PROJECT REPORT

Project Name: Configuring and Implementing Snort as an IDS/IPS System

Course Name: Advanced Diploma in Cyber Defence (ADCD)

Submitted By: Razan.P & Mohamed Akhil Abdul Kder

Trainer: Alex

Institute: Red Hacker Academy Thrissur

Acknowledgement

We would like to take this opportunity to express our heartfelt gratitude to our trainer, Alex, and the entire team at the institute for their invaluable guidance and unwavering support throughout the duration of this project. Their expertise, insights, and encouragement have been instrumental in shaping our understanding of the complexities involved in configuring and implementing the Snort IDS/IPS system. Alex's mentorship has not only enhanced our technical skills but has also inspired us to approach challenges with a more analytical and solution-oriented mindset.

We would also like to extend our sincere appreciation to our peers, whose collaboration and camaraderie made this journey both enjoyable and enriching. The discussions we shared and the diverse perspectives we exchanged played a crucial role in refining our ideas and elevating the quality of this report.

Furthermore, we are deeply grateful for the various resources—books, articles, and online materials—that provided essential information and insights, enabling us to delve deeper into the subject matter. Each of these contributions has been vital to the successful completion of this work.

In conclusion, we acknowledge that this project would not have been possible without the collective support and encouragement of all those mentioned above. Thank you for being an integral part of this learning experience.

Abstract

This project delves into the deployment and configuration of Snort, a widely recognized and open-source Intrusion Detection and Prevention System (IDS/IPS). With the increasing prevalence of sophisticated cyber threats, there is a growing need for advanced tools capable of monitoring, detecting, and preventing malicious activities within a network. Snort, known for its powerful packet inspection capabilities and adaptability, is an ideal solution for meeting these challenges.

The primary focus of this project is to explore Snort's role in enhancing network security by detecting threats in real-time and preventing potential breaches. This is achieved through a systematic approach that includes the installation of Snort, the creation and implementation of custom detection rules, and the configuration of its IDS and IPS functionalities. Additionally, the project employs simulations of various cyber threats, such as Distributed Denial of Service (DDoS) attacks and unauthorized access attempts, to evaluate the effectiveness of Snort in identifying and mitigating these risks.

This report comprehensively documents the entire process, beginning with pre-installation requirements and extending to the detailed steps of configuring Snort for specific scenarios. It highlights the challenges encountered during the deployment and testing phases, providing insights into troubleshooting and optimizing Snort’s performance. The findings emphasize the importance of proactive security measures and demonstrate how Snort can serve as a vital component in safeguarding critical network infrastructures.

By showcasing the practical implementation of Snort as an IDS/IPS, this project contributes to a deeper understanding of its capabilities and potential applications. It aims to offer valuable insights for cybersecurity professionals and network administrators seeking robust solutions to strengthen their security posture in an increasingly vulnerable digital landscape.

Table of Contents

Introduction

Intrusion Detection and Prevention Systems (IDS/IPS) are vital components in safeguarding modern networks against an ever-evolving array of cyber threats. With the proliferation of digital systems and the increasing reliance on interconnected networks, the risk of unauthorized access, malicious attacks, and data breaches has grown significantly. Organizations today must adopt robust solutions to detect, analyze, and prevent these threats before they can compromise the integrity, confidentiality, and availability of their systems.

This project explores Snort, a highly popular open-source IDS/IPS tool renowned for its flexibility, scalability, and effectiveness in identifying and mitigating network threats. Developed by Cisco, Snort combines real-time packet analysis with advanced rule-based detection capabilities, making it a preferred choice for both small-scale networks and large enterprises. Its ability to detect a wide range of threats, from malware and exploitation attempts to port scans and Distributed Denial of Service (DDoS) attacks, showcases its importance in strengthening network defenses.

The primary goal of this project is to configure Snort for real-time traffic analysis and proactive threat detection. This involves setting up Snort in both IDS and IPS modes, developing custom detection rules tailored to specific threat scenarios, and simulating real-world attack patterns to assess its effectiveness. By doing so, the project not only demonstrates the practical implementation of Snort but also highlights its significance in maintaining a secure and resilient network environment.

This report provides a step-by-step guide to the installation and configuration of Snort, detailing its integration into a controlled test environment. It also discusses the challenges encountered during the setup process and the strategies employed to overcome them. Through this project, we aim to emphasize the critical role that IDS/IPS systems play in modern cybersecurity and showcase how Snort can be effectively leveraged to protect against a variety of malicious activities.

Objective

The primary objective of this project is to explore and demonstrate the capabilities of Snort as a robust Intrusion Detection and Prevention System (IDS/IPS) within a controlled environment. By deploying Snort, we aim to configure it for both IDS and IPS functionalities, enabling real-time monitoring of network traffic and proactive defense against potential threats. A key focus of this project is the creation and implementation of custom detection rules tailored to simulate real-world attack scenarios, such as unauthorized access attempts, Distributed Denial of Service (DDoS) attacks, and malware activities. Through these simulations, we seek to evaluate Snort's performance in accurately detecting malicious patterns and mitigating threats effectively. Furthermore, this project emphasizes the importance of documenting the entire process, including installation, configuration, testing, and troubleshooting. By sharing our findings, we aim to provide valuable insights and practical guidance for implementing Snort in professional environments, contributing to a deeper understanding of its role in enhancing network security. Ultimately, this project seeks to highlight the significance of IDS/IPS systems in today's cybersecurity landscape and to showcase how Snort can be leveraged to safeguard networks against increasingly sophisticated cyber threats.

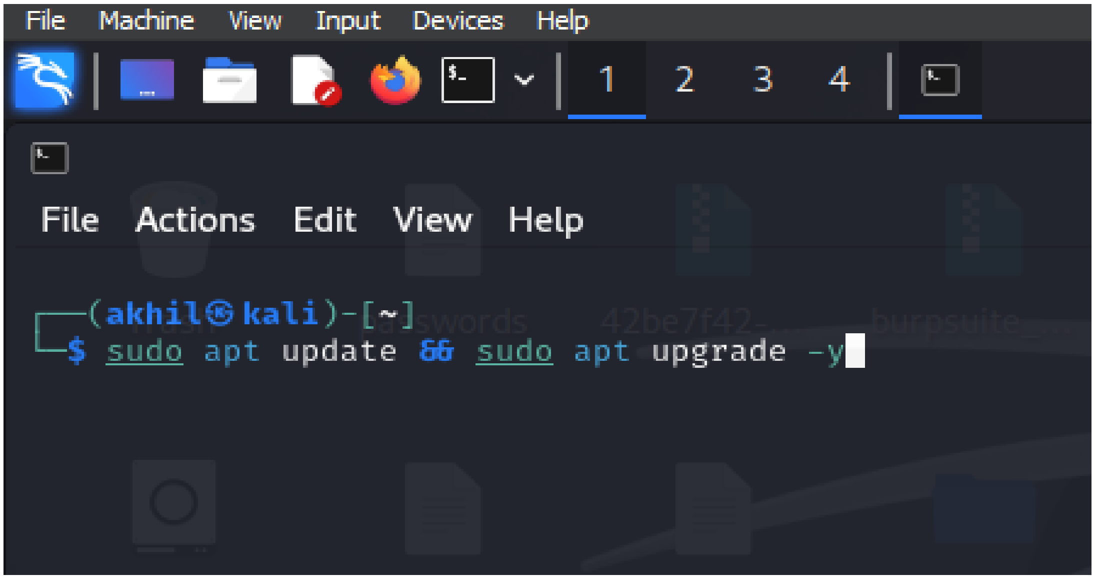
Pre-Installation Requirements

1. Operating System: Ubuntu 20.04 or CentOS 7.
2. Software: Snort 2.9.x or Snort 3.x.
3. Dependencies: `libpcap-dev`, `libpcre3-dev`, `libdumbnet-dev`, etc.
4. Tools for testing: Kali Linux (for attack simulation).
5. Network Setup: Virtual machines or physical devices for Snort and testing environments.

Installation Steps

Step 1: Prepare the Environment

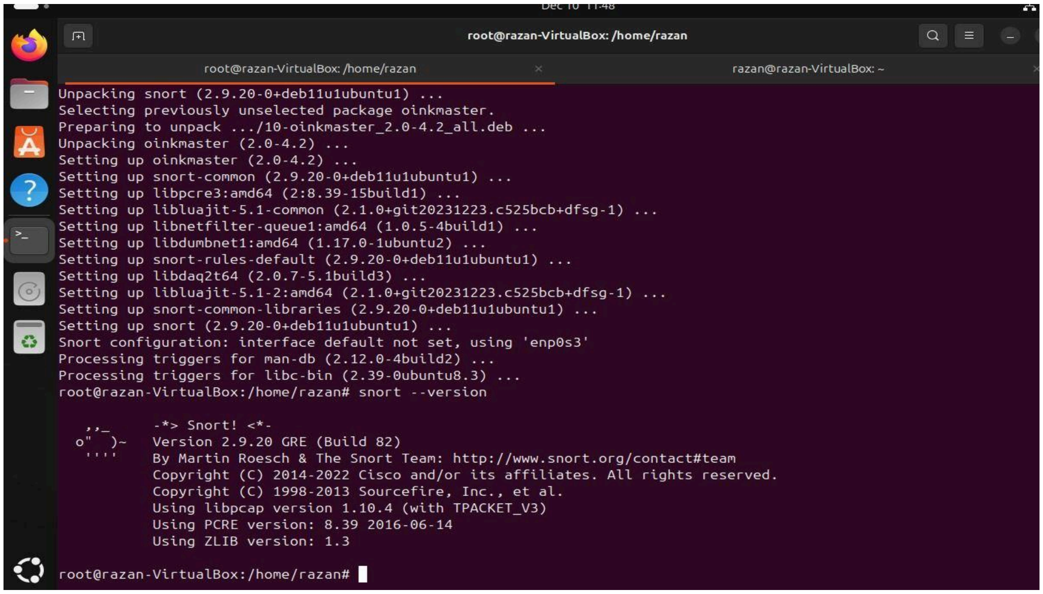
- Update the system and install dependencies.



```
┌──(akhil㉿kali)-[~]
└─$ sudo apt update && sudo apt upgrade -y
```

Step 2: Download and Install Snort

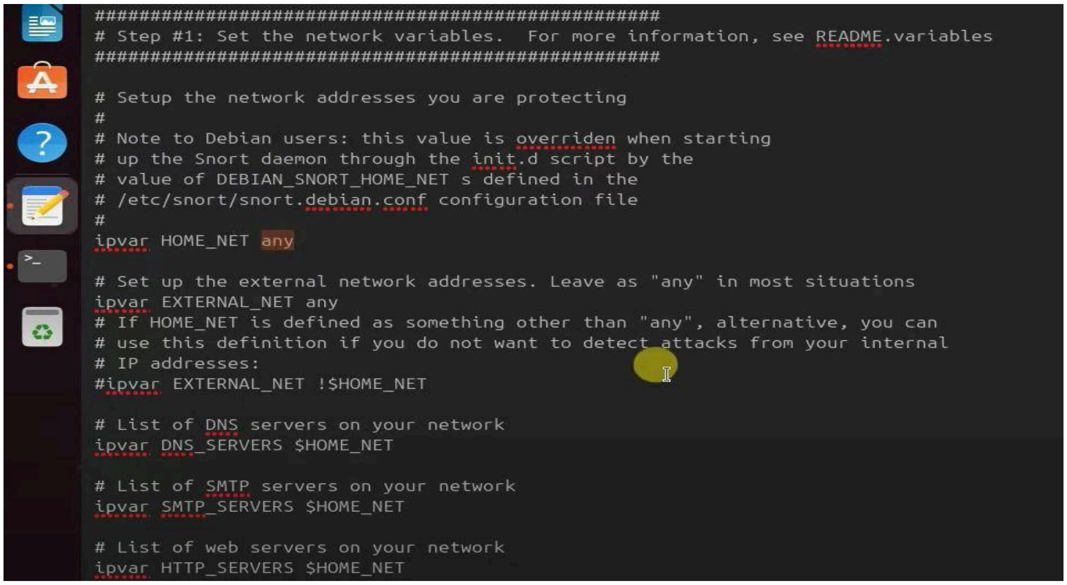
- Download the latest Snort source code.
- Compile and install Snort.


Dec 10 11:48
root@razan-VirtualBox: /home/razan
root@razan-VirtualBox: /home/razan
razan@razan-VirtualBox: ~

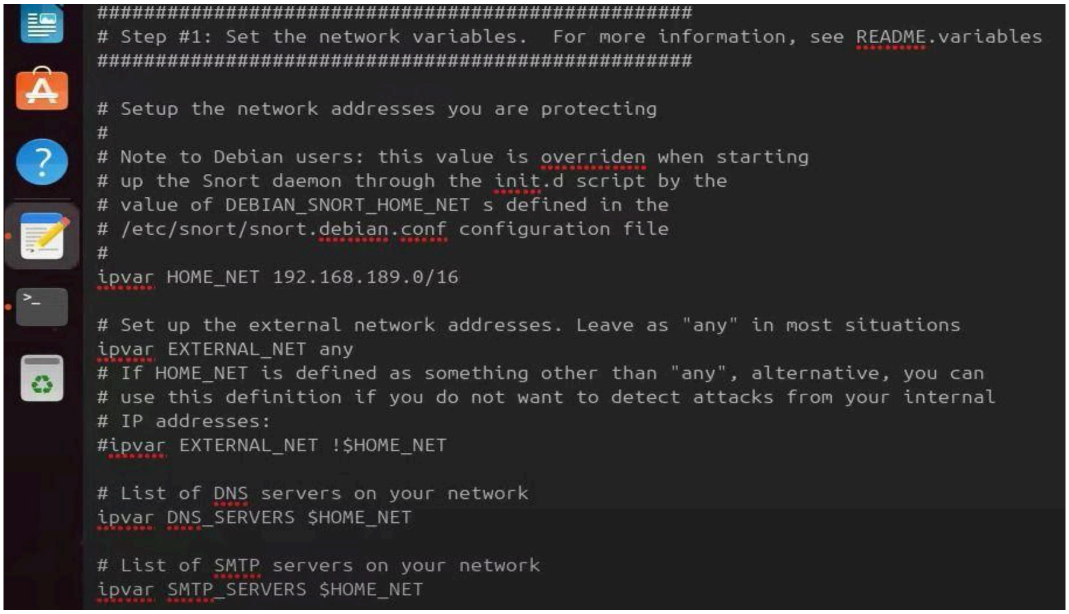
```
Unpacking snort (2.9.20-0+deb11u1ubuntu1) ...
Selecting previously unselected package oinkmaster.
Preparing to unpack .../10-oinkmaster_2.0-4.2_all.deb ...
Unpacking oinkmaster (2.0-4.2) ...
Setting up oinkmaster (2.0-4.2) ...
Setting up snort-common (2.9.20-0+deb11u1ubuntu1) ...
Setting up libpcre3:amd64 (2:8.39-15build1) ...
Setting up libluajit-5.1-common (2.1.0+git20231223.c525bcb+dfsg-1) ...
Setting up libnetfilter-queue1:amd64 (1.0.5-4build1) ...
Setting up libdumbnet1:amd64 (1.17.0-1ubuntu2) ...
Setting up snort-rules-default (2.9.20-0+deb11u1ubuntu1) ...
Setting up libdaq2t64 (2.0.7-5.1build3) ...
Setting up libluajit-5.1-2:amd64 (2.1.0+git20231223.c525bcb+dfsg-1) ...
Setting up snort-common-libraries (2.9.20-0+deb11u1ubuntu1) ...
Setting up snort (2.9.20-0+deb11u1ubuntu1) ...
Snort configuration: interface default not set, using 'enp0s3'
Processing triggers for man-db (2.12.0-4build2) ...
Processing triggers for libc-bin (2.39-0ubuntu8.3) ...
root@razan-VirtualBox:/home/razan# snort --version

   ,,_     -*> Snort! <*-
  o"  )~   Version 2.9.20 GRE (Build 82)
   ''''    By Martin Roesch & The Snort Team: http://www.snort.org/contact#team
           Copyright (C) 2014-2022 Cisco and/or its affiliates. All rights reserved.
           Copyright (C) 1998-2013 Sourcefire, Inc., et al.
           Using libpcap version 1.10.4 (with TPACKET_V3)
           Using PCRE version: 8.39 2016-06-14
           Using ZLIB version: 1.3

root@razan-VirtualBox:/home/razan# 
```

Step 3: Configure Snort

- Set up configuration files, define network variables, and specify logging paths.



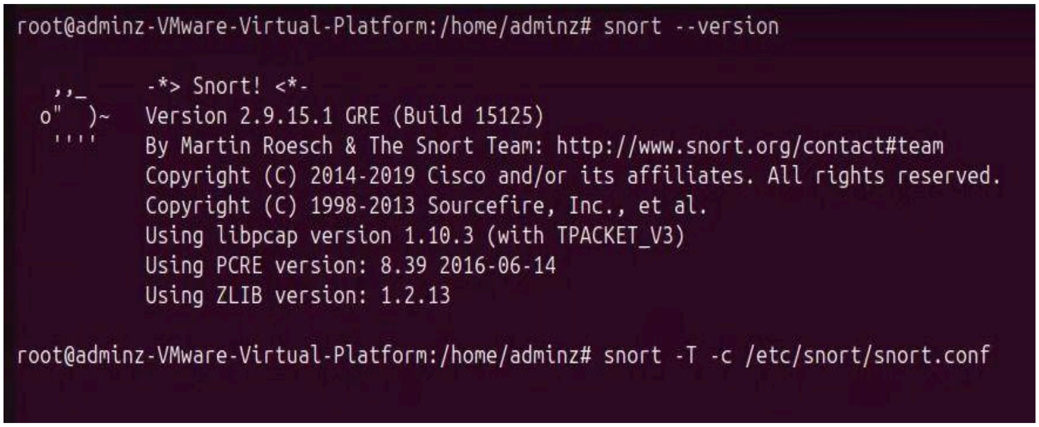
```
###################################################
# Step #1: Set the network variables.  For more information, see README.variables
###################################################

# Setup the network addresses you are protecting
#
# Note to Debian users: this value is overriden when starting
# up the Snort daemon through the init.d script by the
# value of DEBIAN_SNORT_HOME_NET s defined in the
# /etc/snort/snort.debian.conf configuration file
#
ipvar HOME_NET any

# Set up the external network addresses. Leave as "any" in most situations
ipvar EXTERNAL_NET any
# If HOME_NET is defined as something other than "any", alternative, you can
# use this definition if you do not want to detect attacks from your internal
# IP addresses:
#ipvar EXTERNAL_NET !$HOME_NET

# List of DNS servers on your network
ipvar DNS_SERVERS $HOME_NET

# List of SMTP servers on your network
ipvar SMTP_SERVERS $HOME_NET

# List of web servers on your network
ipvar HTTP_SERVERS $HOME_NET
```



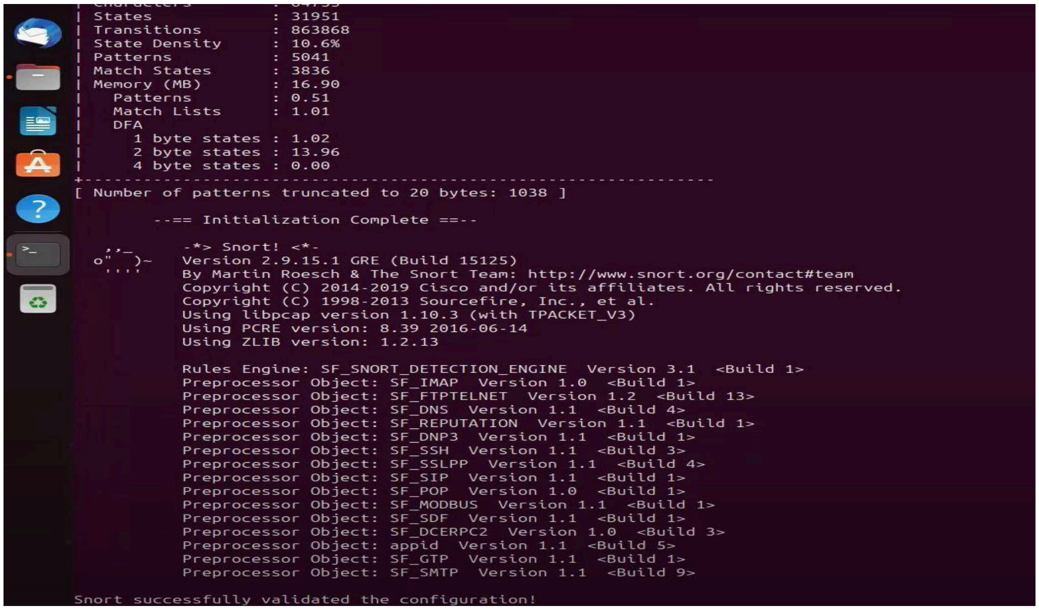
```
###################################################
# Step #1: Set the network variables.  For more information, see README.variables
###################################################

# Setup the network addresses you are protecting
#
# Note to Debian users: this value is overriden when starting
# up the Snort daemon through the init.d script by the
# value of DEBIAN_SNORT_HOME_NET s defined in the
# /etc/snort/snort.debian.conf configuration file
#
ipvar HOME_NET 192.168.189.0/16

# Set up the external network addresses. Leave as "any" in most situations
ipvar EXTERNAL_NET any
# If HOME_NET is defined as something other than "any", alternative, you can
# use this definition if you do not want to detect attacks from your internal
# IP addresses:
#ipvar EXTERNAL_NET !$HOME_NET

# List of DNS servers on your network
ipvar DNS_SERVERS $HOME_NET

# List of SMTP servers on your network
ipvar SMTP_SERVERS $HOME_NET
```

Step 4: Test the Configuration

- Validate the configuration using the `snort -T` command.



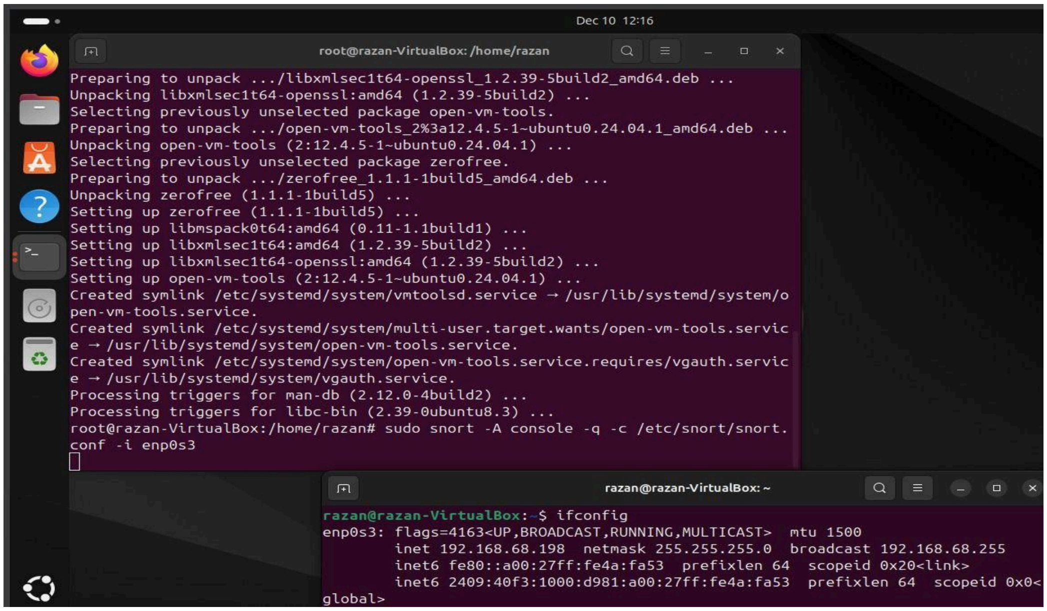
```
root@adminz-VMware-Virtual-Platform:/home/adminz# snort --version

   ,,_     -*> Snort! <*-
  o"  )~   Version 2.9.15.1 GRE (Build 15125)
   ''''    By Martin Roesch & The Snort Team: http://www.snort.org/contact#team
           Copyright (C) 2014-2019 Cisco and/or its affiliates. All rights reserved.
           Copyright (C) 1998-2013 Sourcefire, Inc., et al.
           Using libpcap version 1.10.3 (with TPACKET_V3)
           Using PCRE version: 8.39 2016-06-14
           Using ZLIB version: 1.2.13

root@adminz-VMware-Virtual-Platform:/home/adminz# snort -T -c /etc/snort/snort.conf
```



```
| States             : 31951
| Transitions        : 863868
| State Density      : 10.6%
| Patterns           : 5041
| Match States       : 3836
| Memory (MB)        : 16.90
|   Patterns         : 0.51
|   Match Lists      : 1.01
|   DFA
|     1 byte states : 1.02
|     2 byte states : 13.96
|     4 byte states : 0.00
+-----------------------------------------------------------------
[ Number of patterns truncated to 20 bytes: 1038 ]

        --== Initialization Complete ==--

   ,,_     -*> Snort! <*-
  o"  )~   Version 2.9.15.1 GRE (Build 15125)
   ''''    By Martin Roesch & The Snort Team: http://www.snort.org/contact#team
           Copyright (C) 2014-2019 Cisco and/or its affiliates. All rights reserved.
           Copyright (C) 1998-2013 Sourcefire, Inc., et al.
           Using libpcap version 1.10.3 (with TPACKET_V3)
           Using PCRE version: 8.39 2016-06-14
           Using ZLIB version: 1.2.13

           Rules Engine: SF_SNORT_DETECTION_ENGINE  Version 3.1  <Build 1>
           Preprocessor Object: SF_IMAP  Version 1.0  <Build 1>
           Preprocessor Object: SF_FTPTELNET  Version 1.2  <Build 13>
           Preprocessor Object: SF_DNS  Version 1.1  <Build 4>
           Preprocessor Object: SF_REPUTATION  Version 1.1  <Build 1>
           Preprocessor Object: SF_DNP3  Version 1.1  <Build 1>
           Preprocessor Object: SF_SSH  Version 1.1  <Build 3>
           Preprocessor Object: SF_SSLPP  Version 1.1  <Build 4>
           Preprocessor Object: SF_SIP  Version 1.1  <Build 1>
           Preprocessor Object: SF_POP  Version 1.0  <Build 1>
           Preprocessor Object: SF_MODBUS  Version 1.1  <Build 1>
           Preprocessor Object: SF_SDF  Version 1.1  <Build 1>
           Preprocessor Object: SF_DCERPC2  Version 1.0  <Build 3>
           Preprocessor Object: appid  Version 1.1  <Build 5>
           Preprocessor Object: SF_GTP  Version 1.1  <Build 1>
           Preprocessor Object: SF_SMTP  Version 1.1  <Build 9>

Snort successfully validated the configuration!
```

Configuring Snort Rules

1. Default Rules: Utilize Snort’s default rule sets.
2. Custom Rules: Write rules tailored to detect specific threats.
3. Community Rules: Integrate freely available community rules for enhanced functionality.



```
Preparing to unpack .../libxmlsec1t64-openssl_1.2.39-5build2_amd64.deb ...
Unpacking libxmlsec1t64-openssl:amd64 (1.2.39-5build2) ...
Selecting previously unselected package open-vm-tools.
Preparing to unpack .../open-vm-tools_2%3a12.4.5-1~ubuntu0.24.04.1_amd64.deb ...
Unpacking open-vm-tools (2:12.4.5-1~ubuntu0.24.04.1) ...
Selecting previously unselected package zerofree.
Preparing to unpack .../zerofree_1.1.1-1build5_amd64.deb ...
Unpacking zerofree (1.1.1-1build5) ...
Setting up zerofree (1.1.1-1build5) ...
Setting up libmspack0t64:amd64 (0.11-1.1build1) ...
Setting up libxmlsec1t64:amd64 (1.2.39-5build2) ...
Setting up libxmlsec1t64-openssl:amd64 (1.2.39-5build2) ...
Setting up open-vm-tools (2:12.4.5-1~ubuntu0.24.04.1) ...
Created symlink /etc/systemd/system/vmtoolsd.service → /usr/lib/systemd/system/o
pen-vm-tools.service.
Created symlink /etc/systemd/system/multi-user.target.wants/open-vm-tools.servic
e → /usr/lib/systemd/system/open-vm-tools.service.
Created symlink /etc/systemd/system/open-vm-tools.service.requires/vgauth.servic
e → /usr/lib/systemd/system/vgauth.service.
Processing triggers for man-db (2.12.0-4build2) ...
Processing triggers for libc-bin (2.39-0ubuntu8.3) ...
root@razan-VirtualBox:/home/razan# sudo snort -A console -q -c /etc/snort/snort.
conf -i enp0s3
```

```
razan@razan-VirtualBox:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.68.198  netmask 255.255.255.0  broadcast 192.168.68.255
        inet6 fe80::a00:27ff:fe4a:fa53  prefixlen 64  scopeid 0x20<link>
        inet6 2409:40f3:1000:d981:a00:27ff:fe4a:fa53  prefixlen 64  scopeid 0x0<
global>
```

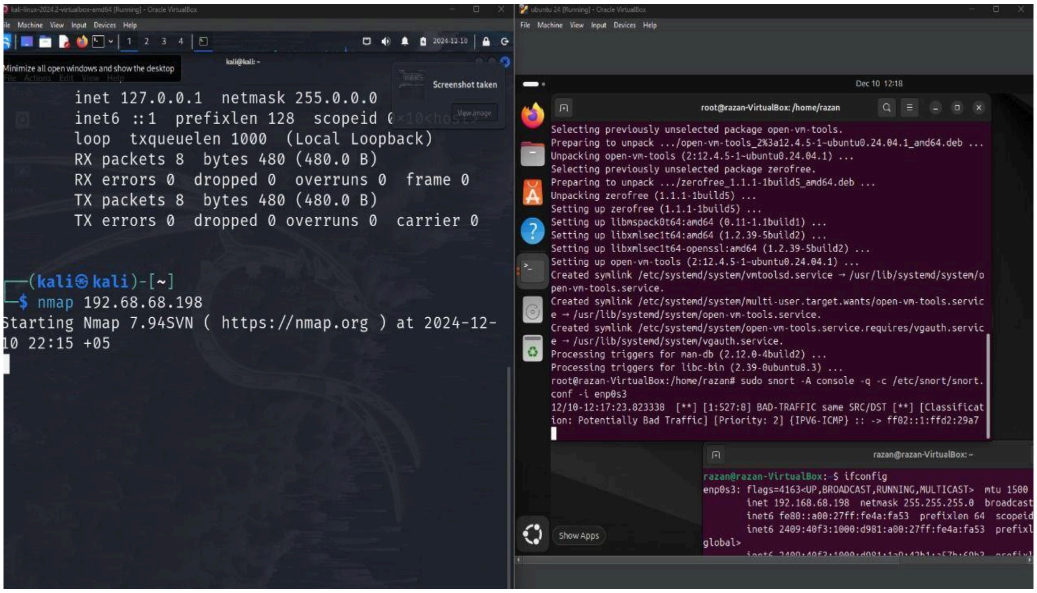
Testing Snort for IDS/IPS

IDS Mode:

- Capture live traffic using Snort in IDS mode.
- Test with benign and malicious traffic to verify alerts.

IPS Mode:

- Use Snort with iptables to drop malicious packets.
- Simulate attacks like SQL injection or DoS and monitor Snort's responses.



kali-linux-2024.2-virtualbox-amd64 [Running] - Oracle VirtualBox
File Machine View Input Devices Help
2024-12-10
Minimize all open windows and show the desktop
kali@kali: ~
Screenshot taken
View image
ubuntu 24 [Running] - Oracle VirtualBox
File Machine View Input Devices Help
Dec 10 12:18
root@razan-VirtualBox: /home/razan
razan@razan-VirtualBox: ~
Show Apps

```
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 8  bytes 480 (480.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 8  bytes 480 (480.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0


┌──(kali㉿kali)-[~]
└─$ nmap 192.68.68.198
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-12-
10 22:15 +05
```

```
Selecting previously unselected package open-vm-tools.
Preparing to unpack .../open-vm-tools_2%3a12.4.5-1-ubuntu0.24.04.1_amd64.deb ...
Unpacking open-vm-tools (2:12.4.5-1-ubuntu0.24.04.1) ...
Selecting previously unselected package zerofree.
Preparing to unpack .../zerofree_1.1.1-1build5_amd64.deb ...
Unpacking zerofree (1.1.1-1build5) ...
Setting up zerofree (1.1.1-1build5) ...
Setting up libmspack0t64:amd64 (0.11-1.1build1) ...
Setting up libxmlsec1t64:amd64 (1.2.39-5build2) ...
Setting up libxmlsec1t64-openssl:amd64 (1.2.39-5build2) ...
Setting up open-vm-tools (2:12.4.5-1-ubuntu0.24.04.1) ...
Created symlink /etc/systemd/system/vmtoolsd.service → /usr/lib/systemd/system/o
pen-vm-tools.service.
Created symlink /etc/systemd/system/multi-user.target.wants/open-vm-tools.servic
e → /usr/lib/systemd/system/open-vm-tools.service.
Created symlink /etc/systemd/system/open-vm-tools.service.requires/vgauth.servic
e → /usr/lib/systemd/system/vgauth.service.
Processing triggers for man-db (2.12.0-4build2) ...
Processing triggers for libc-bin (2.39-0ubuntu8.3) ...
root@razan-VirtualBox:/home/razan# sudo snort -A console -q -c /etc/snort/snort.
conf -i enp0s3
12/10-12:17:23.823338  [**] [1:527:8] BAD-TRAFFIC same SRC/DST [**] [Classificat
ion: Potentially Bad Traffic] [Priority: 2] {IPV6-ICMP} :: -> ff02::1:ffd2:29a7
```

```
razan@razan-VirtualBox:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.68.198  netmask 255.255.255.0  broadcast
        inet6 fe80::a00:27ff:fe4a:fa53  prefixlen 64  scopeid
        inet6 2409:40f3:1000:d981:a00:27ff:fe4a:fa53  prefixl
global>
```

Challenges During Installation and Configuration

1. Dependency issues during installation.
2. Misconfigurations in rules causing false positives or negatives.
3. Performance optimization for high traffic volumes.

Conclusion

This project successfully demonstrates the deployment and configuration of Snort as a robust and reliable Intrusion Detection and Prevention System (IDS/IPS). Through meticulous setup and practical testing, we validated Snort's capability to detect, analyze, and mitigate various network threats in real-time. By simulating real-world attack scenarios, such as Distributed Denial of Service (DDoS) attacks and unauthorized access attempts, we effectively showcased Snort’s ability to safeguard network infrastructures against malicious activities.

The project highlights the importance of proactive threat detection and prevention in modern cybersecurity frameworks. By leveraging Snort's extensive packet analysis and customizable rule-based detection, we emphasized its versatility and adaptability to different environments. Furthermore, the challenges encountered during installation and configuration, such as optimizing rules to reduce false positives and enhancing performance in high-traffic scenarios, provided valuable learning opportunities and practical insights into the nuances of IDS/IPS systems.

The findings of this project underline the critical role of IDS/IPS tools like Snort in enhancing network security. They also offer a foundation for further exploration, including the integration of Snort with advanced security tools and platforms to create more comprehensive monitoring solutions. The knowledge gained from this project serves as a valuable resource for IT professionals, network administrators, and cybersecurity enthusiasts, enabling them to strengthen their defense mechanisms against evolving cyber threats.

In conclusion, the successful implementation of Snort in this project demonstrates its effectiveness as a key component of any robust cybersecurity strategy. As cyber threats continue to grow in complexity and frequency, the insights from this project will guide future security initiatives, fostering a safer and more secure digital environment.

Bibliography

1. Snort User Manual.
2. Official Snort Documentation.
3. Network Security Research Papers.
4. Online Tutorials and Resources.