Published by :
http://www.ijert.org

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 9 Issue 07, July-2020

Snort IDPS using Raspberry Pi 4

Pankaj Varma, Anas Siddiqui, Parag Vadher
Students, Information Technology,
M. H. Saboo Siddik College of Engineering,
Mumbai, India

Vikas Baloda
Assistant Professor, Information Technology,
M. H. Saboo Siddik College of Engineering,
Mumbai, India

***Abstract*—Network attacks are becoming increasingly threatening in the age of the wireless. This makes it an imminent hostility against poorly protected networks. The scale and mobility of the Firewalls make security a costly matter for comparatively smaller organizations. In this paper we design and implement a distributed Intrusion Detection Prevention System system with a cost effective Raspberry Pi 4 using Snort Engine. This paper looks to build a portable Intrusion Detection Prevention System using TALOS/VRT Rules. We simulate some attacks to check the security of the Intrusion Detection Prevention System system. Snort community rules and registered rules are a huge part of our project. Some of these tweaked rules are in the custom rules file. Also a part of an economic Intrusion Detection and Prevention System is the Raspberry Pi 4B and PenTBox Honeypot.**

Keywords—IDPS - Intrusion Detection and Prevention System

I. INTRODUCTION

The need for security analytics tools:

An Investigation Report from Verizon found that the Educational sector is being plagued by errors, social engineering and insufficiently secured email credentials[1]. With regards to attacks, DoS attacks account for more than half of all incidents in the Education Sector. One of the reasons it's so challenging to detect attacks is because they happen very quickly. In addition to this, data indicating an attack is often spread across servers, application logs network devices, and endpoints. Thus it is difficult to analyze a breach in progress which is why it even hinders the capability to assess its effect. Before we go further, let's understand the basics.

A computer device (including hardware, added components) because it contains and stores data is certainly an asset. Because most computer hardware is useless without its programs, software in it is also an asset. Software are those which include an operating systems, utilities and the devices handle applications such as document-processing, media players or programming codes; and even programs that you may have written yourself. A lot of hardware and software is off-shelf, meaning that it is commercially available and that you can easily get a replacement. The thing that makes your computer unique and important to you is its content: pictures, papers, documents, music, emails, calendar information, projects, ebooks, contact information, code you create, run and like. This means that data items on a computer are assets. Unlike most hardware, data can be almost impossible-to recreate or replace.

Existing Intrusion detection systems have a preprocessed code that blocks any anomaly in the system. But this fails when a new way to access information is used by attackers. With the use of HoneyPot, we can collect information and detect attack patterns. After doing so, we can respond to this evidence by building better defenses and countermeasures against future security threats. A high interaction HoneyPot works in tandem with the IDPS Existing datasets will train the algorithm for IDPS. Once the IDPS is set up, the log files from the HoneyPot can be used for future training and improvement. The IDPS software will be uploaded to a couple of Raspberry Pi boards and these will then be connected to each other with one being the master. These will communicate with one another so as to learn from each other in case of future attacks.

Firewalls: The foundation for installing, configuring a firewall is the organization's overall security policy. After you have a solid security policy as your guide, you can design security configurations to support your organization's goals[2].

Intrusion Detection and Prevention System (IDPS): firewalls and proxy servers block intruders or malicious code from entering a net- work. However, using an IDPS with these tools offers an additional layer of protection for a network. An intrusion detection and prevention system (IDPS) works by recognizing the signs of a possible attack and sending a notification to an administrator that an attack is happening (intrusion detection). Some traffic generated can trigger a response which attempts to actively combat the threat (intrusion prevention)[3]

II. LITERATURE REVIEW

Existing System

As of late 2020, our college does not implement any network IDS to monitor network operations among its laboratories. A machine to detect students' activities during lab sessions is equipped by a system monitor that needs heavy monitoring as there is no way to keep track of past activities. This proves to be a very inefficient system should the lab assistant or teacher be occupied elsewhere.

There is no provision to keep track of activities that may ill-fit the network that holds a college of more than a few hundred machines connected. The network which is responsible for a lot of daily classes and studying more than often breaks down and looks like an easy target for an ameteur hacker. The institute suffers from a tedious network management and feels a dire need to update their network architecture.

- ***Using Snort For a Distributed Intrusion Detection System (Michael Brennan, 2002, SANS Institute 2020)***[4]

In organizations that have little to no network monitoring, having an expensive tool for the same is generally not

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 9 Issue 07, July-2020

plausible. This paper focuses on the need for an Intrusion Detection System for companies that cannot afford one of the more quality commercial Intrusion Detection Systems available today. Snort being the most popular free network Intrusion Detection Systems is a better choice for such organizations. Snort's ease of configuration, flexibility and strong packet analysis make it a powerful intrusion detection device. Rules are very easily written, flexible and can be easily inserted into the rule file base. The placement of the remote sensors is one of the most important parts of this system. If the remote sensors are misplaced packets may not be seen by the detection system. Andrew Baker explains it best when he remarks, "In order for Snort to be most effective, it needs to be positioned where it will see the most traffic possible"(Baker). The paper details the placement of snort routers, the system setup(Management server and the remote server) and the network placement. This paper reviewed is an inexpensive solution for setting up distributed network intrusion detection system with Snort and other open source tools.

- ***Pi-IDS: Evaluation of Open-Source Intrusion Detection Systems on Raspberry Pi 2 (Ar Kar Kyaw, Yuzhu Chen and Justin Joseph, 2015, Digital Forensic Research Labs, Auckland University of Technology Auckland, New Zealand)***[5]

The research conducted investigates some important questions, such as,

Q1. Is the Raspberry Pi 2 capable of running Snort IDS and Bro IDS?

Q2. Are IDSs that are running on Raspberry Pi 2 capable of detecting network attacks?

Q3. Can a Raspberry Pi 2 handle all the network traffic?

Q4. Is the performance of Snort IDS better than Bro IDS running on Raspberry Pi 2 model?

The Raspberry Pi 2 handles an immense load of traffic to measure the scope for the 3rd question.

It takes on different network attacks within the LAN to collect the data. Three different types of network attacks such as Synchronization (SYN) Flood, Address Resolution Protocol (ARP) Spoofing and Port Scanning were selected to stress the system. The Bro IDS and Snort IDS are compared on the basis of RAM, Memory used and packet loss for the above mentioned attacks. The results and answers to the questions raised by the paper proposed are answered after experimental testing and excessive testing.

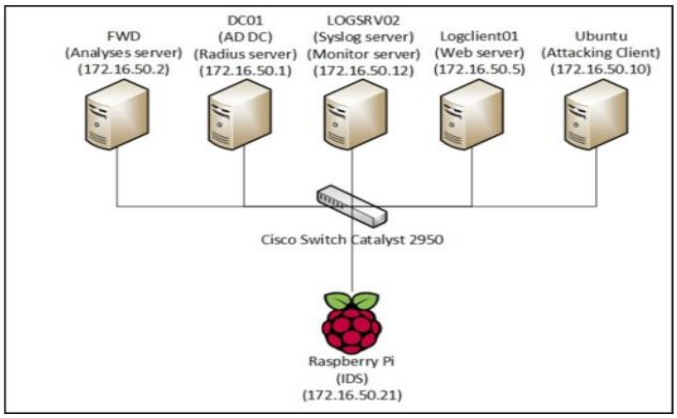

Figure 1: System Architecture for experiment

- ***A Linux-based IDPS using Snort (Ghilman Ahmed, Muhammad Naeem Ahmed Khan, Muhammad Shamraiz Bashir – Shaheed Zulfikar Ali Bhutto Institute of Science and Technology, 2015)***[6]

The components of the framework consist of a firewall and a load balancer. It takes help of the Preprocessor component of Snort once the load balancer distributes the load on multiple servers with its load balancer. Real-time intrusion detection with prevention by Intrusion Detection and Prevention Systems (IDPS) rebuilds the security of a network to an advanced level by toughening the network against malicious activities. This intrusion prevention system is implemented by using the kernel codes of Snort as well as Netfilter. This system is based on three phases: intrusion detection system model; the policy control model; and the firewall. As part of the experimentation process, the detection engine successfully analysed and blocked DoS attacks. The authors analysed the strength of this system against attacks such as HTTP attacks and MITM(man-in-the-middle) attacks by using the detection engine with honeypot support.

III. KEY COMPONENTS

A. Snort

Snort is an open source network intrusion prevention system, which is capable of performing real-time network traffic analysis and packet logging on Internet networks. It can perform protocol analysis, content searching or matching, and can be used to detect a variety of attacks and probes, such as buffer overflows, CGI attacks, port scans, OS fingerprinting attempts, and more[7]. This will be a strategy to alert the system against intruder attacks. It can perform protocol analysis, content searching/matching, The following snort rules are predefined and available freely directly on their website[8]. The rules that comprise the IDS are the ones that start with alert. Network attacks are becoming increasingly threatening in the age of the wireless. This makes it an imminent hostility against poorly protected networks. The scale and mobility of the Firewalls make security a costly matter for comparatively smaller organizations. In this paper we design and implement a distributed IDPS system with a cost effective Raspberry Pi 4 using Snort Engine. This paper looks to build a portable IDPS using TALOS/VRT Rules. We simulate some attacks to check the security of the IDPS system.seen regardless of protocol or encoding. This removes the rule writer from the burden of the following:

a. Maintaining multiple rules to detect the same files and content over unique protocols.

b. Maintaining multiple rules to detect the same files or content traversing in many directions.

c. Creating or modifying Snort rules when new protocols are added.

d. No having to worry about and potentially replace flowbits options in rules [9].

DAQ variables to enable the running of AFPacket in inline (IPS) mode are pcap, ipfirewall and dump mode. [10]

```
parag@parag:/etc/snort/rules$ snort --daq-list
Available DAQ modules:
pcap(v3): readback live multi unpriv
ipfw(v3): live inline multi unpriv
dump(v3): readback live inline multi unpriv
afpacket(v5): live inline multi unpriv
```

Figure 2: DAQ List

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 9 Issue 07, July-2020

B. Barnyard 2 and Database Logging

The alerts generated are used for the detecting and will be stored in a remote server that can be viewed from the site itself. This structured format of the data is necessary for further analysis. Once the data is stored in Mysql readable format we can use the data for training of machine learning algorithms or for future analysis. We have made use of Mysql database and Barnyard2 for the logging of tables in our system. This is relatively easy since Barnyard2 now allows only unified2 as the input plugin. This means that the output for Mysql will be a unified or .u2 extension saved. We create a database named snort and give it necessary privileges.

C. PentBox Honeypot

The PenTBox Honeypot 1.8 is a security feature added to enable the users to constantly review attacks and upgrades on the Intrusion Detection Prevention System as required.

Figure 3: Pentbox Initialization

D. Raspberry Pi

The IDPS software will be uploaded to a couple of Raspberry Pi boards and these will then be connected to each other with one being the master. These will communicate with one another so as to learn from each other in case of future attacks.

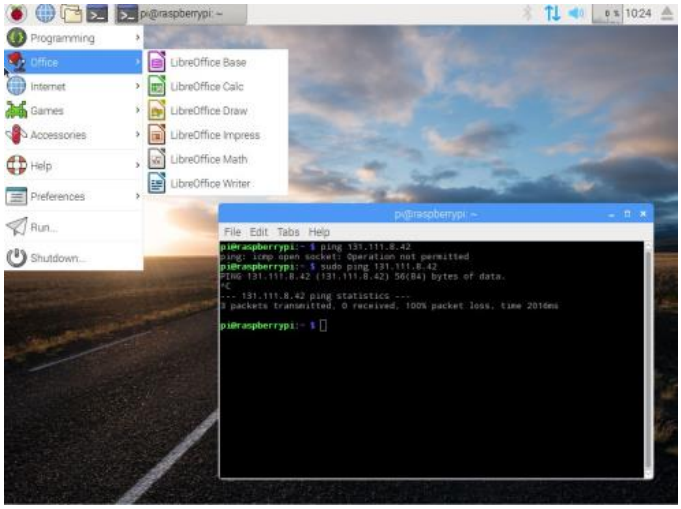

Figure 4: Raspbian OS tested

IV. SOME COMMON MISTAKES

1. The Intrusion Detection & Prevention System is not tweaked to completely fit the college network, this in turn will affect the amount to details being logged to contribute to a huge database. The alerts too may come off as exceedingly cautious or unwanted.
2. The hardware that we have used i.e. Raspberry Pi 4B is a newer model with various bugs that have yet to be solved. Although we have solved a major part of the constituting errors and bugs, having less or no trained personnel at the site of deployment can be a problem.
3. The rate of packet drop(without packet checking) will increase when the load increases the capacity of the Raspberry Pi 4B.
4. The heating of Raspberry Pi 4B is an eminent threat with the powerful CPU that it provides with the extensive RAM for much faster computing.
5. It must be always connected to the network and the Internet to provide logs to the database.

V. CONCLUSION

Our approach combines IoT with networking to present a viable and an economic solution for small organizations and businesses. An Intrusion Detection System is only as strong as customized it is for the network for which it is built. Although the Intrusion Detection Prevention System is built keeping ports and common ports in mind, we are not as informed about the department level network configurations as some of our faculty and Network administrators may be. The Honeypot and its separate log files, the IDS's log files in the database, the log files generated by alerts and log files make it a very manual process. Yet the inline mode makes it the only automated process out of the entire system. The system will need refining once it is implemented.

VI. FUTURE SCOPE

Despite the limitations, they can be overcome with a little more work than what he intended to complete.

1. Correction. Once the IDPS is implemented, it needs to reiterated to check the attacks its prevents and packet drops without checking. This can only be done once the machine is installed. Tweaking the made IDPS and editing rules for further analysis can be a great step ahead in the direction which enables a better system.

2. As time progresses the Raspberry Pi will have Kernel upgrades. One of the upgrades to be released as soon as October 2020. This will bring a stabler Model for testing and implementation.

3. The load needs to be distributed, this can be done by adding more Raspberry Pis and making them communicate among themselves.

4. Cooling systems need to be installed on Raspberry Pis and they need to be checked repeatedly in case of overheating.

5. The Raspberry Pi needs to be installed as a gateway for the network.

The Intrusion Detection Prevention System works in inline mode to prevent attacks, in data logging mode as well as implements a Hoenypot to make it a complete A Machine

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 9 Issue 07, July-2020

learning algorithm can be built to learn from the distributed system to prevent attacks to make it a completely automated system.

REFERENCES

[1] Verizon website: https://enterprise.verizon.com/resources/reports/dbir/2019/educational-services
[2] Guide to network defense and countermeasures, Randy Weaver, Dawn Weaver, Dean Farwood, chp1, pg 15.
[3] Guide to network defense and countermeasures, Randy Weaver, Dawn Weaver, Dean Farwood, chp1, pg 16.
[4] Michael P. Brennan "Using Snort For a Distributed Intrusion Detection System", version 1.3 in SANS Institute 2020, January 29, 2002
[5] [4] Ar Kar Kyaw, Yuzhu Chen and Justin Joseph "Pi-IDS: Evaluation of Open-Source Intrusion Detection Systems on Raspberry Pi 2" in "ISBN ©2015 IEEE", 2015
[6] [5] Ghilman Ahmed, Muhammad Naeem Ahmed Khan, Muhammad Shamraiz Bashir, Shaheed Zulfikar, "A Linux-based IDPS using Snort"
[7] https://www.snort.org/faq/what-is-snort
[8] Available rules www.snort.org/documents
[9] Available rules www.snort.org/documents
[10] Firewall and pcap instructions https://blog.talosintelligence.com/2010/08/snort-29-essentials-daq.html