finally, we gave a comprehensive test in RedHat9 Linux system about how to use ACID to analyze alerts generated by Snort based on some specific rules.

#### **References**

1. Snort User Manual: <http://www.snort.org/docs/snort_manual/>
2. Snort, Apache, PHP, MySQL, ACID on Redhat 9.0 Installation Guide: <http://www.snort.org/docs/snort_acid_rh9.pdf>
3. Network Intrusion Detection Using Snort: <http://www.linuxsecurity.com/feature_stories/feature_story-49.html>
4. Intrusion Detection with SNORT: Advanced IDS Techniques Using SNORT, Apache, MySQL, PHP, and ACID: <http://www.informit.com/title/0131407333#>