Experiment Number: C2

TITLE:
Install and Use Latest IDS (Open Source)

OBJECTIVES:

1. To develop problem solving abilities using Mathematical Modeling.
2. To apply algorithmic strategies while solving problems.

THEORY:

An intrusion detection system (IDS) inspects all inbound and outbound network activity and identifies suspicious patterns that may indicate a network or system attack from someone attempting to break into or compromise a system.

There are several ways to categorize IDS:

- **Misuse detection** vs. **anomaly detection**: in misuse detection, the IDS analyzes the information it gathers and compares it to large databases of attack signatures. Essentially, the IDS looks for a specific attack that has already been documented. Like a virus detection system, misuse detection software is only as good as the database of attack signatures that it uses to compare packets against. In anomaly detection, the system administrator defines the baseline, or normal, state of the networks traffic load, breakdown, protocol, and typical packet size. The anomaly detector monitors network segments to compare their state to the normal baseline and look for anomalies.
- **Network-based** vs. **host-based systems**: in a network-based system, or NIDS, the individual packets flowing through a network are analyzed. The NIDS can detect malicious packets that are designed to be overlooked by a firewall simplistic filtering rules. In a host-based system, the IDS examines at the activity on each individual computer or host.
- **passive system** vs. **reactive system**: in a passive system, the IDS detects a potential security breach, logs the information and signals an alert. In a reactive system, the IDS responds to the suspicious activity by logging off a user or by reprogramming the firewall to block network traffic from the suspected malicious source.
- IPS and IDS:

  Ans:

Sr no.	Parameters	Intrusion Prevention System	Intrusion Detection System
1.	Placement in network infrastructure	Part of the direct line of communication(inline)	Outside direct line of communication(Out of band)
2.	System Type	Active and/or passive	Passive
3.	Detection Mechanisms	1.Stastical anomaly based detection	Signature detection

		2. Signature detection	
4.	Usefulness	Ideal for blocking web destruction	Ideal for identifying blocking attacks
5.	Traffic Requirement	"Original" traffic is required	Traffic replication is required

Types of IDS:

1. Active and passive IDS:

An **active Intrusion Detection Systems (IDS)** is also known as Intrusion Detection and Prevention System (IDPS). Intrusion Detection and Prevention System (IDPS) is configured to automatically block suspected attacks without any intervention required by an operator.

A **passive IDS** is a system that's configured to only monitor and analyze network traffic activity and alert an operator to potential vulnerabilities and attacks. A passive IDS is not capable of performing any protective or corrective functions on its own.

2. Network Intrusion detection systems (NIDS) and Host Intrusion detection systems (HIDS)

Network Intrusion Detection Systems (NIDS) usually consists of a network appliance (or sensor) with a Network Interface Card (NIC) operating in promiscuous mode and a separate management interface. The IDS is placed along a network segment or boundary and monitors all traffic on that segment.

A **Host Intrusion Detection Systems (HIDS)** and software applications (agents) installed on workstations which are to be monitored. The agents monitor the operating system and write data to log files and/or trigger alarms.

3. Knowledge-based (Signature-based) IDS and behavior-based (Anomaly-based) IDS

A **knowledge-based (Signature-based) Intrusion Detection Systems (IDS)** references a database of previous attack signatures and known system vulnerabilities. The meaning of word signature, when we talk about Intrusion Detection Systems (IDS) is recorded evidence of an intrusion or attack. Each intrusion leaves a footprint behind (e.g., nature of data packets, failed attempt to run an application, failed logins, file and folder access etc.). These footprints are called signatures.

A Behavior-based (Anomaly-based) Intrusion Detection Systems (IDS) references a baseline or learned pattern of normal system activity to identify active intrusion attempts. Deviations from this baseline or pattern cause an alarm to be triggered.

Advantages:

1. It can detect the unauthorized user
2. It can detect password cracking and denial of services
3. It can catch illegal data manipulations
4. Monitors the operations of firewalls
5. Allows administrator to tune, organize operating system audit trails other logs

Disadvantages:

1. Host-based IDS works well for a single machine, extremely labor-intensive of monitor multiple machines.
2. If host is compromised then no more alerts will be generated
3. There is no detection with unknown signatures

Example:

Examples of Network IDS:

- SNORT

Examples of HIDS:

- OSSEC - Open Source Host-based Intrusion Detection System
- Tripwire
- AIDE - Advanced Intrusion Detection Environment
- Prelude Hybrid IDS

Snort is a free and open source network intrusion prevention system (NIPS) and network intrusion detection system (NIDS) created by Martin Roesch in 1998. In 2009, Snort entered InfoWorld's Open Source Hall of Fame as one of the "greatest [pieces of] open source software of all time".

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 matching. These basic services have many purposes including application-aware triggered quality of service, to de-prioritize bulk traffic when latency-sensitive applications are in use.

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.

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 analyze it against a rule set

defined by the user. The program will then perform a specific action based on what has been identified.

Installation steps for snort:

First we need to install all the prerequisites from the Ubuntu repositories:

```
sudo apt-get install -y build-essential libpcap-dev libpcre3-dev libdumbnet-dev
bison flex zlib1g-dev liblzma-dev openssl libssl-dev
```

Breakdown of the packages you are installing:

build-essential: provides the build tools (GCC and the like) to compile software.

bison, flex: parsers required by DAQ (DAQ is installed later below).

libpcap-dev: Library for network traffic capture required by Snort.

libpcre3-dev: Library of functions to support regular expressions required by Snort.

libdumbnet-dev: the libdnet library provides a simplified, portable interface to several low-level networking routines. Many guides for installing Snort install this library from source, although that is not necessary.

zlib1g-dev: A compression library required by Snort.

liblzma-dev: Provides decompression of swf files (adobe flash)

openssl and libssl-dev: Provides SHA and MD5 file signatures

Next install ethtool:

```
sudo apt-get install -y ethtool
```

now edit /etc/network/interfaces as an admin and append the following two lines for each network interface we will have Snort listen on

(in our case eth0):

```
post-up ethtool -K eth0 gro off
```

```
post-up ethtool -K eth0 lro off
```

Important note for people running Ubuntu 15.10: In Ubuntu 15.10, for new installations (not upgrades), network interfaces no longer follow the ethX standard (eth0, eth1, …). Instead, interfaces names are assigned as Predictable Network Interface Names. This means you need to check the names of your interfaces using ifconfig -a and replace eth0 with whatever network interface you find.

Next we will create a directory to save the downloaded tarball files:

```
mkdir snort_src
```

Snort uses the Data Acquisition library (DAQ) to abstract calls to packet capture libraries. DAQ is downloaded and installed from the Snort website:

```
cd snort_src
wget https://www.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
```

Now we are ready to install Snort from source. When we configure the build of Snort, we use the –enable-sourcefire flag, which enables Packet Performance Monitoring (PPM), and matches the way the sourcefire team builds Snort.

```
cd snort_src
wget https://www.snort.org/downloads/snort/snort-2.9.8.0.tar.gz
tar -xvzf snort-2.9.8.0.tar.gz
cd snort-2.9.8.0
./configure --enable-sourcefire
make
```

```
sudo make install
```

Run the following command to update shared libraries:

```
sudo ldconfig
```

Since the Snort installation places the Snort binary at /usr/local/bin/snort, it is a good policy to create a symlink to /usr/sbin/snort:

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

The last step of our Snort installation is to test that the Snort Binary runs. Execute Snort with the -V flag, which causes Snort to show the version number:

```
/usr/sbin/snort -V
```

you should see output similar to the following:

```
user@snortserver:~$ /usr/sbin/snort -V

   ,,_     -*&amp;gt; Snort! &amp;lt;*-
  o&amp;quot;  )~   Version 2.9.8.0 GRE (Build 229)
   ''''    By Martin Roesch &amp;amp; The Snort Team:
http://www.snort.org/contact#team
           Copyright (C) 2014 Cisco and/or its affiliates. All rights reserved.
           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

user@snortserver:~$
```

For all the following instructions become a superuser and then execute them.

Create the following directories and files:

```
mkdir /etc/snort

mkdir /etc/snort/rules

mkdir /var/log/snort

mkdir /etc/snort/preproc_rules

touch /etc/snort/rules/white_list.rules

touch /etc/snort/rules/black_list.rules

touch /etc/snort/rules/local.rules
```

Change permissions:

```
chmod -R 5775 /etc/snort/

chmod -R 5775 /var/log/snort/

chmod -R 5775 /usr/local/lib/snort
```

Copy *.conf and *.map files from snort download directory to /etc/snort:

```
cp /home/snort_src/snort-2.9.8.0/etc/*.conf* /etc/snort/

cp -v /home/snort_src/snort-2.9.8.0/etc/*.map* /etc/snort/
```

Before editing snort.conf get the backup of that file first:

```
cp /etc/snort/snort.conf /etc/snort/snort.conf_orig
```

Give following Command:

```
sed -i 's/include \$RULE\_PATH/#include \$RULE\_PATH/' /etc/snort/snort.conf
```

Note:Above Command will comment all rulesets which we will edit line by line

Go to line 45 of /etc/snort/snort.conf, edit to make like below

```
ipvar HOME_NET 192.168.1.0/24
ipvar EXTERNAL_NET !$HOME_NET
```

(Note: replace above ip address with your ip address)

MATHEMATICAL MODEL:

Let P be the solution perspective.

P ={ S , E , I , O, F }

S = { Initial state of the IDS, Initial snort configuration file }

I = Input of the system → { I1 , I2 }
where,
I1 = { List of local rules }
I2 = { List of blocked IP's }

O = Output of the system → { O1 , O2, O3, O4 }
where,
O1 = { list_blocked_ip }
O2 = { display_firewall_rules}
O3 = {Display network traffic}
O4 = {Allow_IDS_ON/OFF}

F = Functions used → { f1 , f2 , f3 }
where
f1 = { IDS on / off }
f2 = { Blocked_ip's }
f3 = { Display network traffic }
f4 = {Display_rules}
E = End state of the system shows successfully installing snort and handling and updating the IDS and firewall rules of the system.

Input :-
List of blocked IP's and list of firewall rules

Expected Output :-

1. Allow IDS on/off
2. Display Blocked IP list
3. Display Firewall rules
4. Display network traffic

Execute the program with the following commands.
Note: Make sure you should be sudo user.
>sudo su
>Now start snort as a daemon process with the help of -D option and set the file path where the logs will be generated using following command :
snort -D -c /etc/snort/snort.conf -l /var/log/snort/
>To view the current network traffic along with packet headers and the data do:
snort -vd
>To view the list of blocked IP addresses do:
cat /etc/snort/rules/iplists/black_list.rules
>To view the list of firewall rules do:
cat /etc/snort/rules/local.rules
>To kill the snort daemon do:
pkill snort

TEST CASES:

Test ID	Description	Input	Expected output	Actual output
1.	List all blocked IPs	List of IP's to be blocked	Should list all blocked IPs	Listing all blocked IPs
2.	Start IDS	IDS install	Should start the IDS	IDS got started
3.	Start IDS	IDS not install	Should provide the exception and not able to start IDS	IDS not started
4.	List Firewall rules	Local rules	Should display Firewall rules	Displaying firewall rules

CONCLUSION:

Hence based on the rule of firewall and IDS, we have learnt about the blocking and unblocking the IPs from the IP tables and configuring the IDS.