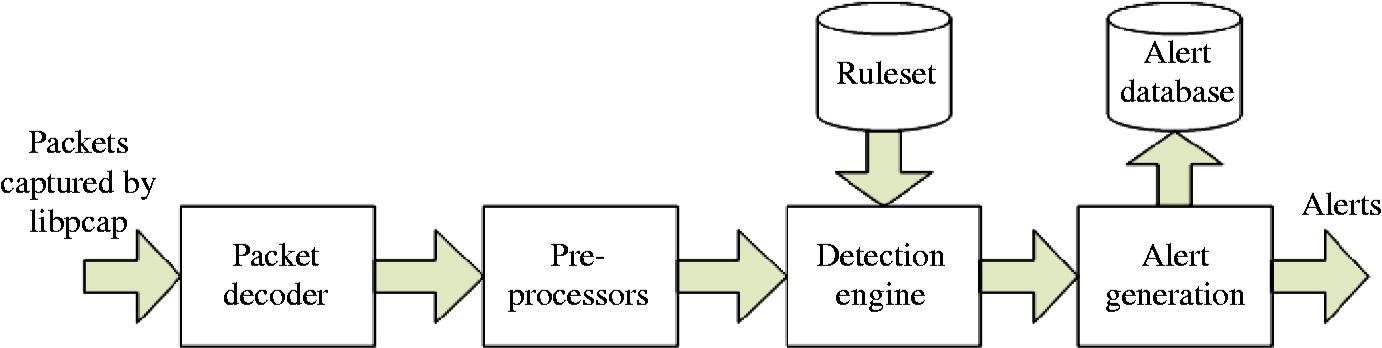
|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | **Hope Foundation’s,**  **Finolex Academy of Management and Technology, Ratnagiri** | | | | | | | | | |
| **Department of Information Technology** | | | | | | | | | |
| Subject name: SECURITY LAB | | | | | | | | Subject Code: ITL502 | | | |
| Class | | TE IT | | Semester – V (CBCGS) | | | | Academic year: 2018-19 | | | |
| Name of Student | | **Kazi Jawwad A Rahim** | | | | | **Quiz Score: 06** | | | | |
| Roll No | | **32** | | | Experiment No. | | | | | 10 | |
| Title**. Case Study:**  **Set up Snort and study the logs.** | | | | | | | | | | | |
|  | | | | | | | | | | | |
| **Course objectives applicable:**  To be able to set up firewalls and intrusion detection systems using open source technologies and to explore email security. | | | | | | | | | | | |
| **Course outcomes applicable:**  Apply and set up firewalls and intrusion detection systems using open source technologies and to explore email security. | | | | | | | | | | | |
| **Learning Objectives:**  The learner will be able to:-   * Simulate intrusion detection system using snort tool. * To use current techniques, skills, and IDS tools necessary for computing practice. | | | | | | | | | | | |
| **Hardware Requirements**:   1. PC with 4GB RAM, 500GB HDD.   **Software Requirements:**  1. Snort tool in linux | | | | | | | | | | | |
|  | | | | | | | | | | | |
|  | | | | | | | | | | | |
| **Experiment/Assignment Evaluation:** | | | | | | | | | | | |
| **Sr. No.** | **Parameters** | | | | | | | | **Marks obtained** | | **Out of** |
| **1** | Technical Understanding | | | | | | | | NA | | 0 |
| **2** | Neatness/presentation | | | | | | | |  | | 5 |
| **3** | Punctuality | | | | | | | |  | | 5 |
| **Date of performance (DOP)** | | |  | | | **Total marks obtained** | | |  | | **10** |
| **Date of checking (DOC)** | | |  | | | **Signature of teacher** | | | | | |

**Introduction**

Cloud computing has evolved into a new way of computing model providing resources and services on the internet. Three types of services provided over cloud are Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS) (Mell & Grance, 2011). Any attack on the Cloud vulnerabilities affects CIA (confidentiality, integrity and availability) of cloud resources. According to a survey conducted by IDC, security of cloud services is the major challenge (Gens, 2008). Attacks may comprise of IP spoofing, DDoS, port scanning etc. Major cloud providers use firewall to prevent the outside attacks and is considered first line of defense. But firewall cannot detect insider attack from within the network and more complex forms of attack go un-noticed through firewalls. This calls for a more robust security system to be implemented and NIDS (Network intrusion detection system) performs exactly what is needed. In this case study, we use Snort (an open source NIDS) and explore the possibilities of securing the cloud system and propose some recommendations to secure cloud services.Snort (a product of Cisco) is a signature based intrusion detection system which allows monitoring of network traffic. It analyzes network traffic for any type of intrusion and generates alerts. It is an open source NIDS available under GPL, and runs on Windows, Mac OS, Linux based operating systems.

**Snort Architecture**

As described in Snort manual (available on <https://www.snort.org/>), Snort comprises of mainly five components working together to monitor and analyze all network traffic and look for any signs of intrusions and generate alert. As shown in figure 1 below, the five major components are as follows:

*Packet Decoder*: Captures packets from the network traffic and sets them up for preprocessor orfor the detection engine.

*Preprocessors*: Processes the captured packets against certain plugins. These plugins check forknown type behavior or anomalies. Preprocessors are indispensable part of any IDS to prepare data packets to be checked by detection engine against rules in the detection engine as intruders may modify those packets to escape any detection.

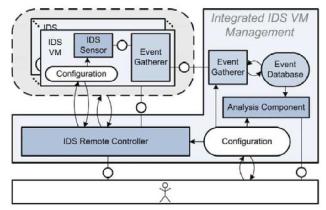
*Detection Engine*: Once the data is handled by the preprocessors, it is then passed on to detectionengine. The detection engine is the most critical component of the signature based IDS in Snort. It matches the data packets with the set of rules for any intrusion signatures contained in the data packets. If the rules match the data packets then it is passed on to the alert processor. It may take different amount of time for responding to several types of packets irrespective of the computing system it is running on.

*Logging and Alerting System*: Generation of alerts and logging is handled by this system. All thealerts and logs are kept in simple plain text files or tcp-dump style files.

*Output Module*: Output module helps save the logs generated by logging and alerting system indiverse ways like in simple plain text log files, logging to database like MySQL or Oracle or generating XML depending on the configuration set into Snort configuration file.

**IDS Architecture for Cloud**

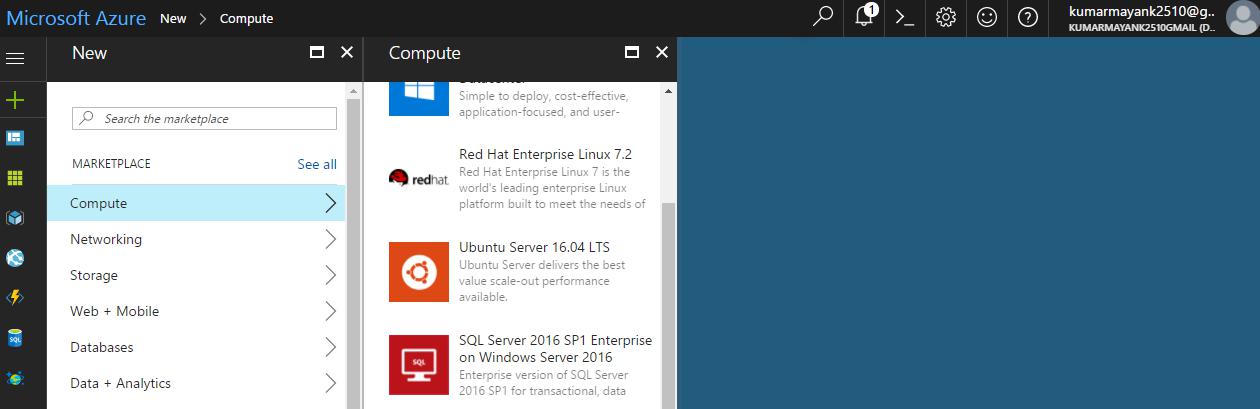
In Figure 2 (S. Roschke & Meinel 2009) proposed a virtual machine integrated IDS architecture. It consists mainly of two components, IDS management unit and IDS sensor. Event gatherer built in IDS management unit collects anomalous behavior detected by IDS sensor and stores into event database. Analysis component then analyzes these logged events in the database according to the configuration. IDS Remote Controller can communicate with IDS-VMs and IDS sensors and hence manages the IDS-VMs. This approach of IDS deployment helps securing the virtual machines on cloud and the services running on it.



**Implementation**

In this experimental setup, we have chosen Microsoft Azure as our cloud platform and created a Ubuntu Server 16.0.4 LTS virtual machine using 30 days free trial which give a 300 NZD worth credit and can be utilized to choose from various plans and features to incorporate into your virtual machine depending upon the computing resources required. The setup takes 10-15 minutes and has a well-defined user interface and guidance system which helps create the virtual machine seamlessly easy.

|  |  |
| --- | --- |
| SNORT IN THE CLOUD | 5 |



Once the virtual machine is up and running, we start the installation of Snort. But before we can install Snort, we need to install some pre-requisites.

Snort has 4 essential pre-requisites:

pcap (libpcap-dev) available from the Ubuntu repository

PCRE (libpcre3-dev) available from the Ubuntu repository

Libdnet (libdumbnet-dev) available from the Ubuntu repository

DAQ (http://www.snort.org/downloads/) compiled from source

Firstly, we need to install all tools required for building software.

sudo apt-get install -y build-essential

Once build-essential is installed, we now install all the prerequisite packages which are available

at Ubuntu repositories.

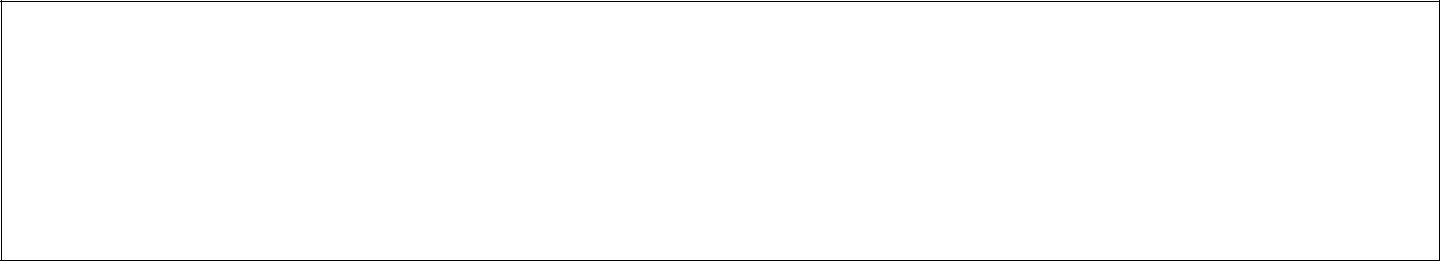
sudo apt-get install -y libpcap-dev libpcre3-dev libdumbnet-dev

The Snort DAQ (Data AcQuisition library) needs few pre-requisites that need to be installed:

sudo apt-get install -y bison flex

Install the latest version of DAQ using following set of commands. The PCAP DAQ module is the default module, used for getting packets into Snort from a file or an interface.

|  |  |
| --- | --- |
|  |  |



wget https://snort.org/downloads/snort/daq-2.0.6.tar.gz

tar -xvzf daq-2.0.6.tar.gz

cd daq-2.0.6

./configure

make

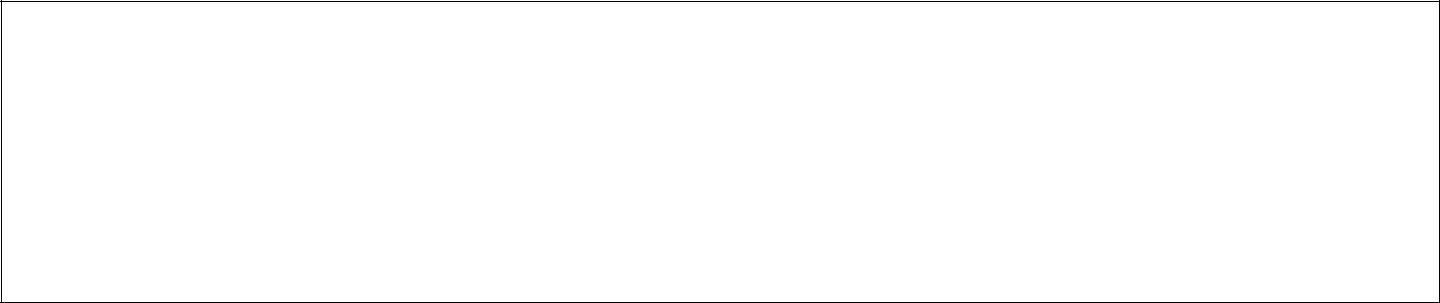
sudo make install

*Installing Snort:*

Four optional libraries that improves functionality are: liblzma-dev three of which provide decompression of swf files (adobe flash), openssl, and libssl-dev which both provide SHA and MD5 file signatures:

sudo apt-get install -y zlib1g-dev liblzma-dev openssl libssl-dev

After all pre-requisites are installed, now we are ready to download the Snort source tarball, compile, and install. The --enable-Sourcefire option gives Packet Performance Monitoring (PPM), which lets us do performance monitoring for rules and pre-processors, and builds Snort the same way that the Snort team does:



mkdir ~/snort\_src

cd ~/snort\_src

wget https://snort.org/downloads/snort/snort-2.9.9.0.tar.gz tar -xvzf snort-2.9.9.0.tar.gz cd snort-2.9.9.0 ./configure --enable-sourcefire

make

sudo make install

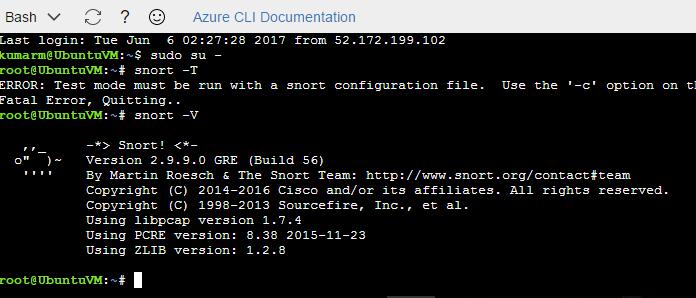
Update the shared libraries by running the following command (skipping this step will result in error while running Snort):

sudo ldconfig

Place a symlink to the Snort binary in /usr/sbin:

sudo ln -s /usr/local/bin/snort /usr/sbin/snort

Test Snort by running the binary as a regular user, passing it the -V flag (which tells Snort to verify itself and any configuration files passed to it).

snort -V

*Configuring Snort:*

Create few directories and files as shown below for Snort and setup their permission.



# Create the Snort directories:

sudo mkdir /etc/snort

sudo mkdir /etc/snort/rules

sudo mkdir /etc/snort/rules/iplists

sudo mkdir /etc/snort/preproc\_rules

sudo mkdir /usr/local/lib/snort\_dynamicrules

sudo mkdir /etc/snort/so\_rules

# Create files that store rules and ip lists

sudo touch /etc/snort/rules/iplists/black\_list.rules

sudo touch /etc/snort/rules/iplists/white\_list.rules

sudo touch /etc/snort/rules/local.rules

sudo touch /etc/snort/sid-msg.map

# Create our logging directories:

sudo mkdir /var/log/snort

sudo mkdir /var/log/snort/archived\_logs

# Adjust permissions:

sudo chmod -R 5775 /etc/snort

sudo chmod -R 5775 /var/log/snort

sudo chmod -R 5775 /var/log/snort/archived\_logs

sudo chmod -R 5775 /etc/snort/so\_rules

sudo chmod -R 5775 /usr/local/lib/snort\_dynamicrules

To copy the configuration files and the dynamic preprocessors, run the following commands:

cd ~/snort\_src/snort-2.9.9.0/etc/

sudo cp \*.conf\* /etc/snort

sudo cp \*.map /etc/snort

sudo cp \*.dtd /etc/snort

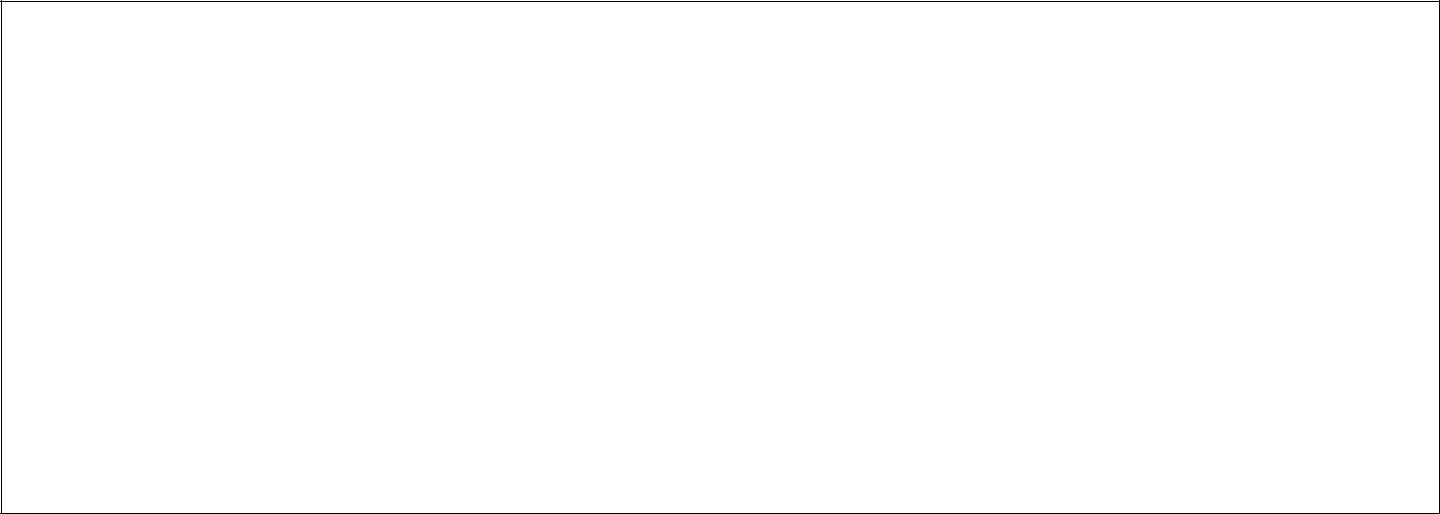
cd ~/snort\_src/snort-2.9.9.0/src/dynamicpreprocessors/build/usr/local/lib/snort\_dynamicpreprocessor/ sudo cp \* /usr/local/lib/snort\_dynamicpreprocessor/

Edit Snort’s main configuration file, snort.conf. To run Snort in NIDS mode, this file needs to be passed as an argument.

Next, open /etc/snort/snort.conf file in any editor:

nano /etc/snort/snort.conf

Provide the machine’s ip address and define paths as shown below:



* Setup the network addresses you are protecting ipvar HOME\_NET 10.0.0.4/24
* Set up the external network addresses. Leave as "any" in most situations ipvar EXTERNAL\_NET any

var RULE\_PATH /etc/snort/rules

var SO\_RULE\_PATH /etc/snort/so\_rules

var PREPROC\_RULE\_PATH /etc/snort/preproc\_rules

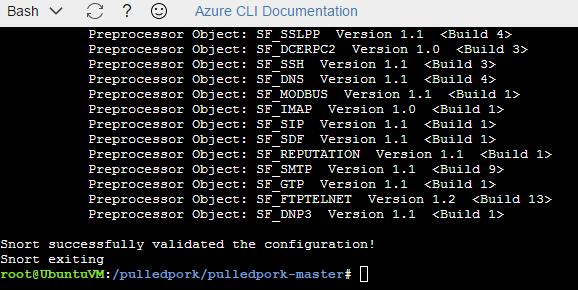
var WHITE\_LIST\_PATH /etc/snort/rules

var BLACK\_LIST\_PATH /etc/snort/rules

include $RULE\_PATH/local.rules

Save and close the file. Next, using below command validate the configuration file:

|  |  |
| --- | --- |
| Sudo /usr/local/bin/snort -A console -q -u snort -g snort -c /etc/snort/snort.conf -i eth0 |  |



Testing Snort

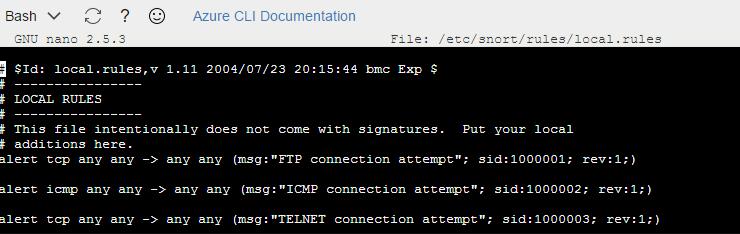
Now Snort can be tested but before that few rules need to be created to test local data packets on the network.

Lastly, create some rules to test Snort.

First, edit the local.rules file:

nano /etc/snort/rules/local.rules

Add the following lines:

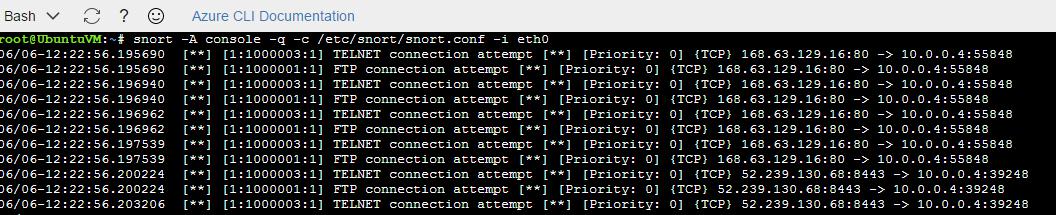


Save and close the file.

The above rules will generate alerts when someone tries to Ping, FTP or Telnet to the server.

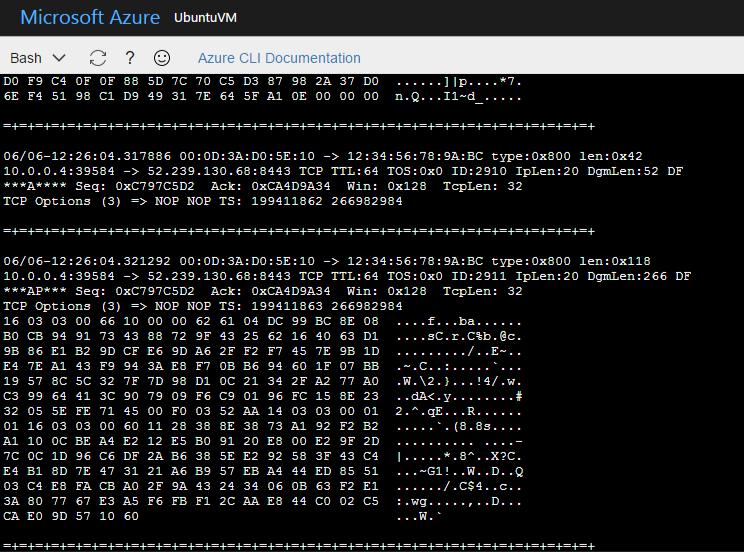
Now, run Snort in NIDS mode and send alert output to the console:

snort -A console -q -c /etc/snort/snort.conf



*Snort in sniffer mode:*

Snort -vde



**Discussion**

Snort is a widely used open source NIDS, supporting various operating system environments. With many plugins supported by Snort, it’s functionality and efficiency can be modified or enhanced to the deployment environment. It can act as a host based intrusion detection system when installed on virtual machines in Cloud or as a network intrusion detection system when installed on network interface cards or routers. The major advantage Snort holds is its ability to be configured as per the needs of any organization and creation of own rule sets which can be used by advanced users to implement soft computing techniques and generate their own rule sets to be used with snort. It is very scalable, has a low computational cost, low false positives and low false negatives. During the implementation, Snort could detect traffic through the network interface going through the virtual machine on Azure. It was also able to distinctly identify several types of data packets such as TCP and TELNET on the network and generated custom message alerts as defined in the local rules.

Snort can be used by a variety of organizations and businesses. However, an area of oversight is the security of library information systems and using Snort in this context. A library information system may also include an EZproxy component for authenticating and providing online library resources to off-campus users (Erturk & Iles, 2015). Snort, for example, can be used to analyze EXproxy packets when there are unusual issues (Feinstein, 2006). This is a technical area that requires more documentation, and can be studied in the future with the most recent versions of Snort.

**Conclusion**

Although installing Snort on Virtual Machine in cloud provides protection to the virtual machine from malicious activity, it cannot detect intrusion coming from outside of a network. As a result, a more distributed form of IDS placement is required on the Cloud network (Modi et al., 2012).The placement of IDS on a central server can help in detecting both internal as well as external intrusions. Also, placing the IDS on each VM instance provides more robust security to the cloud system. Snort can be used along with any other relational database such as MySQL or Oracle to log the alerts for analysis by any other analyzer plugin or Barnyard2 can be also used which is a unified2 binary output file reader to read the log files generated by Snort. Snort provides constant updates to the rules sets for the new threat signatures which can be downloaded manually and updated, or Pulled Pork plugin can be used to maintain and update the rule sets. As it is evident from the activity conducted that Snort is light weight easy to use, highly configurable (with many plugins supports to enhance its functionality) IDS. As the popularity of Cloud services grows, so does the related risk of security; hence IDS hold an important key in solving this issue.

**References** :

1. http://www.thegeekstuff.com/2010/08/snort-tutorial/
2. https://www.howtoforge.com/intrusion\_detection\_base\_snort
3. https://www.snort.org/