# **ETHICAL HACKING**

HAN22080031

HAN22080204

HAN22080221

HAN22080120

HAN22080144

# I) Introduction

Company named DigiTech just be attacked by an anonymous hacker. During the investigation, we found unusually activity from Mr. S account. The following report describe attack techniques in each phase and proposing solution as the defensive side to prevent similar threat in the future.

# II)  Phase 1: Reconnaissance

## 1. What happened?

The attacker utilised the OSINT framework to gather information of the company from free sources. From the company website, company emails [Mr.s@definatelyarealcompany.com](mailto:Mr.s@definatelyarealcompany.com) and [notascammer@definatelyarealcompany.com](mailto:notascammer@definatelyarealcompany.com) were found. The email domains were scanned using public tools and were found to not have anti-spoofing mechanisms.

A screenshot of a computer

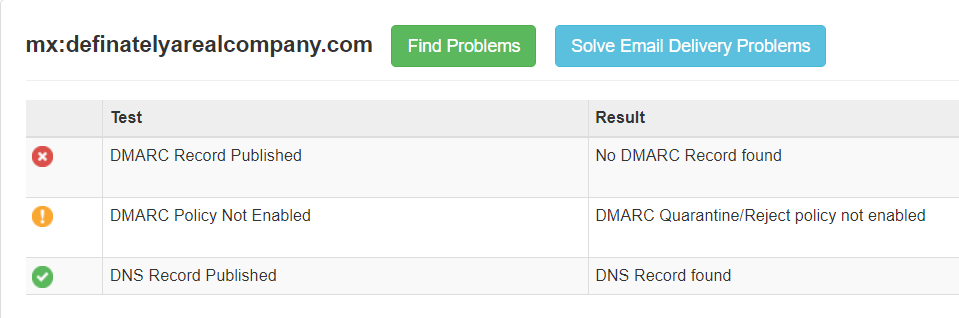
Description automatically generated

Figure 2.1.0.1&2.1.0.2 Mail scan result from [https://mxtoolbox.com](https://mxtoolbox.com/) and <https://checkcybersecurity.service.ncsc.gov.uk/email-security-check>.

The attacker then attempted to forge a phishing mail to **Mr. S** pretending to be **notascammer** using Gophish **(appendix 1)**.

A screenshot of a computer

Description automatically generated

Figure 2.1.0.3 Phishing email in Mr. S’s outlook.

A screenshot of a computer code

Description automatically generatedA screenshot of a computer

Description automatically generated

Figure 2.1.0.4&2.1.0.5 phishing mail header and company hmailserver unsecure configuration.

Upon inspecting the mail headers, the group noticed that the mail server was unsecured and did not require authentication on email headers, allowing unauthenticated and malicious emails as such being delivered.

## 2. Email secure protocols SPF, DKIM, DMARC

This part will cover process of securing company email domains for both self-hosted email servers and third-party email services, in this case hmailserver and google workplace.

### 2.1 Sender Policy Framework (SPF)

SPF prevents spoofing by specifying the mail servers allowed to send email using the company domain. SPF publishes a list authorized IP addresses in a DNS TXT record, the server validates the source IP address of email address using the domain based on the list. If email is not sent from an authorized source, it will be marked as spam or will be rejected.

A screenshot of a computer

Description automatically generatedFigure 2.2.1.1SPF graph **​**(Mantra, 2024)**​**.

Adding SPF to domain requires creating a DNS TXT record on your domain. Since the mail server is hosted at 192.168.1.7, the following SPF record is configured so only this server can send mail from the company domain (definatelyarealcompany.com).

A close-up of a computer code

Description automatically generated

Figure 2.2.1.2 Mail server SPF configuration.

Similarly, google workplace emails uses google SPF record.

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

Figure 2.2.1.3 Google workplace SPF configuration.

### *2.2* DomainKeys Identified Mail *(DKIM)*

DKIM ensures email integrity and authenticity by using a private key as a digital signature. When there is an email, the server fetches the public key to verify the signature. Emails failed to verify their signature will be quarantined or rejected.

A diagram of a server

Description automatically generated

Figure 2.2.2.1 DKIM graph ​(Nicolas, 2024)​.

A screen shot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

A black text on a white background

Description automatically generated

Figure 2.2.2.2&2.2.2.3&2.2.2.3 RSA key generation and mail server DKIM configuration.

A DKIM record usually contains the version of DKIM, encryption algorithm and public key. Adding DKIM to domain requires a pair of public and private key and creating a DNS TXT record. For local server, a pair of keys can be generated using OpenSSL. The server will hold the private key while the DNS TXT record uses the public key.

On google workplace, keys are generated from the admin console and DNS TXT record “google.\_domainkey” must be created. A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

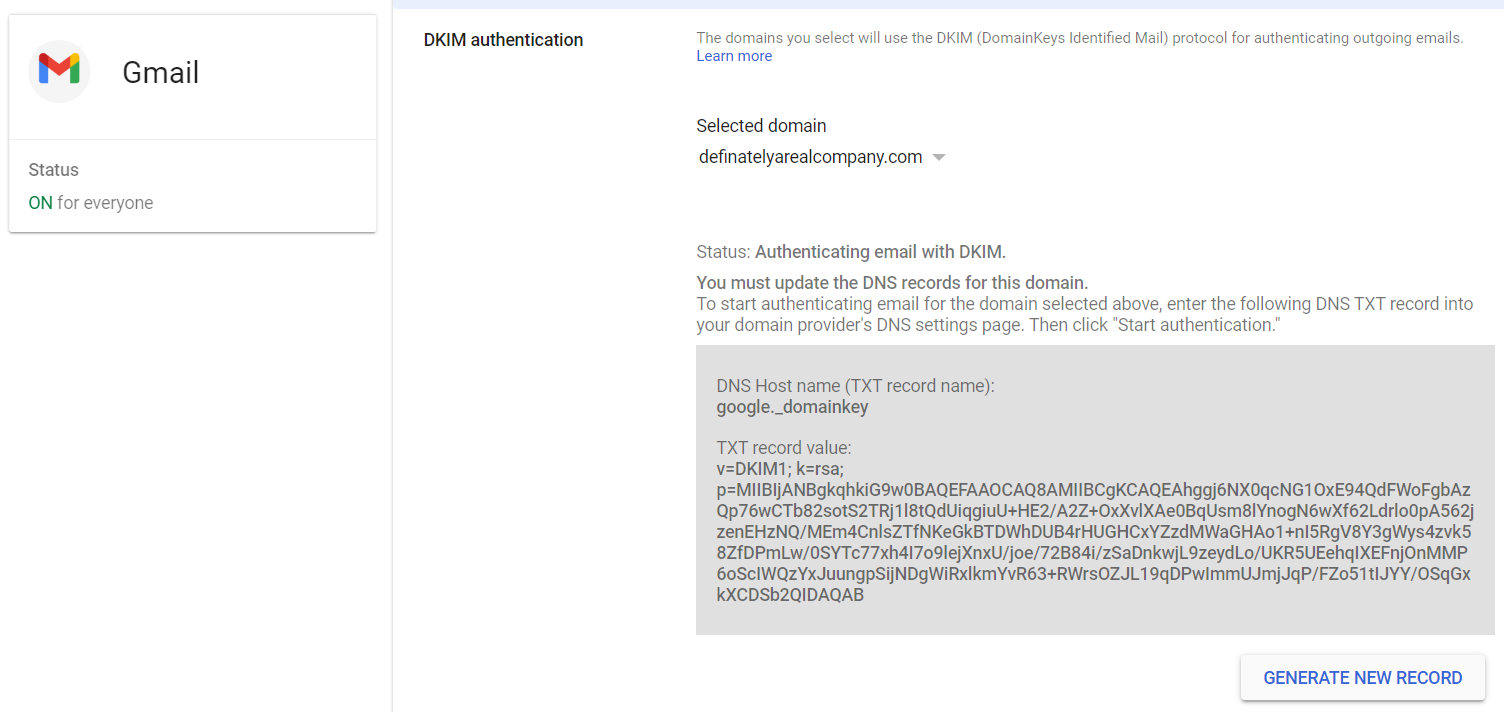


Figure 2.2.2.4&2.2.2.5 Google workplace RSA keys generation and DKIM configuration.

### *2.3* Domain-based Message Authentication, Reporting, and Conformance *(DMARC)*

DMARC provides a unified policy to handles emails that fail SPF or DKIM checks and report them to domain owner. DMARC also ensures the email “FROM” header aligns with the domain in SPF and DKIM checks, providing additional spoofing protection.

A computer screen shot of a computer

Description automatically generated

*Figure* 2.2.3.6DMARC graph ​(Mantra, 2024)​.

A DMARC record usually contains the version of DMARC, policy to handle email failing SPF or DKIM and an email to send report to. Adding DMARC requires a DNS TXT record. 

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

*Figure* 2.2.3.7&2.2.3.8 Google workplace and mail server DMARC configuration.

### 2.4 Testing

A computer screen shot of a computer code

Description automatically generatedA screenshot of a computer

Description automatically generated

*Figure* 2.2.4.1&2.2.4.2Email domain scan and email headers.

Now, the company email domain will be scanned with a test email. Since, email security protocols are running, DMARC should trigger if the attacker tries to spoof the email domain.

A screenshot of a email

Description automatically generated

A screen shot of a computer

Description automatically generated

*Figure* 2.2.4.3&2.2.4.4 DMARC prevents email spoofing.

## 3. Countermeasures against email phishing

To raise awareness against social engineering and spoofing emails, conduct regular training sessions on identifying suspicious emails, implement phishing simulations to test and educate employees. Make a policy requiring employees to verify sender addresses for every email and avoid clicking suspicious links. Furthermore, research also shows an automated solution to detect deception. (Abd, et al., 2021)

# III) Phase 2: Scanning

## 1) Introduction

The second phase is scanning, where hackers examine open and closed ports to identify a potential entry point into the system. The hacker attempts to create a comprehensive overview of the target network including IP addresses or running services on systems.

A diagram of a diagram

Description automatically generated*Figure 3.1.0.1*: attacker trying to scan inside network

## 2) Firewall

To block access from an attacker, the defender must build and configure a firewall to stop access from the attacker. Here our team uses the IPTABLES as a method to block Nmap and ping or access from the attacker.

A screenshot of a computer program

Description automatically generated

*Figure 3.2.0.1*

In both ‘chain INPUT’ and ‘FORWARD’, rules number 1 to 4 are for blocking port scan. Any attempt to conduct a port scan on our system will be automatically banned for a full 24-hour period. After the day has ended, delete them from the ports can list. Subsequently, these regulations include scanners in the port scan list and record the attempted action. For the rest rules of ‘chain INPUT’ will block TCP attacks which include SYN-FIN, SYN-RST, X-Mas, NULLflags, ALLflags attacks and nmap FIN scan.

A screenshot of a computer screen

Description automatically generated

*Figure 3.2.0.2*

As an example, Nmap shows no-response means Nmap by an attacker has been blocked. On another example, when attack conduct specific port, it will show as filtered port (meaning a firewall or filter is blocking the port so that Nmap cannot tell whether it is open or closed):

A computer screen with white text

Description automatically generated

*Figure 3.2.0.3*

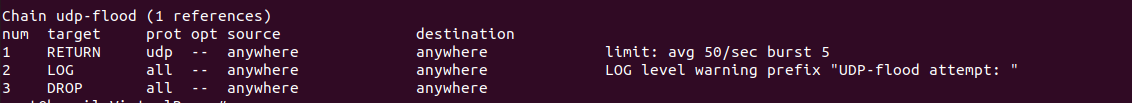
Beside that it can also block ping from the attacker:

A screenshot of a computer screen

Description automatically generated

*Figure 3.2.0.4*

We also set some UDP rules to prevent UDP flood attacks:



*Figure 3.2.0.5*

## 3) Snort

However, in certain cases, the attacker tries to bypass the firewall by fragmenting the Nmap packet. Therefore, we employ a network intrusion detection system known as Snort to prevent it. Snort employs a set of rules to identify and classify dangerous network activity (Snort.org, 2020). It then scans packets for matches against these rules and provides alerts for users. After installing and configuring Snort, we use a command to active Snort for recording attack and access from attackers:

A screenshot of a computer

Description automatically generated

*Figure 3.3.0.1*

In this situation, we use kali Linux as an attacker and try to do fragment Nmap to port 22 of ubuntu user:

A screenshot of a computer program

Description automatically generated

*Figure 3.3.0.2*

Immediately Snort recorded that request and alerted to computer screen:

A screenshot of a computer program

Description automatically generated

*Figure 3.3.0.3*

Figure 3.3.0.4 is the rule that we configure for Snort for detection.

A screenshot of a computer screen

Description automatically generated

*Figure 3.3.0.4*

The format of each rule will include alert, protocol, external net, home net with the specific message and id. To get more specific alert, defender can configure rules to record:

A screenshot of a computer

Description automatically generated

*Figure 3.3.0.5*

Here are some basic rules that record TCP, Xmas, FIN attacks or UDP. For example, we can do -sU and -sX

A screenshot of a computer

Description automatically generated

*Figure 3.3.0.6: nmap -sU -f*

A screenshot of a computer

Description automatically generated

*Figure 3.3.0.7 nmap -sX -f*

Immediately Snort recorded -sU as “NMAP UDP” alert and -sX as “XMAS TREE” and “NMAP TCP” alert.

# IV) Phase 3: Gaining access

## 1. Incident Report:

After investigating the incident, we found that the username “Sang” have run a malware name “Not a Malware”, and that give the access to the attacker

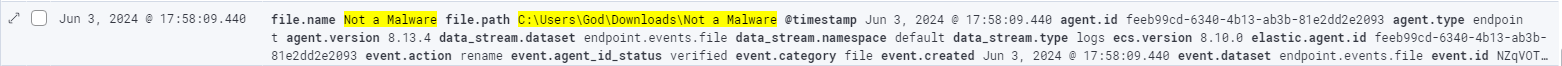


Figure 4.1.0.1.: Elastic show that user “Sang” run “Not a Malware”

After interrogating the employee “Sang”, we learn that the malware entered the system through phishing email. To stop the spread of the malware, we pinpoint all the end-devices that were affected by the malware and isolate them. After that, we download the malware to a sandbox to further analyze it.

## 2. Attack vectors analyze:

1. *Dynamic analyzes:*
2. *Tool use in this phase*

*Regshot:*

A screenshot of a computer

Description automatically generated

Figure 4.2.1.1: The regshot tool

Regshot is an open-source (LGPL) registry compare utility that allows you to quickly take a snapshot of your registry and then compare it with a second one (Sudhir, 2014)

2.1.2 *Conducting analyze:*

We take a snapshot of the machine before running the malware.

A screenshot of a computer

Description automatically generated

Figure 4.2.1.2: Taking the 1st snapshot

After the snapshot, we ran the malware that had infected our system name “Not a Malware”.

A screenshot of a computer

Description automatically generated

Figure 4.2.1.3: The “Not a Malware” malware

When the malware finishes running, we take the second snapshot of the machine and then compare them.

A screenshot of a computer

Description automatically generated

Figure 4.2.1.4: Taking the 2nd snapshot

A screenshot of a computer

Description automatically generated

     Figure 4.2.1.5: The output of the comparison

Compared to the first snapshot, the second have: 19 deleted keys, 6 keys were added, 58 values were deleted, 25 values were added, 120 values were modified, 23 folders deleted, 4 folders added, 63 files deleted, 27 files added, 169 files attributes modified.

1. *Static analyzes:*

*2.2.1 Tool use in this phase:*

*DIE*

A screenshot of a computer

Description automatically generated

Figure 4.2.2.1: DIE tool

Detect it easy or DIE is a tool used to identify the types of files.

HxD

A screenshot of a computer

Description automatically generated

Figure 4.2.2.2: HxD tool

HxD is a freeware hex editor, disk editor, and memory editor which can open files, edit the raw contents of disk drives, display and edit the memory used by running processes. (A, Monnappa, 2024)

2.2.2. Conducting analyzes:

The first step is to identify the file type of the malware by using DIE.

A screenshot of a computer

Description automatically generated

Figure 4.2.2.1: DIE detecting the malware

After running it in DIE, we learn that this malware type is DLL and it runs in PE32, which shows that the sample is from a 32-bit DLL file.

To understand more about the malware, we then use HxD to read its hex value.

A screenshot of a computer

Description automatically generated

Figure 4.2.2.2: same hex values in the malware seen in HxD

Witnessing this, we conclude this malware is trying to overflow our machine, this is the patent of buffer overflow attack.

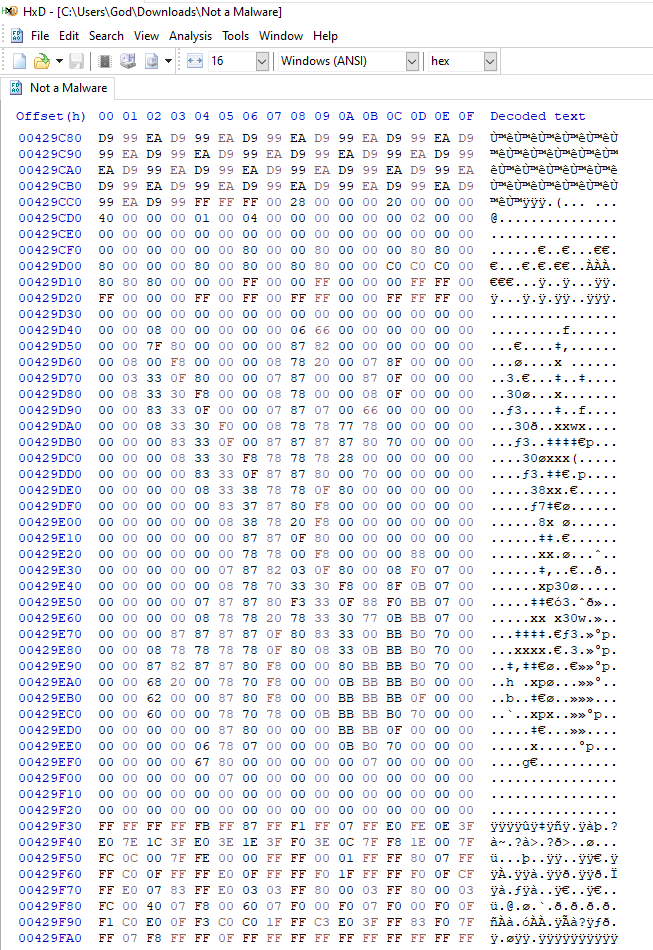


Figure 4.2.2.3: Payload after the loop values

When scrolling down, we seem to find the payload of the malware. For further investigation and finding ways to prevent this malware in the future, we use strings command on the malware to see what strings it might contain.

A screen shot of a computer

Description automatically generated

Figure 4.2.2.4: The strings of the malware using strings command

Looking at the strings, we can 100% confirm this malware is trying a buffer overflow attack on our machine by attempting to access the Registry.

## 3. Migration:

1. *Tool use in this phase:*

*Yara:*

YARA is a powerful pattern-matching tool and rule format used for identifying and classifying files based on specific patterns, characteristics, or content. SOC analysts commonly use YARA rules to detect and classify malware samples, suspicious files, or indicators of compromise (IOCs).

*Elastic stack:*

The Elastic stack, created by Elastic, is an open-source collection of mainly three applications (Elasticsearch, Logstash, and Kibana)that used for comprehensive search and visualization of log file sources.

1. *Implement migration:*

To prevent this malware in the future, we must create the Yara rule by identifying strings that are in the malware. After that, we create a Yara rule:

A screenshot of a computer program

Description automatically generated

Figure 4.3.2.1: Yara rule

Finally, we apply the Yara rule to elastic so whenever this type of malware enters the system, we can detect and prevent it

A screenshot of a computer

Description automatically generated

Figure 4.3.2.2: Elastic block and alert the malware

# V) Phase 4: Maintaining Access

## 1. Attack technique

In this phase, the attacker aims to maintain persistent access to the compromised system. By leveraging SSH (Secure Shell) techniques, the attacker configures SSH configuration to disable password authentication and enable public key authentication.

A computer screen shot of a computer screen

Description automatically generated figure 5.1.0.1: sshd configuration

The attacker generates an SSH key pair, placing the public key in the system's authorized keys file.

A screenshot of a computer

Description automatically generated figure 5.1.0.2: rsa generation

Then the attacker can add his private key to the system, or he can change the value of existing private key to his private key value to guarantee no-one can access except him.

## 2. Wazuh introduction

Therefore, we introduce Wazuh as a solution for detecting and responding for the persistence threats. Wazuh is an open-source security platform that includes XDR and SIEM capabilities. Like Elastic, Wazuh collects events from the agents or network devices, then forwards them to the server to check matched events or alerts. (Wazuh, 2024)

## 3. Demonstration

### 3.1) SSH backdoor prevention

To prevent attackers from gaining backdoor through SSH protocol, we need to configure file and directory integrity of .ssh directory and sshd\_config file in Wazuh, so any changes will be alert to SOC team to investigate closer. Firstly, we need to edit the Wazuh configuration file of the agent in the /var/ossec/etc/ossec.conf path.



figure 5.3.1.1: edit ossec.conf

A screenshot of a computer program

Description automatically generated figure 5.3.1.2: Wazuh agent configuration

Then we write the alert rule by specifying the folder that we are going to monitor for changes which is .ssh in this case.

A screenshot of a computer program

Description automatically generated figure 5.3.1.3: Adding rules send alert to Wazuh manager

After that, to make the wazuh server display on the dashboard, we need to write the event information like id, rules, messages on the local\_rule.xml

A computer screen shot of text

Description automatically generated figure 5.3.1.4: Adding rules to display alert on Wazuh manager

Finally, we are going to test the alert by adding, modifying and removing the key in .ssh folder, which is the attacker’s technique to maintain persistence in the system.

A screen shot of a computer code

Description automatically generated figure 5.3.1.5: Testing on agent

A screenshot of a computer

Description automatically generated figure 5.3.1.6: Dashboard alert

The figure 5.3.1.6 demonstrates changes in .ssh file can result in the alerts in Wazuh as we configured before. By following these steps, we are going to configure the same with the sshd\_config file, then we can totally detect attacker presence in the system.

### 3.2) MITRE ATT&CK detection

Thanks to the integration with MITRE ATT&CK, Wazuh can reduce the process by generating alerts corresponding to the attacker techniques in MITRE framework. For the demonstration, we are going to use Invoke –atomic red team which is the tool that automatically replicates red team’s techniques in the MITRE framework, then we are going to use Wazuh to detect it. Firstly, install the Invoke Atomic.

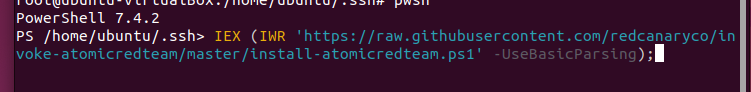


figure 5.3.2.1: Install Invoke Atomic

After that, choose any technique from MITRE then execute it in the program. The technique that we chose is T1548.003 which is privilege escalation and defense evasion tactic.

A screenshot of a computer program

Description automatically generated

figure 5.3.2.2: result of executing Invoke-Atomic

We can see the sudoers file with its configuration after executing this tactic. Now, we are going to check Wazuh’s alert and event.

A screenshot of a graph

Description automatically generated figure 5.3.2.3: Alert of T1548.003

A screenshot of a graph

Description automatically generated figure 5.3.2.4: dashboards of detected techniques

As we can see, Wazuh not only successfully detects the event, but it also visualizes data for all detected tactics.

VI) Phase 5: Prevention of clearing tracks

## 1) What is the clearing track phase?

After maintaing access, hacker can delete log files if the system has not set privileges correctly.

## 2) Solution for prevent clearing track

### 2.1) Solution 1: A real-time log file alert via email from the agent (Wazuh).

Step 1: setup wazuh real-time email alert with SMTP server with authentication

1. Mail server configuration type setup.

Command: apt-get update && apt-get install postfix mailutils libsasl2-2 ca-certificates libsasl2-modules

1. Configure Postfix server relay in /etc/postfix/main.cf directories.

A screenshot of a computer screen

Description automatically generated

*Figure  6.2.1.1*

1. Set sender email address and password.

In this step, we configure using Gmail host to send the Gmail, so we need to use Google APP password.

A screenshot of a black screen

Description automatically generated

*Figure  6.2.1.2*

1. Secure password DB file to only root user has privileges.

A screenshot of a computer

Description automatically generated

*Figure  6.2.1.3*

1. Restart Postfix.

Command: sudo systemctl restart PostFix

1. Configure email notifications in Wazuh server.

A computer screen shot of a program

Description automatically generated

*Figure  6.2.1.4*

Step 2: Configure <syscheck> on Wazuh Agent side.

1. Open configuration file

Command: subl nano /var/ossec/etc/ossec.conf

1. Add on <syscheck> section to monitor desire log file.

A screenshot of a computer

Description automatically generated

*Figure  6.2.1.5*

*Figure*

1. Save and restart Wazuh-Agent.

A screenshot of a computer

Description automatically generated

*Figure  6.2.1.6*

Step 3: Create rules on Wazuh-Manager.

1. Open rules file

Command: sudo nano /var/ossec/etc/rules/local\_rules.xml

1. Add custom rule to detect deletion.

A screenshot of a computer program

Description automatically generated

*Figure  6.2.1.7*

1. Save and restart agent.

Command: sudo systemctl restart wazuh-manager

Step 4: Verification and testing

1. Test Log file deletion and Security events on Wazuh-manager

* Command: sudo rm /var/log/syslog

A screenshot of a computer

Description automatically generated

Figure 6.2.1.7

1. Email Alert Notification

A white background with black text

Description automatically generated

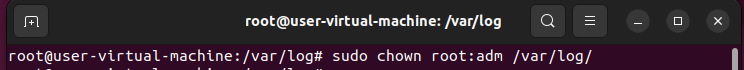
*Figure  6.2.1.9*

### 2.2) Solution 2: Additional Measures to Prevent Log File Deletions or Modifications (Wazuh).

1. Change File Permissions

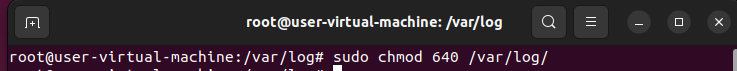
In this step, we will only give root the permissions.

Step 1: Change ownership and permission (root) of the log files.



*Figure  6.2.2.1*

Step 2: Ensure syslog has write permission.



*Figure  6.2.2.2*

Step 3: Verify with different user.

A screenshot of a computer

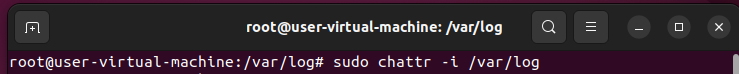
Description automatically generated

*Figure  6.2.2.3*

2. Use Immutable Attribute

This step is to set immutable attributes to prevent any changes to log file.

Step 1: Set immutable attributes.



*Figure  6.2.2.4*

3. Use Auditd to monitor directories

Auditd is an auditing framework used on Ubuntu to help monitor log activities of the system. Auditd can track user activities such as command execution or login attempts but also it can detect any system changes.

Step 1: Install ‘auditd’

Command: sudo apt install auditd audispd-plugins



*Figure  6.2.2.5*

Step 2: Add rule to monitor configuration in log file.

Command: sudo auditctl -w /var/log -p wa -k log\_monitoring

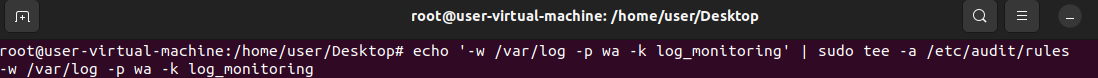


*Figure  6.2.2.6*

Step 3: Make this rule persistent and restart ‘auditd’

Command: echo '-w /var/log -p wa -k log\_monitoring' | sudo tee -a /etc/audit/rules

sudo systemctl restart auditd



*Figure  6.2.2.7*

A screen shot of a computer

Description automatically generated

*Figure  6.2.2.8*

# VI)  Network Security Challenges

While tools mentioned in previous sections are robust at protecting the network, they also have drawbacks. The following section will discuss the challenges related to securing network, focusing on the following tools:

* IPTables
* Snort
* Elastic
* Wazuh

Firstly, the tools need to be configured manually, which requires the equivalent expertise and time to set up. Security knowledge is vital to use these tools effectively, acknowledging characteristic of the traffic and the network itself as well as the configuration. Furthermore, significant customization is necessary when implementing of these tools to existing network and infrastructure, making it complex and time consuming. Due to complexity of implementation and management, they can be easily misconfigured, causing performance issues and vulnerabilities in the network.

Secondly, monitoring and filtering is a continuous process, these tools demand on computer power, storages and network capabilities, these issues are easily addressed with additional hardware and process power but come at high cost. In general, IDS and IPS are known to affect network performance such as latency, Alka and Lalit (2020) shows that Snort performance is degrade in large traffic and its struggle to deal with larger packets, as shown in the increased packet drop rate. In addition, having a large set of IP rules also slower packet inspection, such as Iptables is affected by this. In conclusion, these tools are resource extensive and may affect network performance.

Thirdly, systems such as SIEM, firewall, IPS and IDS sometimes produces inaccurate results, false positives and false negatives are unavoidable and requires cautions from those monitoring on these systems.

Lastly, most tools presented are open source, meaning users have limited feature access. For example, Snort and selfhosted Wazuh requires manual security rules update while paid products such as Wazuh-cloud, paid Snort, and Elastic are updated automatically. Another example comparing Iptables to modern firewalls, Iptables fall short of features such as deep packet inspection, VPN capabilities and user friendliness. While provided tools are effective solutions, they have limited features or less effectiveness compared to dedicated/paid tools.

# VII) Conclusion

As the defensive side, we must understand the attacker techniques to prevent him in each phase. During reconnaissance, we should implement mail security protocols like SPF, DKIM and DMARC for email spoofing prevention. Next, implement zone-based firewall that can segment the network and configure the Iptables properly along with Snort can prevent scanning. After that, we should isolate affected network devices and analyze malware behaviors, then write Yara rules for SIEM tool like elastic to prevent it entering the system in the future. Configure Wazuh to detect any anomalies action and prevent log deletion by using Wazuh or Auditd.

**Word count: 3,221**

# VIII) References

1. Wazuh (n.d.). *Configuring email alerts - Wazuh server · Wazuh documentation*. [online] documentation.wazuh.com. Available at: https://documentation.wazuh.com/current/user-manual/manager/manual-email-report/index.html [Accessed 18 Jun. 2024].
2. ‌Wazuh. “Basic Settings - File Integrity Monitoring · Wazuh Documentation.” *Documentation.wazuh.com*, documentation.wazuh.com/current/user-manual/capabilities/file-integrity/basic-settings.html. Accessed 18 June 2024.
3. Pandey, Sudhir Kumar, and B.M. Mehtre. “Performance of Malware Detection Tools: A Comparison.” *2014 IEEE International Conference on Advanced Communications, Control and Computing Technologies*, May 2014, <https://doi.org/10.1109/icaccct.2014.7019422>.
4. A, Monnappa K. *Learning Malware Analysis: Explore the Concepts, Tools, and Techniques to Analyze and Investigate Windows Malware*. *Google Books*, Packt Publishing Ltd, 29 June 2018, books.google.com.vn/books?hl=en&lr=&id=QsNiDwAAQBAJ&oi=fnd&pg=PP1&dq=HxD+malware+analysis&ots=FJRCLmki5N&sig=4xf6XdhU0yP8ST4zPe8NMg8AiHc&redir\_esc=y#v=onepage&q=HxD%20malware%20analysis&f=false. Accessed 19 June 2024.
5. Wazuh. “Components - Getting Started with Wazuh · Wazuh Documentation.” *Documentation.wazuh.com*, documentation.wazuh.com/current/getting-started/components/index.html.
6. Wazuh. “Detecting and Removing Malware Using VirusTotal Integration.” *Documentation.wazuh.com*, documentation.wazuh.com/current/proof-of-concept-guide/detect-remove-malware-virustotal.html.
7. Snort.org. (2020). *Snort - Network Intrusion Detection & Prevention System*. [online] Available at: <https://snort.org/>.
8. Shaimaa H. Abd, Ivan A. Hashim and Ali Sadeq A. Jalal, “Automated Deception Detection Systems, a Review” (2021) Iraqi Journal of Science, pp. 70–80. doi:10.24996/ijs.2021.SI.2.8.
9. Alka Gupta and Lalit Sen Sharma, “Performance Analysis and Comparison of Snort on Various Platforms” (2020) International Journal of Computer Information Systems and Industrial Management Applications. ISSN 2150-7988 Volume 10 (2020) pp. 023-032
10. Hu, H., Peng, P. and Wang, G. (2018). *Towards Understanding the Adoption of Anti-Spoofing Protocols in Email Systems*. [online] IEEE Xplore. doi:https://doi.org/10.1109/SecDev.2018.00020.

## Appendix

A screenshot of a computer

Description automatically generated