International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395 -0056
Volume: 04 Issue: 04 | Apr -2017 www.irjet.net p-ISSN: 2395-0072

Intrusion Detection System Using SNORT

Rishabh Gupta[1], Soumya Singh[2], Shubham Verma[3], Swasti Singhal[4]

[1]UG Scholar, Dept. of IT, GCET, Greater Noida, UP, India
[2]UG Scholar, Dept. of IT, GCET, Greater Noida, UP, India
[3]UG Scholar, Dept. of IT, GCET, Greater Noida, UP, India
[4]Assistant Professor, Dept. of IT, GCET, Greater Noida, UP, India
Galgotias College of Engineering and Technology

Abstract - *Nowadays corporate company's network can generate false alarms and are a major target of exploits. They have lots of sensitive data which can be misused to leak information which are critical to the company and its employees. In order to avoid these kinds of attack, companies use Intrusion Detection System. Intrusion Detection System (IDS) inspects every packet passing through the network and raise alarm if these is any attempt to perform malicious activity. IDS ensure a security policy in every single packet passing through the network. Snort is an open- source , lightweight tool which captures every detail of packet passing through the network and generate alerts if any one packets matches the signatures inserted given by the company. The signatures are basically the rules written so that IDS can know on which packets it should generate the alert. In this paper we have implemented Intrusion Detection System using Snort in order to detect signature based network attacks.*

Key Words: Intrusion Detection System, Snort, Signature-based, barnyard, Anomaly-based

1. INTRODUCTION

Network security is one of the biggest challenges that companies are facing from time to time. There are lots of attempts by the black hat hackers to break and compromise with the security of Company's network and some of them are even successful. As the use of internet increasing, these malicious activities are gaining popularity among the black hats.

Everyday large amount of data is being generated and passed on and lots of these data holds sensitive information about the company and its employees. Thus securing network is one of the most important task for a company to survive .To make this easier and efficient we use Intrusion Detection System , it helps to collect information about any malicious packet that passes across a company's network.

2. Intrusion Detection System

Intrusion detection system (ID) is a type of security system for computers and computer networks. Intrusion Detection basically helps in detecting outer and inner attacks performed by either user or hackers.

An ID system collects information from various sources and analyzes information from various areas within a computer or a network to identify possible security breaches, which include both intrusions (attacks from outside the organization) and misuse (attacks from within the organization). ID uses vulnerability assessment (sometimes referred to as *scanning*), which is a technology developed to assess the security of a computer system or network

Advantages of IDS

- Track any changes in the behavior of network.
- Inspects system activity
- Can differentiate between normal and abnormal activities in the network
- Automated

Disadvantages of IDS

- Sometimes gives false alarms i.e. the packet wasn't malicious but IDS might still generate an alert.
- Time consuming
- Is not 100% safe from attacks

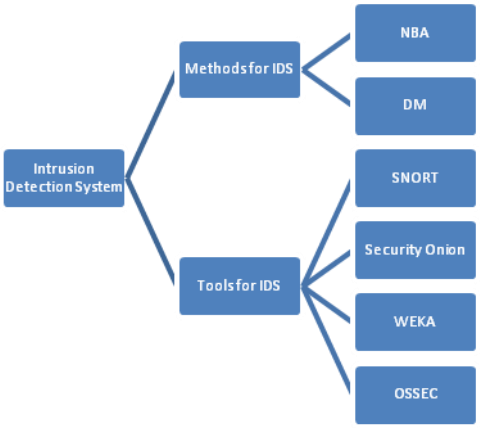


Fig -1: Flowchart for Intrusion Detection

2.1 Methods of IDS

There are mainly 2 methods in IDS

2.1.1 NBA (Network Behavior Analysis)

Network Behavior Analysis (NBA) has been one of these emerging technologies that have been sold as a security management tool to improve the current network security status. The main focus of NBA is to monitor inbound and outbound traffic associated with the network to ensure that nothing is getting into the servers, software, and application systems which helps enhance the overall security of the network at all levels. Approximately 25% of large enterprises systems start using NBA since 2011.

2.1.2 DM (Detection method)

This is the method which will detect the intrusion in any specific network. There might be the case of false alarm but probability of false alarm is very less. This method is only capable of detecting the intrusions it cannot remove intrusion from the system.

2.2 Tools used in IDS

There are various tools to implement Intrusion detection system. Some of the most widely used tools are

- SNORT
- Security Onion
- WEKA
- OSSEC

Here in our project we are using SNORT for IDS implementation

2.2.1 SNORT

Snort is a light-weight intrusion detection tool which logs the packets coming through the network and analyzes the packets. Snort checks the packets coming against the rules written by the user and generate alerts if there are any matches found. The rules are written by the user in a text file which is linked with snort.conf file where all the snort configurations are mentioned. There are few commands which is used to get snort running so that it can analyze network behavior.

2.2.1.1 Advantage of SNORT over other tools.

1. **Scalability**: Snort can be successfully deployed on any network environment.
2. **Flexibility and Usability**: Snort can run on various operating systems including Linux, Windows, and Mac OS X.
3. **Live and Real-Time**: Snort can deliver real-time network traffic event information.
4. **Flexibility in Deployment**: There are thousands of ways that Snort can be deployed and a myriad of databases, logging systems, and tools with which it can work.
5. **Speed in Detecting and Responding to Security Threats**: Used in conjunction with a firewall and other layers of security infrastructure, Snort helps organizations detect and respond to system crackers, worms, network vulnerabilities, security threats, and policy abusers that aim to take down network and computer systems.
6. **Modular Detection Engine**: Snort sensors are modular and can monitor multiple machines from one physical and logical location. Snort be placed in front of the firewall, behind the firewall, next to the firewall, and everywhere else to monitor an entire network. As a result, organizations use Snort as a security solution to find out if there are unauthorized attempts to hack in the network or if a hacker has gained unauthorized access into the network system.

2.2.1.2 Configuration of SNORT

Setting up Snort from the source code consists of a couple of steps: downloading the code, configuring it, compiling the code and lastly installing it. First up make a temporary

download folder to your home directory and then move into it with the these commands

```
mkdir ~/snort_src

cd ~/snort_src
```

After it is downloaded we have to configure the downloaded code. Following is the snapshot of the commands for the configuration of SNORT.

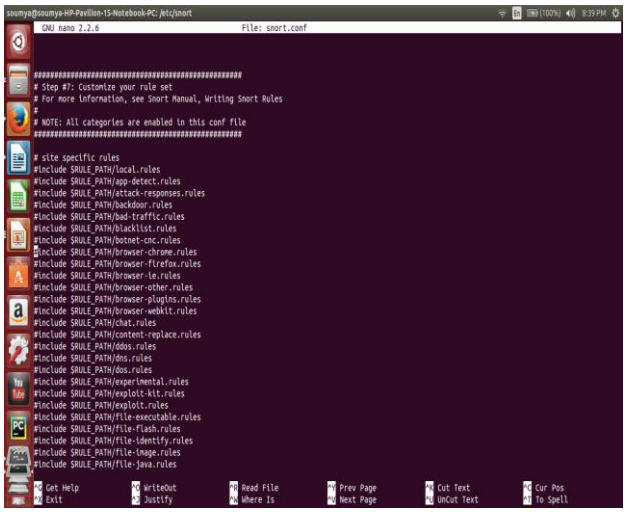


Fig -2: Snort.Config

2.2.2 Snort in packet sniffer mode

If we want to print out the TCP/IP packet headers to the screen following command is used:

```
./snort -v
```

If we want to see the application data in transit, following command is used

```
./snort -vd
```

This instructs Snort to display the packet data as well as the headers.

If we want more descriptive display, showing the data link layer headers, following command need to be run.

```
./snort -vde
```

As an aside, notice that the command line switches can be listed separately or in a combined form. The last command could also be typed out as:

```
./snort -d -v –e
```

This will produce the same result.

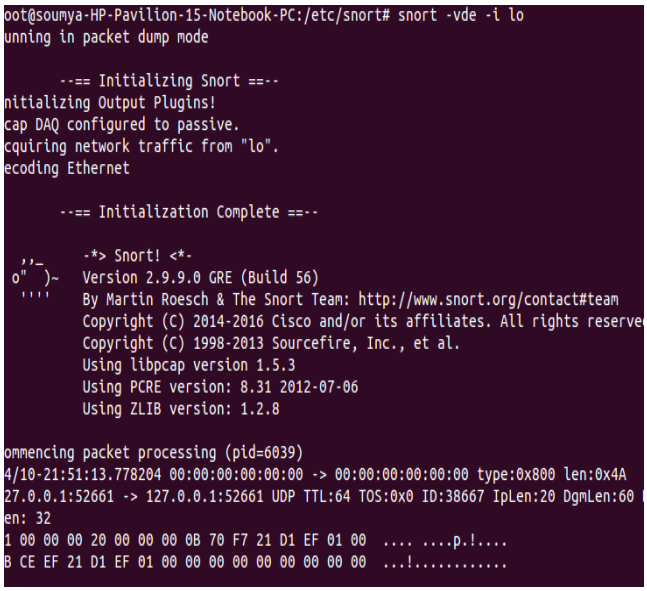


Fig -3: Running snort in packet sniffer mode

2.2.3 Snort using snort.conf file

Snort uses a configuration file at start up time. A sample configuration file snort.conf is included in the Snort distribution. You can use any name for the configuration file, however snort.conf is the conventional name. You use the -c command line switch to specify the name of the configuration file. The following command uses /opt/snort/snort.conf as the configuration file.

We can also save the configuration file in our home directory as snortrc, but most commonly used method is specifying it on the command line. There are other advantages to using the configuration file name as a command line argument to Snort. It is possible to invoke multiple Snort instances on different network interfaces with different configuration.

$ *sudo /usr/local/bin/snort -A console -q -u snort -g snort -c /etc/snort/snort.conf -i lo*

This command should be run in our terminal to run snort using our snort configuration file. It can be modified

according to the user suitability. Snort has various modes; few of them are listed here

Description of the command:

-c: specifies the config file

-i : specifies the interface mode , if a loopback address is running then “lo” will be written , for Ethernet “eth0” or “eth1” will be written.

-A: It will print the output to the console

Once we run this command, then type $ ping 127.0.0.1

We should see that the snort logs this packet and displays it on the terminal. Here is the image of the terminal logging the ping packets.

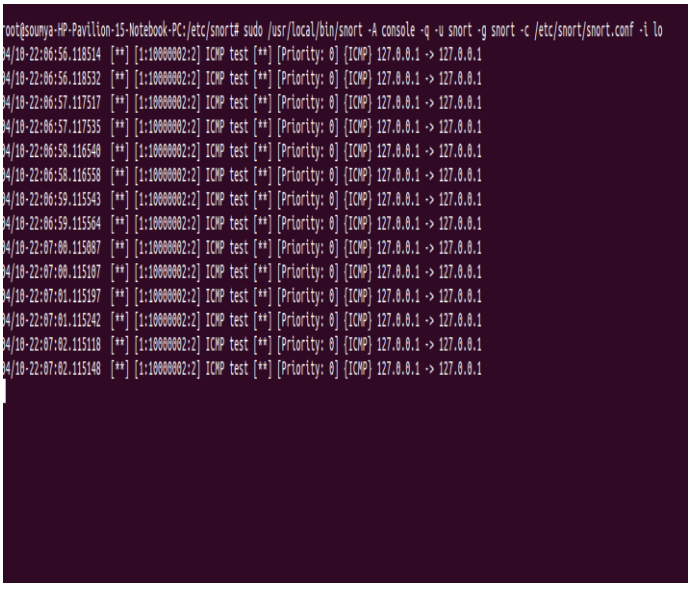


Fig -4: Ping packets are displaying a message “ICMP test”, it will print whatever message is provided by the user in rules file.

2.3 Writing rules

Rules are written by the user, snort will log the packets and generate alert if there if finds any match with the rules that user defined in the rules file. Here is an example of how to write rules.

1. alert ip $EXTERNAL_NET any -> $HOME_NET any (ip_proto:igmp; rev:1000000)

For igmp traffic

2. alert tcp any any -> any 80 (content:"ABC"; content:"EFG"; http_raw_cookie;rev:1000001)

detects unnormalised cookie header

3. alert tcp any any -> any (msg:"exploit"; content:"|90|"; rev:1000002)

4.alert $EXATERNAL_NET any -> $HOME_NET any (flags: SF,12; msg:"SYN FIN scan"; rev:1000003)

5.alert any any -> $HOME_NET 21(msg:"Incoming FTP";rev:1000004)

6. alert tcp $HOME_NET any -> $EXTERNAL_NET 80(msg:”Invalid Content Found”; content:”terrorism”;nocase; rev:1000005)

7.alert icmp $EXTERNAL_NET any -> $HOME_NET any(msg:”PING ALERT”; icode:0;itype:8;rev:1000006)

8.alert tcp $EXTERNAL_NET any -> $HOME_NET 80 (msg:"EXPLOIT ntpdx overflow"; dsize:>128; classtype:attempted-admin; priority:10)

9.log tcp !192.168.0/24 any -> $HOME_NET any (msg: "mounted access" ;)

10. $ alert tcp any any -> $HOME_NET 21 (msg:"Possible FTP Login"; sid:1000004; rev:004;)

alert: it will generate alert packets

tcp: protocol which is being used

any: it specifies that log packets coming from any IP address.

$HOME_NET: It is our local IP address, it is mapped in snort.conf file

21 : It tells snort to generate alerts of any packet which try to send request to port 21.

3. Barnyard

Barnyard automatically inserts all the alerts generated by snort into a database. In our research paper we have used mysql to access the information given by snort. Barnyard is easy to set up and runs by typing the following command:

$ sudo barnyard2 -c /etc/snort/barnyard2.conf -d /var/log/snort -f snort.u2 -w /var/log/snort/barnyard2.bookmark -g snort -u snort

-d: tells where to save the output
-g: run as a specific user

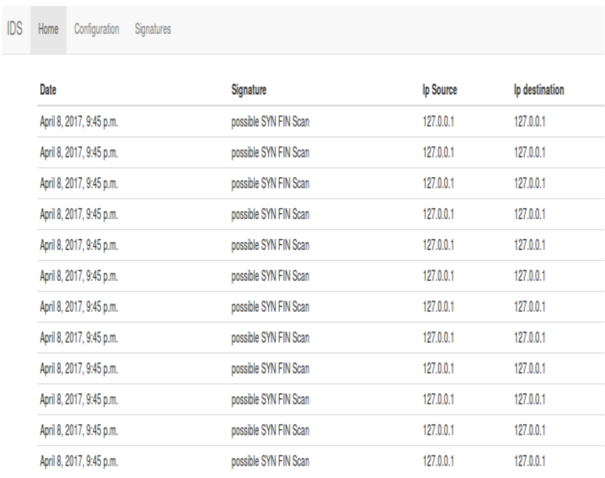
-c: specifies the user

-w: gives the bookmark file used in barnyard

4. Results

The result of our project will be the display of all packets which matches the snort defined by the administrator. The information will get stored in a mysql database using which we have made a UI to display all the necessary Ind=formation about the alert generated. The information includes Source IP, Destination IP, Alert generated, Date and Time of when the packet was received.

In this case we have used a single system for testing purpose therefore the source and destination IP are my loopback address , however when run on a server it will give the Source and Destination IP of the systems generating and receiving the packets.



Date	Signature	Ip Source	Ip destination
April 8, 2017, 9:45 p.m.	possible SYN FIN Scan	127.0.0.1	127.0.0.1
April 8, 2017, 9:45 p.m.	possible SYN FIN Scan	127.0.0.1	127.0.0.1
April 8, 2017, 9:45 p.m.	possible SYN FIN Scan	127.0.0.1	127.0.0.1
April 8, 2017, 9:45 p.m.	possible SYN FIN Scan	127.0.0.1	127.0.0.1
April 8, 2017, 9:45 p.m.	possible SYN FIN Scan	127.0.0.1	127.0.0.1
April 8, 2017, 9:45 p.m.	possible SYN FIN Scan	127.0.0.1	127.0.0.1
April 8, 2017, 9:45 p.m.	possible SYN FIN Scan	127.0.0.1	127.0.0.1
April 8, 2017, 9:45 p.m.	possible SYN FIN Scan	127.0.0.1	127.0.0.1
April 8, 2017, 9:45 p.m.	possible SYN FIN Scan	127.0.0.1	127.0.0.1
April 8, 2017, 9:45 p.m.	possible SYN FIN Scan	127.0.0.1	127.0.0.1
April 8, 2017, 9:45 p.m.	possible SYN FIN Scan	127.0.0.1	127.0.0.1
April 8, 2017, 9:45 p.m.	possible SYN FIN Scan	127.0.0.1	127.0.0.1

Fig -5: Matched packets

6. Reference

[1] Rafeeq Ur Rehman, Intrusion Detection Systems with Snort: Advanced IDS Techniques with Snort, Apache, Prentice Hall PTR

[2] Douglas J. Brown, Bill Suckow and Tianqiu Wang, A Survey of Intrusion Detection Systems.

[3] Hilmi Gunes Kayacik, A. Nur Zincir-Heywood, A case study of three open source security management tools.

[4] Vern Paxson, Jim Rothfuss, Brian Tierney, Bro Quick Start Guide, University of California and the International Computer Science Institute.

[5] Martin Roesch, Snort - Lightweight Intrusion Detection for Networks, 13th USENIX Systems Administration Conference – LISA '99, Seattle, Washington, November 1999