Intrusion Prevention Detection

Rúben Correia Pereira
up202006195

Rui Pedro Borges Silva
up202005661

Simão Queirós Rodrigues
up202005700

I. Introduction

Enhancing systems for detecting unauthorized entries (IDS) is a pivotal aspect of digital security, aimed at fortifying online barriers against evolving cyber threats. Fine-tuning these systems, akin to the precision needed in drafting scientific manuscripts, is essential for ensuring that IDS operate within predefined limits, thus maximizing their efficacy in identifying potential threats.

These prerequisites, encompassing methods like signature recognition and algorithms for spotting irregularities, mirror the deliberate design choices vital for warding off emerging 'zero-day' vulnerabilities. The strategic anticipation and preset confines built into IDS mirror the meticulous composition of academic papers, aiming not just for distinction but also for seamless assimilation within the broader digital security landscape.

Maintaining the integrity of IDS standards is paramount for their effectiveness. Any deviation, much like altering the rules of formatting in scholarly writing, could potentially compromise the entire system. This concept intertwines with the need for continuous upgrades in cyber security and intrusion prevention systems, underscoring the importance of both meticulous design and proactive adaptation in the face of new challenges.

II. Environment Configuration

A. Environment Description

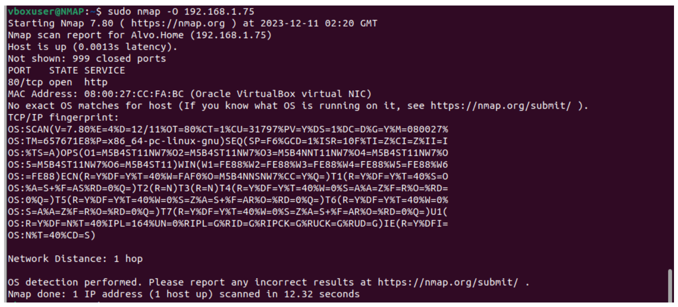
This project involves setting up a network environment consisting of three virtual machines (VMs), all running on Ubuntu 20.04 operating system. The network is configured in Bridge Adapted mode, allowing each VM to function as if it were directly connected to the physical network.

B. Target Machine Configuration

The target machine, one of the three VMs, does not receive specific configurations in this scenario. Its primary role is to be the recipient of monitoring and attacking actions in test activities.

C. Attacker Machine Configuration

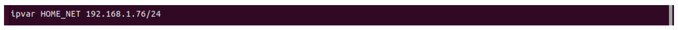
The attacker machine is set up with the necessary tools for penetration testing. After updating and upgrading the system with the commands **sudo apt-get update** and **sudo apt-get upgrade**, we install Nmap, a network exploration and security auditing tool, with **sudo apt-get install nmap**. Once



```
vboxuser@NMAP:~$ sudo nmap -O 192.168.1.75
Starting Nmap 7.80 ( https://nmap.org ) at 2023-12-11 02:20 GMT
Nmap scan report for Alvo.Home (192.168.1.75)
Host is up (0.0013s latency).
Not shown: 999 closed ports
PORT   STATE SERVICE
80/tcp open  http
MAC Address: 08:00:27:CC:FA:BC (Oracle VirtualBox virtual NIC)
No exact OS matches for host (If you know what OS is running on it, see https://nmap.org/submit/ ).
TCP/IP fingerprint:
OS:SCAN(V=7.80%E=4%D=12/11%OT=80%CT=1%CU=31797%PV=Y%DS=1%DC=D%G=Y%M=080027%
OS:TM=657671E8%P=x86_64-pc-linux-gnu)SEQ(SP=F6%GCD=1%ISR=10F%TI=Z%CI=Z%II=I
OS:%TS=A)OPS(O1=M5B4ST11NW7%O2=M5B4ST11NW7%O3=M5B4NNT11NW7%O4=M5B4ST11NW7%O
OS:5=M5B4ST11NW7%O6=M5B4ST11)WIN(W1=FE88%W2=FE88%W3=FE88%W4=FE88%W5=FE88%W6
OS:=FE88)ECN(R=Y%DF=Y%T=40%W=FAF0%O=M5B4NNSNW7%CC=Y%Q=)T1(R=Y%DF=Y%T=40%S=O
OS:%A=S+%F=AS%RD=0%Q=)T2(R=N)T3(R=N)T4(R=Y%DF=Y%T=40%W=0%S=A%A=Z%F=R%O=%RD=
OS:0%Q=)T5(R=Y%DF=Y%T=40%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)T6(R=Y%DF=Y%T=40%W=0%
OS:S=A%A=Z%F=R%O=%RD=0%Q=)T7(R=Y%DF=Y%T=40%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)U1(
OS:R=Y%DF=N%T=40%IPL=164%UN=0%RIPL=G%RID=G%RIPCK=G%RUCK=G%RUD=G)IE(R=Y%DFI=
OS:N%T=40%CD=S)

Network Distance: 1 hop

OS detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 12.32 seconds
```



```
ipvar HOME_NET 192.168.1.76/24
```

installed, Nmap is used to scan the target machine using the command **sudo nmap -O [target_machine_ip]**, where **[target_machine_ip]** is the IP address of the target machine.

D. Snort Machine Configuration

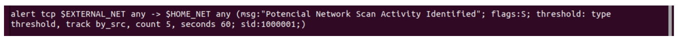
The machine with Snort, an intrusion detection and prevention system, is configured to monitor the network. First, we enable promiscuous mode on the network interface with **sudo ip link set enp0s3 promisc on**. After system updates, we install Snort with **sudo apt-get install snort**. We configure Snort to monitor the network by setting **ipvar HOME_NET 192.168.1.76/24** in the configuration file **/etc/snort/snort.conf**, where **192.168.1.76** is the IP address of the Snort machine.

We add the detection rule **alert tcp $EXTERNAL_NET any -¿ $HOME_NET any (msg:"Potential Network Scan Activity Identified"; FLAGS:S; threshold: type threshold, track by_src, count 5, seconds 60; sid:1000001;)** in the file **/etc/snort/rules/local.rules**.

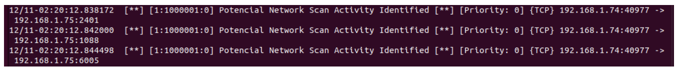
Finally, we activate Snort with **sudo snort -d -l /var/log/snort/ -A console -c /etc/snort/snort.conf**, where the options configure the display of application layer data, log directory, screen alerts, and configuration file, respectively.

III. Characterization of the Intrusion

In the security evaluation process, the Nmap tool was deployed using the **sudo nmap -O [target_machine_ip]** command to initiate a scan aimed at identifying the operating system of the specified host. This technique involves sending a series of TCP and UDP probes to the target and analyzing the



```
alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"Potencial Network Scan Activity Identified"; flags:S; threshold: type
threshold, track by_src, count 5, seconds 60; sid:1000001;)
```



```
12/11-02:20:12.838172  [**] [1:1000001:0] Potencial Network Scan Activity Identified [**] [Priority: 0] {TCP} 192.168.1.74:40977 ->
 192.168.1.75:2401
12/11-02:20:12.842000  [**] [1:1000001:0] Potencial Network Scan Activity Identified [**] [Priority: 0] {TCP} 192.168.1.74:40977 ->
 192.168.1.75:1088
12/11-02:20:12.844498  [**] [1:1000001:0] Potencial Network Scan Activity Identified [**] [Priority: 0] {TCP} 192.168.1.74:40977 ->
 192.168.1.75:6005
```

received responses. The goal is to compare these responses against a database of known OS signatures to infer the operating system in use. However, the scan results did not conclusively determine the operating system, suggesting the potential use of an atypical configuration or a system not yet cataloged in the existing database, or that the data collected was not sufficient for a definitive identification.

IV. Snort's Reaction

In response to potential network scanning activities, the Snort intrusion detection system was deployed. It was configured to reference a predefined rule that detects scanning activities indicated by a sequence of SYN packets—a hallmark of scanning techniques. This rule is designed to reduce false positives by setting a threshold that requires a specific number of such packets within a particular time frame before triggering an alert. The presence of such network traffic prompts Snort to issue an alert, signaling to network security administrators the need for further analysis and potential action to address any security concerns identified by the alert.

V. Intrusion Prevention System Enhancement

The cybersecurity landscape is in a state of perpetual transformation, with threats becoming increasingly sophisticated, including 'zero-day' exploits aimed at previously unidentified vulnerabilities. It is vital, in the context of Intrusion Prevention System (IPS) enhancement, to account for the dynamic nature of these threats.

A. Behavioral Detection Integration

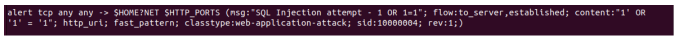
The implementation of behavior-based detection methodologies is pivotal. This involves the integration of artificial intelligence, machine learning, and sophisticated algorithms aimed at identifying anomalous behavior and suspicious network patterns, thereby transcending the capabilities of traditional signature-based detection.

B. Heuristic Analysis and Big Data Utilization

Advancements in heuristic analysis coupled with Big Data capabilities facilitate the recognition of irregular activities. This is crucial for identifying potential threats in scenarios where no existing signatures are present, which may be indicative of an active cyberattack.

C. Snort Machine Configuration

IPS systems require instantaneous updates and modifications to combat newly emerging threats. Leveraging insights from historical attack patterns is essential for refining defensive measures against future threats.



```
alert tcp any any -> $HOME?NET $HTTP_PORTS (msg:"SQL Injection attempt - 1 OR 1=1"; flow:to_server,established; content:"1' OR
'1' = '1"; http_uri; fast_pattern; classtype:web-application-attack; sid:10000004; rev:1;)
```

VI. SQL Injection Attempt

This report outlines the steps for updating system packages, installing web development components, configuring MySQL, installing DVWA, setting up PHP, and enhancing security against SQL injection attacks.

A. System Package Update and Installation

Initially, the system packages are updated using the commands **sudo apt-get update and sudo apt-get upgrade**. This is followed by the installation of Apache, MySQL, and PHP through a series of **sudo apt-get install** commands for apache2, mysql-server, php, php-mysqli, php-gd, and libapache2-mod-php.

B. MySQL Configuration

The second step involves configuring MySQL. This includes running a security script to set the MySQL root password **sudo mysql_secure_installation** and creating a DVWA database and user within MySQL. The commands for this step include logging into MySQL as root, creating the DVWA database, creating a user for DVWA, granting all privileges to this user on the DVWA database, and flushing the privileges.

C. DVWA Installation

Step three is the installation of DVWA. This involves navigating to the **/var/www/html** directory, downloading DVWA from GitHub, changing permissions of the DVWA directory, copying and configuring the DVWA configuration file. Specific configuration changes are required in **config.inc.php**, including setting the database user, password, and database name.

D. PHP Configuration

The fourth step is the configuration of PHP for DVWA. It requires editing the **php.ini** file, enabling **allow_url_include and display_errors** settings, and restarting Apache to apply these changes.

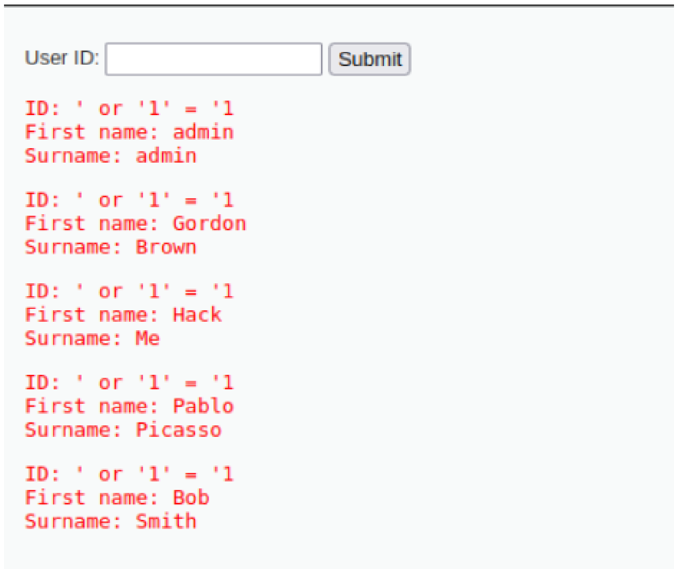
E. Accessing DVWA

The final step, step five, details accessing DVWA through a browser using the URL **http://target_machine_ip/DVWA** and logging in with the default credentials (admin/password).

F. Security Enhancement

Additionally, to enhance security against SQL injection attacks, particularly the "1 OR 1=1" attack, a rule is added to the **local.rules** file. The rule is: **alert tcp any any -¿ $HOME_NET $HTTP_PORTS (msg:"SQL Injection attempt - 1 OR 1=1"; flow:to_server,established; content:"1' OR '1' = '1"; http_uri; fast_pattern; classtype:web-application-attack; sid:1000004; rev:1;)**.

This rule is designed to detect and alert any attempt of SQL injection where the attacker tries to exploit the database by using the **1 OR 1=1** condition, a common technique in SQL injection attacks.



```
ID: ' or '1' = '1
First name: admin
Surname: admin

ID: ' or '1' = '1
First name: Gordon
Surname: Brown

ID: ' or '1' = '1
First name: Hack
Surname: Me

ID: ' or '1' = '1
First name: Pablo
Surname: Picasso

ID: ' or '1' = '1
First name: Bob
Surname: Smith
```

VII. Conclusion

This document has meticulously examined critical aspects of cybersecurity, particularly emphasizing the importance of intrusion detection and prevention mechanisms like Snort within network infrastructures. It explored the complex interplay between aggressive cyber tactics and the strategic responses deployed by Snort, highlighted during a simulated network breach. The essential preparatory steps—configuring virtual machines, deploying Snort, and crafting specific rules in the **local.rules** file—establish the groundwork for identifying and mitigating hidden network probes.

Snort's response to the simulated intrusion demonstrates the strength of these security protocols, evident in the triggered alerts and the blocking of unauthorized access. This performance reaffirms the vital role of robust intrusion detection systems in recognizing and proactively responding to potential cyber threats.

The practical exercises carried out have offered valuable insights into network security dynamics, underscoring the effectiveness of advanced defensive measures. By simulating attack scenarios and observing the response mechanisms, we have gained a deeper understanding of the nature of cyber attacks and their counteractions.

Furthermore, this report has detailed the steps for system package updates, web development component installation, MySQL configuration, DVWA installation, PHP setup, and the enhancement of security against SQL injection attacks. Each phase, from updating system packages with **sudo apt-get update** and **sudo apt-get upgrade** to implementing intricate configurations in MySQL and DVWA, has been critical. The inclusion of a specific rule in the **local.rules** file to counter the ”1 OR 1=1” SQL injection attack is a testament to the comprehensive approach towards securing network environments.

In conclusion, the findings stress a continuous commitment to advancing cybersecurity measures and strategically developing intrusion detection systems. The report underscores the necessity of strategic foresight and proactive defense strategies, ensuring the ongoing protection of network environments against the ever-evolving landscape of cyber threats.