Edge Security

1Umar Shakeeb, 2Sanket Milke
1UG Scholar, 2Assistant Professor
MGM JNEC Aurangabad, Maharashtra, India

***Abstract* - Corporate organizations are vulnerable to deceptions and vulnerabilities, making them an excellent target for hackers looking to steal sensitive data. Organizations use an Interruption Identification Framework to protect themselves against these attacks (IDS). IDS inspects each packet that passes through the organization and alerts the organization if there is any indication of malignant activity. Snort is a well-known open-source tool that may be used for this design. Snort gathers specific information about packages passing through the organization and issues warnings if they fit a set of predetermined criteria or markings assigned by the organization. In this paper, we show how an IDS can detect and prevent network-based attacks using libpcap and Snort.**

***keywords* - Intrusion Detection System (IDS), Network Security, False Alarms, Network-based Attacks,Signature-based Detection, Intrusion Prevention,Snort**

1. Introduction

Network security is conceivably of the best test that associations are looking from time to time. There are many undertakings by dull cap software engineers to break and mull over security of the's association, some of which are even productive. As the usage of the web extending, these pernicious activities are obtaining unmistakable quality among the dull covers.
Reliably a ton of data is being delivered and passed on and heaps of this data holds sensitive information about the association and its laborers. Subsequently, getting an association is one of the fundamental endeavors for an association to make due. To simplify this and more successful we use Interference Disclosure Structure, it helps with social occasion information about any malevolent package that passes across an's association.

1.1 Intrusion Detection System

An intrusion detection system (ID) is a sort of safety framework for PCs and PC organizations. Interruption Identification fundamentally helps in distinguishing external and internal assaults performed by either clients or programmers. An ID framework gathers data from different sources and investigates data from different regions inside a PC or an organization to distinguish conceivable security breaks, which incorporate the two interruptions (assaults from outside the association) and abuse (assaults from inside the association). ID utilizes weakness evaluation (some of the time alluded to as filtering), which is an innovation created to survey the security of a PC framework or organization.

1.2 Advantages of IDS

- Track any progressions in the way of behaving of the organization.
- Investigates framework movement
- Can separate among ordinary and unusual exercises in the network
- Computerized Automation

1.2 disadvantages of IDS

- Some of the time gives misleading problems i.e., the packet wasn't malevolent however IDS could in any case produce a caution.
- Tedious
- Isn't 100 percent protected from assaults

2. Tools used in IDS

An intrusion detection system may be implemented using a variety of techniques. Some of the most popular tools include

- SNORT
- Security Onion
- WEKA
- OSSEC

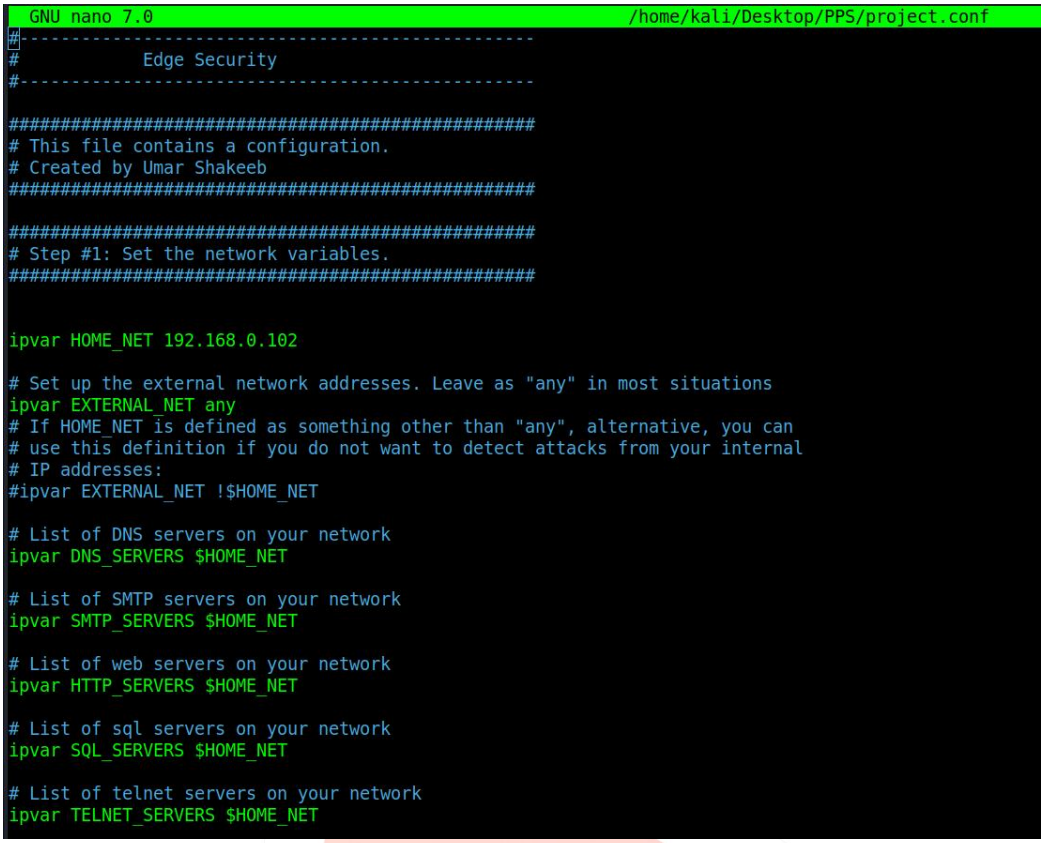
3. Edge Security

ES is a light-weight interruption identification instrument that logs the bundles getting through the organization and investigates the parcels. Snort checks the bundles coming contrary to the guidelines composed by the client and produces cautions in the event that there are any matches found. The principles are composed by the client in a text document that is connected with a project.conf record where every one of the designs are referenced. There are a couple of orders which are utilized to get project running with the goal that it can investigate network conduct.

3.1 Advantages of Edge Security over other tools.

1. Scalability: ES can be successfully deployed on any network environment.
2. Flexibility and Usability: Snort can run on various operating systems including Linux, Windows, and Mac OSX.
3. Live and Real-Time: ES can deliver real-time network traffic event information.
4. Flexibility in Deployment: There are thousands of ways that ES 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, ES help organizations detect and respond to system crackers, worms, network vulnerabilities, security threats, and policy abusers that aim to take down network and computer systems.



Configuration File

4. Edge Security using project.conf file

ES (Edge Security) uses a configuration file at start-up time. A sample configuration file snort. You use the - c command line switch to specify the name of the configuration file. The following command uses /opt/snort/ES.conf as the configuration file. We can also save the configuration file in our home directory as ES, but the most commonly used method is specifying it on the command line. There are other advantages to using the configuration file name as a commandline argument to a project. It is possible to invoke multiple Snort instances on different network interfaces with adifferent configuration.

$ sudo -A console -q -i wlan0 -c /etc/snort/snort.conf

This command should be run in our terminal to run Edge Security using our project configuration file. It can be modified according to user suitability. Snort library has various modes; a few of them are listed here

Description of the command:

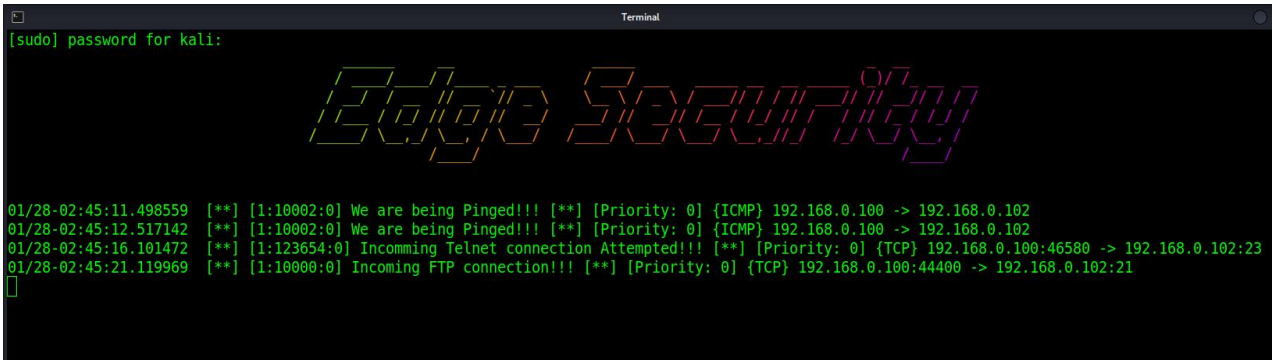
-c: specifies the config file

-i: specifies the interface mode, if a loopback address is running then “wlan0” 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 192.168.0.102

We should see that the project. Logs this packet and displays it on the terminal. Here is the image of the terminallogging the ping packets.



Monitoring Screen

4.1 Writing rules: -

Rules are written by the user; it generates an alert if there if finds any match with the rules that the user defined in the rules file. Here is an example of how to write rules.

1. alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"We are being Pinged!!!"; icode:0; itype:8; sid:10002;)

for ICMP ping.

2. alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS 80 (msg:"Triggered /etc/passwd"; flow:to_server,established; content:"/etc/passwd"; nocase; sid:1122;)

When someone try OS command Injection

3. alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"XSS attack attempted (cross site scripting attempt)"; flow:to_server,established; content:"SCRIPT"; nocase; sid:1497;)

When XXS (Cross Site Scripting) attack is perform

4. alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"COMMUNITY SQL-INJECTION BXCP Sql Injection attempt"; flow:to_server,established; uricontent:"/index.php"; nocase;uricontent:"where="; nocase; uricontent:"union"; nocase; uricontent:"select"; nocase; sid:100000690; rev:2;)

When SQL injection attack is performed

5. alert tcp any any -> $HOME_NET 80 (flags:S; msg:"Possible Dos Attack Type:SYN FLOOD";flow:stateless; sid:3; detection_filter:track by_dst, count 20, seconds 10;)

When DOS (Denial of Service) attack is performed

6. alert tcp $EXTERNAL_NET any -> $HOME_NET 21 (msg:"Incoming FTP connection!!!"; flags:S; sid:10000;)

5. Result

The consequence of our undertaking will be the presentation of all packets which match the Snort characterized by the executive. The data which gets on Screen is Source IP, destination IP, Alert produced, and Date and Time when the packet was gotten. For this situation, we have involved a solitary framework for the end goal of testing, consequently, the source and objective IP are my loopback address, be that as it may, when run on a server it will give the Source and Destination IP of the frameworks producing and getting the parcels. For this situation, we have involved a solitary framework for the end goal of testing consequently the source and objective IP are my loopback address, be that as it may, when run on a server it will give the Source and Destination IP of the frame works producing and getting the packets

6. CONCLUSIONS

The objective of this paper was to plan and assess a framework that would inactively screen the remote organization traffic of a little home organization. Such a framework, in the event of assault identification, endeavors to disturb distinguished assaults utilizing the packet infusion technique. During tests was an endeavor to complete DoS by ICMP flood on the access point. This kind of assault was picked as a result of its straightforward identification and execution. The result of Snort measurements was shown diminishing of assailant's traffic by 95% with proposed IDPS framework conveyed - when contrasted with how much aggressor's traffic without organization of IDPS. From the outcomes we presume that the deauthentication of the assailant effectively disassociates the aggressor from the access point, and in this manner restricts the assault. Starting from the commencement of the assault the framework had the option to respond in 0.2 seconds, as is finished up from diagrams produced by Wireshark measurements. Aireplay-ng standard result was utilized to ascertain measurements of deauthentication. From these measurements is expected that the deauthentication of the assailant was in all cases effective, by the by, the level of gotten deauthentication affirmations from the aggressor was simply 42% when contrasted with the 100 percent of gotten affirmations from the access point.

7. REFERENCES

[1] Karen, Scarfone & Peter Mell, (2007) “Guide To Intrusion Detection And Prevention Systems (IDPS)”. Washington, D.C.: National Institute of Standards and Technology, Special Publication 80094, 128 p.

[2] Michael Rash, (2007) “Linux Firewalls - Attack Detection And Response With Iptables”, Psad And Fwsnort.San Francisco: No Starch Press, 388 p.

[3] Allen, Lee (2012) “Advanced Penetration Testing for Highly--Secured Environments: The Ultimate SecurityGuide”. Birmingham: Packt Publishing Ltd., 414p.

[4] “Linux Wireless - Hostapd Linux Documentation Page”. [online]. [cit. 14. April. 2014]. Available online: <http://wireless.kernel.org/en/users/Documentation/hostapd>.

[5] KAZIENKO, Przemyslaw; DOROSZ, Piotr. Intrusion detection systems (IDS) Part 2-Classification; methods; techniques. WindowsSecurity. com, 2004. [10] CARL, Glenn, et al. Denial-of-service attack-detection techniques. Internet Computing, IEEE, 2006, 10.1: 82-89.

[6] Gupta, R., Singh, S., Verma, S. and Singhal, S., 2017. Intrusion detection system using SNORT. International Research Journal of Engineering and Technology (IRJET), 4(04), pp.2100-2104.