Server & Cloud Security

SCHOOL OF INFOCOMM TECHNOLOGY

Diploma in Cyber Security & Forensics

Diploma in Information Technology

GROUP ASSIGNMENT

Duration: Weeks 5 -17

Weightage: 15% out of Total 30%

Tutorial Group: T02 **Team Number: 3** **Team Grade: _____**

Student Name	Student Number
1. Kok Jia Cheng Glenn	S10208391E
2. Quak Ze Shuo, Tommy	S10194190H
3. Reanee Chua Yun Lin	S10206117E

Table of Contents

1 Introduction

When performing deployment of any servers, it is a crucial step to perform server hardening to improve the security posture of the server by reducing its surface of vulnerability by patching and implementing security measures to secure the system.

In this case, the Linux web server will be secured, which is essential to protect the data and intellectual properties from the hands of hackers. An out of the box installation of Linux will usually contain a relatively high level of security risks due to the presence of unneeded services and lack of security controls. As such, it is crucial to take the relevant steps to eliminate as many security risks as possible to reduce the attack surface.

In this report, the steps taken to harden the server will be outlined. The methods and tools to ensure that the OS is continuously patched and kept up to date automatically will be explored, which would ensure that most vulnerabilities present in older versions can be stamped out quickly and effectively.

Server OS hardening will also be performed, which would allow the attack surface to be reduced by disabling unnecessary components, as well as improve the server security by implementing security controls through strict account management and properly configured resource controls respectively.

To ensure that the server can be accessed remotely and in a secure manner for administrative purposes, Secure Shell (SSH) will be implemented and utilized on the server. Alongside that, relevant SSH hardening measures will also be taken to further bolster security. An Intrusion Prevention System (IPS) will be configured as a layer of protection to detect malicious attempts to access the server and take the appropriate remedial actions.

In addition to hardening the server, it is also necessary to secure the web service that can potentially compromise the server. Measures such as the inclusion of an SSL certificate would help keep interaction between client and server secure via HTTPS. Also, a web application firewall is implemented to protect against application layer attacks. Directory listing will also be disabled to prevent attackers from enumerating through the site directory and obtaining sensitive information.

To prevent unauthorized access to the server, a host-based firewall will be implemented with the use of Linux’s Uncomplicated Firewall (UFW) to reduce the

attack surface. Also, an anti-malware software will also be installed to detect and rectify any potential malware incursions.

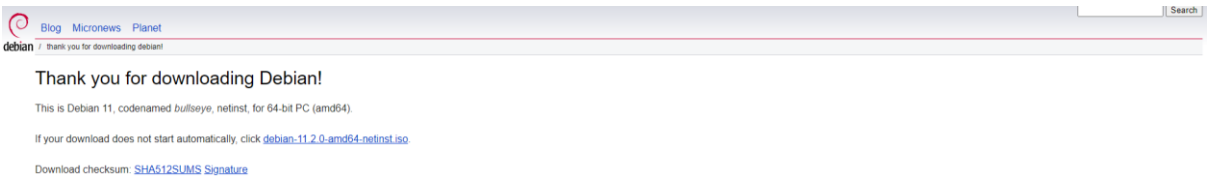
Moreover, full disk encryption will be implemented to ensure that there is a layer of security measure in place to prevent hackers from accessing data even if they have physical access to the disk.

Other measures which are deemed effective for bolstering security of the server will also be implemented. These include but are not limited to addition of boot loader security, disabling external media and a backup solution.

Lastly, once the control measures are deployed and configured, vulnerability scanning, and security auditing will be carried out to gauge the effectiveness of the server hardening process. Following that, the security issues detected by the tools would be addressed to further harden the server. The tools and software installed on the server to reduce the attack surface will be discussed in this report in the following sections.

2 Operating System

After research, the Linux distribution chosen as the most suitable Operating System (OS) for the Web Server is Debian. It is an open-source Linux-based OS that is known for being stable and secure as it provides reasonable default configuration for every package as well as regular security updates during the packages' lifetimes. Debian also provides smooth upgrades as it is easy to keep the system up to date. However, a system is only as secure as its administrator is capable of making it. Hence, this report would discuss the measures to reduce the attack surface and secure the server.



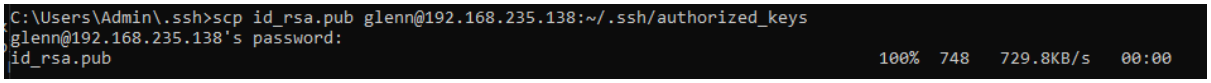
2.1 Server Roles

The server should be a dedicated, single purpose host. Thus, unnecessary services and applications should be disabled, uninstalled, or not installed at all. Since the server is to be deployed as a web server, Apache service would be installed as well as Secure Shell (SSH) for secure remote access.

2.1.1 Secure Shell (SSH)

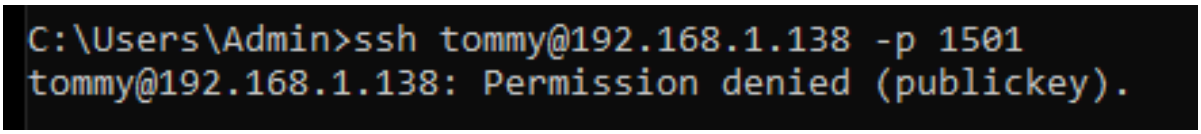
SSH service provides convenience and efficiency in accessing and configuring the server remotely. However, it can potentially attract many threats and compromise the server if not protected correctly. Hence, the SSH service must be hardened and protected from potential threats.

After installation of the SSH service, key-based authentication should be used instead of password-based authentication for remote access. This can significantly reduce the risk of unauthorized access into the server through means such as credentials brute-forcing as only clients with the correct key-pair are allowed access to the server. To achieve this, each user is assigned a unique RSA key-pair which is generated with a key size of 4096 bits to increase security. Due to the large key size, it would be almost impossible to crack the keys. The public key is then sent over to the server and stored in the home directory of the respective user under a hidden folder named “.ssh”. The hidden folder is also configured to restrict access and modification by other users of the system as this file is highly sensitive.



With this implementation, remote access to the server via SSH would only permit users that have the private key corresponding with the public key on the server. For Windows Systems, the private key can be stored in “C:\Users\<USER>\.ssh”. Furthermore, hardening the service is also necessary. This includes disabling root login and disallowing password authentication. As seen below, a successful SSH session was created without entering a password since key-based authentication was implemented.

Below is an example of a user accessing the server without a valid private key.

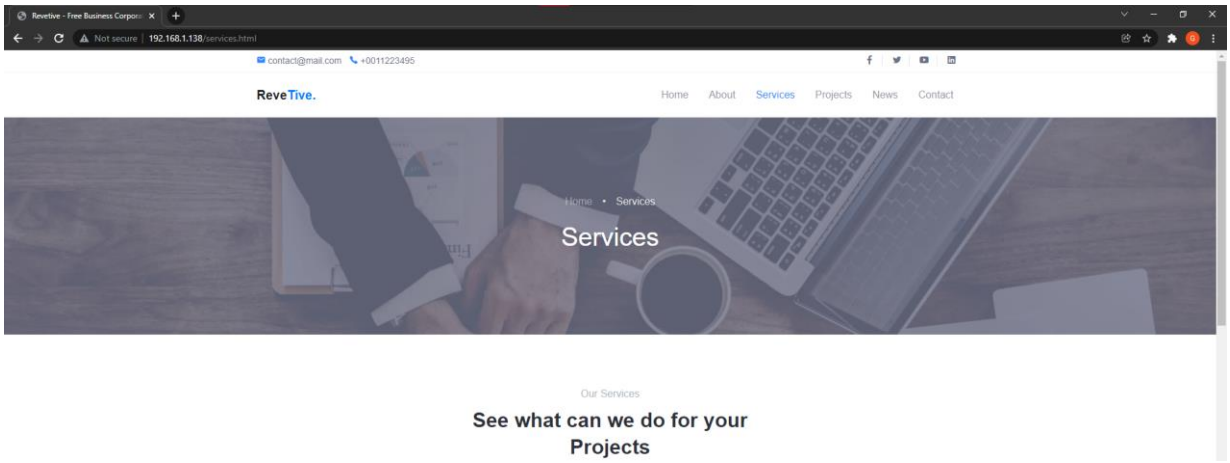


Additionally, changing the default port of SSH (22) to a non-standard port (1024 and above), or registered port, does not provide any enhancement to security but it reduces the risk of detection from port scanning tools and techniques that uses standard ports (below 1024). Hence, the SSH port was changed to port 1501. When accessing via SSH, the port number must now be specified.

```
C:\Users\Admin>ssh glenn@192.168.1.138 -p 1501
Linux scs-webserver 5.10.0-10-amd64 #1 SMP Debian 5.10.84-1 (2021-12-08) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sat Jan  1 15:14:26 2022 from 192.168.1.214
glenn@scs-webserver:~$
```

2.1.2 Apache Web Server

The web service to be installed is Apache which is a free open-source web server that delivers web content through the Internet. Despite Apache being built to be stable and secure, it will only be as secure as the administrator that configures it. Hence, once Apache is installed, it is important to configure the server as having default configuration can reveal much sensitive information which may help hackers to prepare for attack towards the web server. The control measures in securing the web service are addressed in Application Layer Hardening.



3 Patch and Upgrade the Operating System (OS)

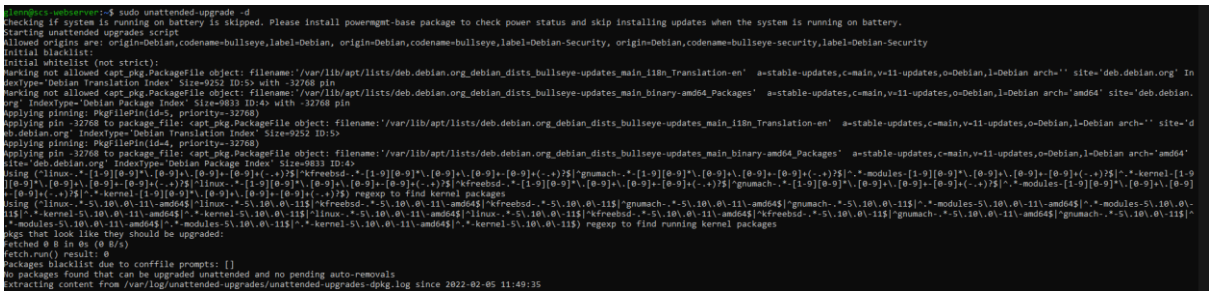
Once the OS is installed, it is critical that the system is patched and upgraded to correct for known vulnerabilities. It is important that the known vulnerabilities in the system are fixed before deploying for production or hosting a server and exposing it to untrusted users. This section would address the control measures implemented into the system as well as the processes the administrator should follow to ensure that the server is patched and known vulnerabilities are promptly addressed and corrected.

3.1 Automatic Upgrades

Manually checking for new patches every day can be tedious and requires unnecessary overheads when patch management automation is available. Unattended-upgrades is a package that keeps the system updated with the latest security patches automatically. Once it is configured, the server would automatically perform a self-update without the need for an administrator.

As seen below, unattended-upgrades is set to run updates and upgrades automatically daily to ensure that the server is always up-to-date with the latest patches and fixes to correct known vulnerabilities.

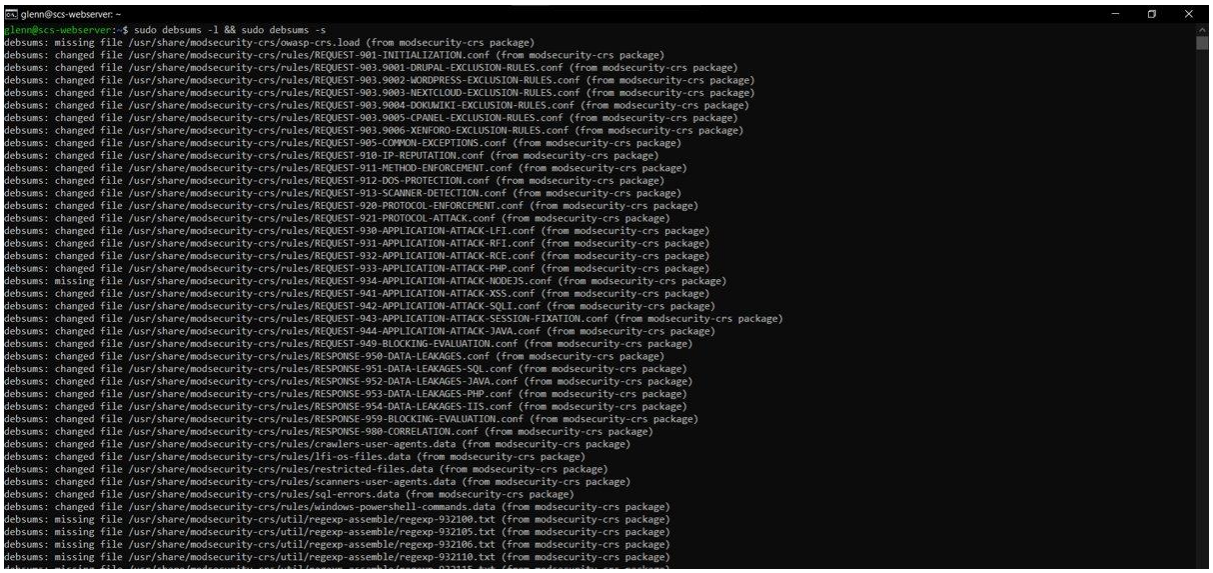
The administrator can also manually run unattended-upgrades and force updates and upgrades when needed. Using the debug mode, it will start checking for available updates and perform the necessary actions if a newer version is discovered.



3.2 Verifying Installed Packages

Despite having a feature to perform automatic upgrades, verification of the packages installed must also be performed. Debsums is implemented and is intended primarily

as a way of determining what installed files have been locally modified by the administrator or damaged by media errors. It can be used to list packages that do not have an MD5 checksum file which administrators should follow up and attend to as it indicates that these packages are corrupted and should be re-installed or fixed. As seen below, the command used can reveal packages that have missing MD5 hashes as well as suspected errors within the files.



3.3 Implement Patching Process

It is important that a patching process is implemented and strictly followed to ensure the security of the server. Also, patching remediates bugs, security vulnerabilities and adds functionality to the software. Some patches fix issues with drivers and software running on the system. Hence, integrating a patch management system can automatically detect updates, download them, and then deploy them to the server. The administrator should have a scheduled routine to check the system for updates every month and patch the system. Despite implementing automatic upgrades discussed earlier in this section, it is also critical for the administrator to do routine checks.

In addition to verifying the integrity of the packages using Debsums, it is equally as important to ensure that all packages are up to date. Using the apt-show-versions package would allow the administrator to list packages that are not updated when performing their routine checks.

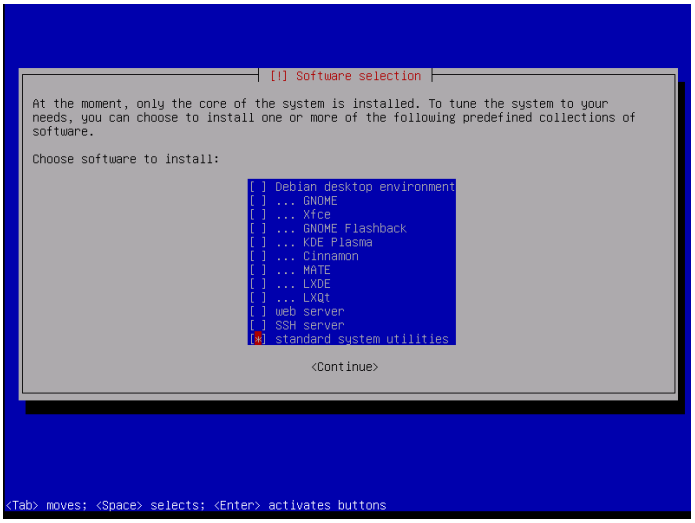
4 Server Operating System Hardening

Hardening refers to securing the server by reducing its surface of vulnerability. Thus, a hardened system is configured and updated to protect against cyberattacks. It is also the most effective preventive measure when designing system security as it ensures that all aspects of Server OS are protected. This section will address the controls implemented on the server to reduce the surface of vulnerability to attackers.

4.1 Remove Unnecessary Service

The server should be installed as a dedicated, single-purpose host. Hence, common types of services should be removed, disabled, or not installed. Since the server is to be deployed as a web server, Apache service would be installed as well as Secure Shell (SSH) for ease of remote administration. Moreover, services such as Simple Mail Transfer Protocol (SMTP) for email services, File Transfer Protocol (FTP) for file sharing, and other applications not needed on the system and should not be present on the server. This will mitigate the risks of having vulnerable components on the web server.

Additionally, installing the Operating System with no Graphical User Interface (GUI) reduces the attack surface and minimizes the complexity of the system. Thus, security is enhanced through having a minimalistic system and lack of unnecessary applications or services installed on the system. Only necessary services are installed onto the server.

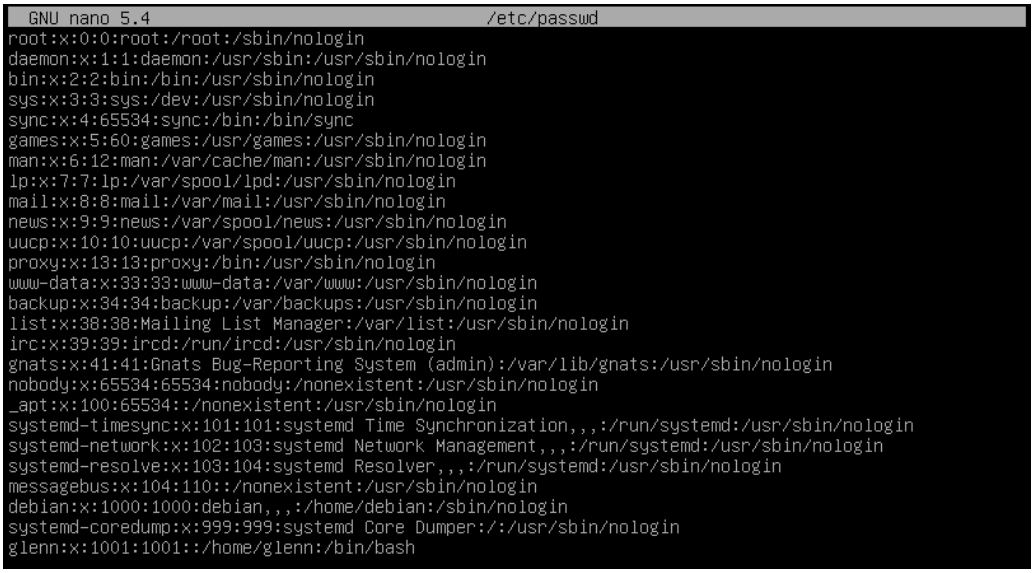


4.2 Configure Operating System User Authentication

Authorization for administration and configuration of the Operating System should be limited to a small number of designated server administrators within the Sudo group. Each user that has access to the server is assigned to a group. For example, the web developers are assigned to the web-dev group while log monitoring users are assigned to the adm group, which is a built-in Linux group used for monitoring files in the /var/log folder. Access to the server must be enforced as well by implementing appropriate user authentication. On the server, it is also best practice to disable the default accounts as well as the root account. The table below depicts the user accounts and configurations.

Type	Username	Group	Password	Login Status
Superuser	root	root	Sc$#w3bSvr!	Disabled
Default user	debian	debian	Gu3$s_!t#	Disabled
User	glenn	sudo	P@$$w0rd	Enabled
User	tommy	web-dev	P@$$w0rd	Enabled
User	reanee	web-dev	P@$$w0rd	Enabled
User	john	adm	P@$$w0rd	Enabled

As seen in below, the root user account and default account, Debian, are disabled.



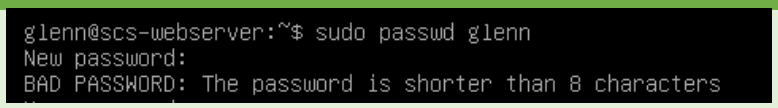
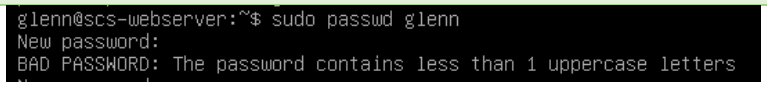
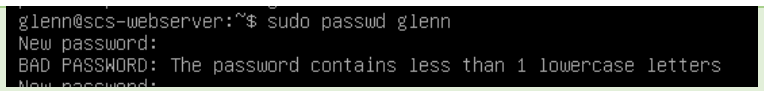
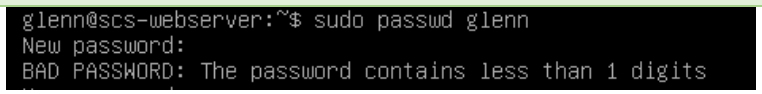
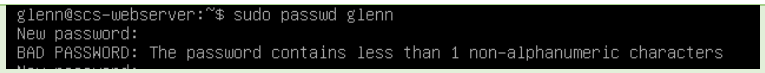
4.2.1 Password Policy

In addition to restricting access, security controls must also be implemented on users that have access. Hence, a password policy is enforced on the system with the addition of libpam-pwquality package. Based on best practices, users must change their password after 90 days and they are only allowed to change their password once a day. This would ensure that the passwords are regularly changed and that attackers have a reduced timeframe to gain access once the credentials are leaked. Hence, changing passwords often reduces the risk that an attacker will have frequent access to the server.

Moreover, a strict password policy was enforced with strong requirements based on best practices which are as follows:

- Minimum length of 8 characters
- At least 1 uppercase character
- At least 1 lower-case character
- At least 1 digit
- At least 1 non-alphanumeric character

Some examples of testing the password policy can be seen in the table below. The complex password requirements will ensure that user accounts are not susceptible to brute force attacks.

Requirement	Test Results
Character Length	
Uppercase Letter	
Lowercase Letter	
Digit	
Symbol	

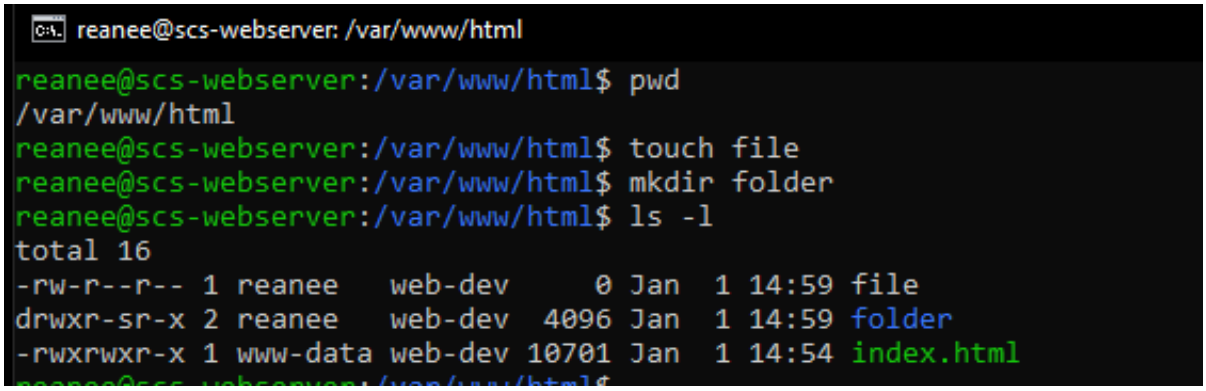
4.3 Resource Controls

By default, Linux uses a Discretionary Access Control (DAC) approach, the users would be assigned to groups where the permissions on the files or folders would determine the access rights of the user. Following the security principle of least privilege, the users should only be given the minimum amount of privilege needed to perform the job. Hence, the web developers should only be allowed full access within their own home folders and the /var/www folder to configure the website. As for the adm group users, it is strictly used for system monitoring tasks, and members of this group are only allowed to read log files in /var/log. Since root is disabled, members of the sudo group are in-charge of administering the server.

4.3.1 Directory Permission

After installation of the Apache service, resource control to the web root directory must be assigned to the web-dev group. It should also be configured such that the owner and web-dev group have full access to the /var/www directory. Also, new files or directories created will belong to the web-dev group. This would allow other members of the web-dev group to access the resources.

For the web developers, in the server, any files and directories created in the /var/www folder would inherit the user group which is web-dev. Thus, new files created by a user belonging to the web-dev group will automatically be allowed to be read by other users of the web-dev group.



5 Additional Security Tools and Configurations

To enhance security of the Linux web server, multi-layered security controls should be implemented to harden the server. With application of this concept, an attacker would have to bypass all security measures in place to fully compromise the system. However, despite the current security controls in place, it is not possible to withstand against all attacks. Hence, additional security control must be installed and configured on the server to enhance the security.

5.1 Anti-Malware

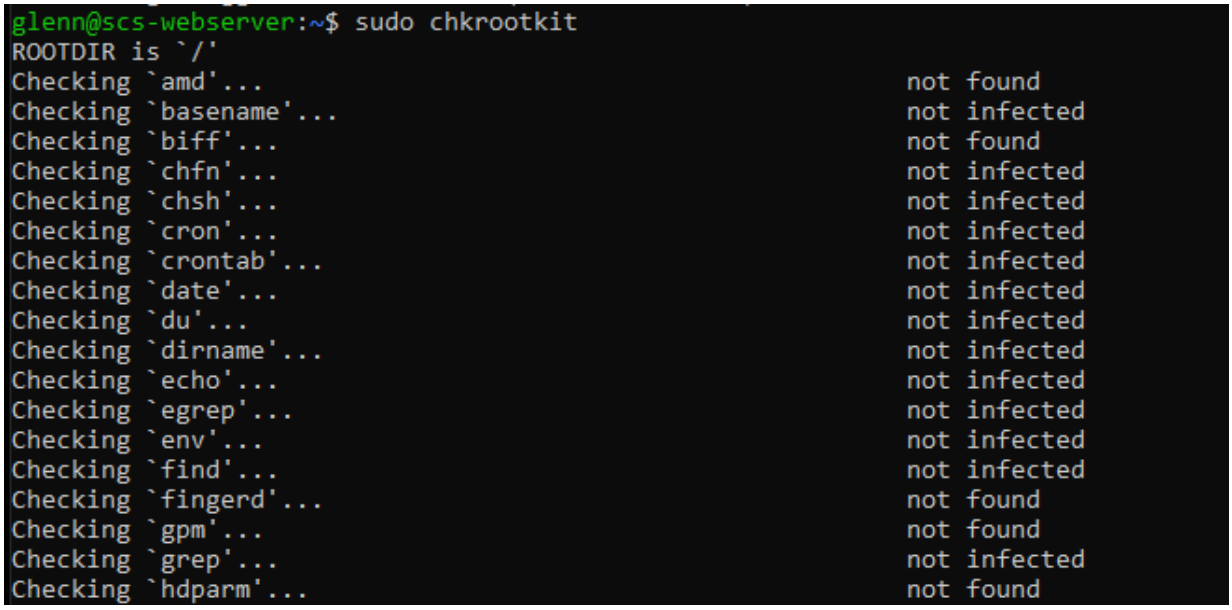
It is important to install anti-malware software to any system due to the rise of malware attacks such as ransomware, worms and other viruses which are able to severely compromise a system. Thus, in order to minimize the risk and impact of such attacks,

anti-malware software is needed to protect the operating system from malware and to detect and eradicate any infections that occur within the system.

The initial anti-malware that was installed was RKHunter, but it was later removed during installation as the package required Exim for email services. However, this also automatically configures the system into an SMTP server, which was not acceptable since this server should be a dedicated host server.

Therefore, chkrootkit was selected to be the anti-malware solution. Chkrootkit is a shell script that checks system binaries for rootkit modification. Rootkits are dangerous as it allows an unauthorized user to gain access to the system, thus, with the help of this tool, an administrator is alerted to potential threats within the system so they can plan the further steps to take to contain the impact of the malware or to narrow down the source for removal.

As mentioned before, chkrootkit is able to search for potential threats within the system and then list potentially malicious files for administrators to investigate. It makes use of signature detection, performing a detailed process check and scanning system binaries to detect kit signatures.

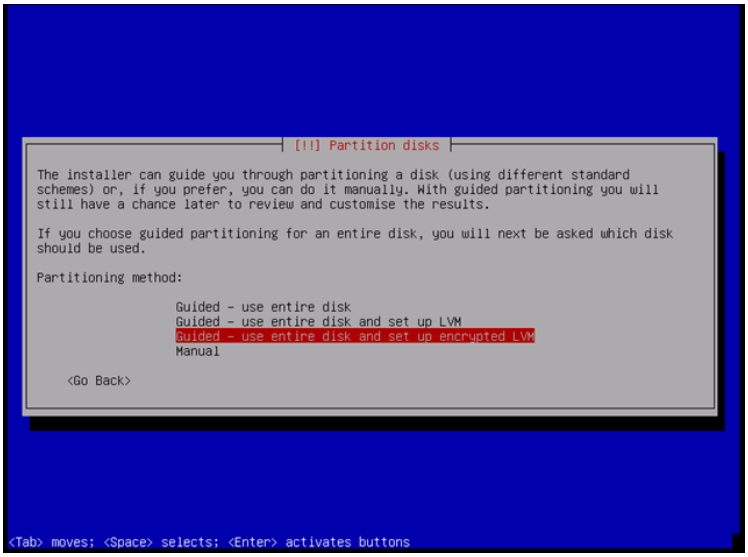


5.2 Disk Encryption

Apart from securing and hardening the software components of the server, it is critical to implement security on the physical aspects of the server as well. In case an attacker manages to gain physical access to the server, it is important to protect the data on the server's drive so that they cannot gain access to confidential information.

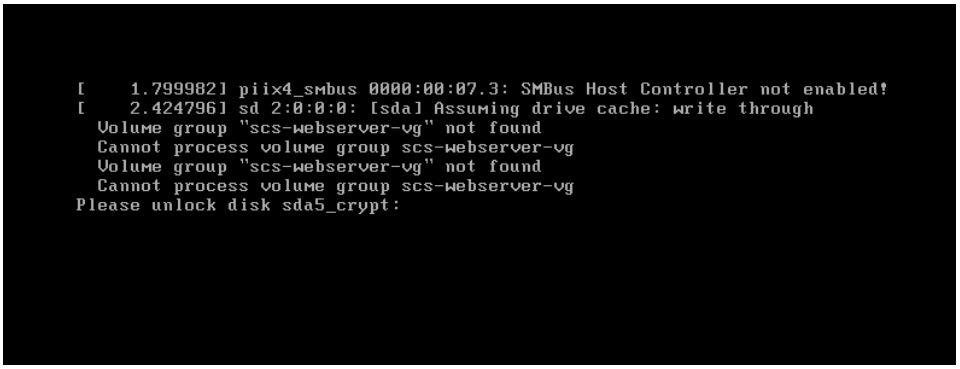
This is where encryption comes into play. By implementing full disk encryption on the server, additional security is provided beyond the existing OS security mechanisms by protecting the device's contents even if it has been physically removed from the system. If an attacker gains access to the device, they will still have to decrypt the contents of the drive with the decryption key before they can access the information on it, thus preventing an attacker from attempting to access the information.

During the installation of the Operating System, the entire disk was configured to be encrypted as seen below.



The disk can only be decrypted and read with a passphrase upon bootup. This prevents threat actors from being able to extract the physical drive to access the information directly and attempting to read the data on a different machine as they will need to boot the drive and enter the correct passphrase in order to decrypt it.

However, the level of security for this method depends on the strength and secrecy of the password. The password should be a strong, secure, and uncrackable password and its security must be ensured by keeping it confidential within authorized individuals. If the passphrase is disclosed, intruders can easily decrypt the drive and gain unauthorized access.



As seen below, the entire drive is encrypted using the Linux Unified Key Setup (LUKS) which establishes an on-disk format for the data, as well as a passphrase. In this case, the passphrase was configured as “@W,k4+(A:Lxfq;D_/4f=”.

```
root@scs-webserver:~# blkid /dev/sda5
/dev/sda5: UUID="b48ba998-f9b3-476a-b64f-1156b360290a" TYPE="crypto_LUKS" PARTUUID="c613ceec-05"
```

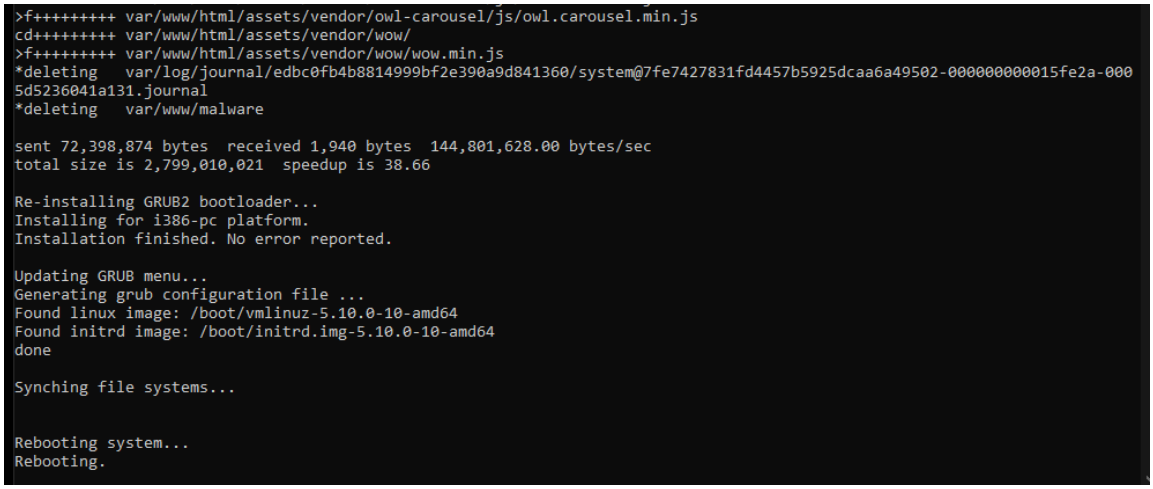
5.3 Backup Recovery

Along with hardening the system, disaster recovery is also crucial if the system is unavailable or gets taken down as security is never 100 percent guaranteed. In case the system’s functionalities are compromised, a backup of the system is crucial in restoring the functionality and services of the server to a state similar or better from before it was compromised. Thus, backups must be made as part of a Disaster Recovery Plan.

Timeshift for Linux is an application that provides functionality for system restoration by capturing a snapshot. When a security incident occurs, the latest backup can be used to restore the web server to the point in time the snapshot backup was created.

```
glenn@scs-webserver:~$ sudo timeshift --create --comments "weekly backup" --tags W
First run mode (config file not found)
Selected default snapshot type: RSYNC
Mounted '/dev/dm-1' at '/run/timeshift/backup'
Selected default snapshot device: /dev/dm-1
--------------------------------------------------------------------------------
Estimating system size...
Creating new snapshot...(RSYNC)
Saving to device: /dev/dm-1, mounted at path: /run/timeshift/backup
Synching files with rsync...
Created control file: /run/timeshift/backup/timeshift/snapshots/2022-01-09_18-06-10/info.json
RSYNC Snapshot saved successfully (53s)
Tagged snapshot '2022-01-09_18-06-10': ondemand
--------------------------------------------------------------------------------
```

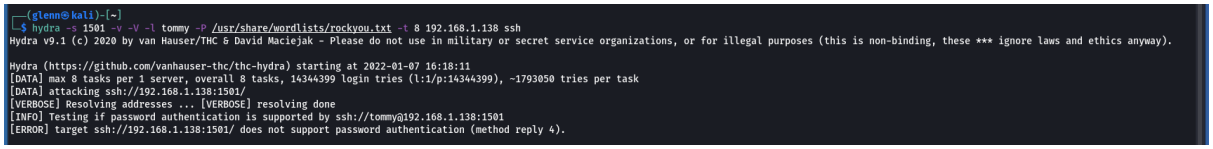
For example, in a scenario where a malware was planted within the system, Timeshift can delete and create any files based on the last backup taken. As seen below, it is capable of deleting the malware file and restoring any files that were previously removed or damaged to the point where the backup was performed.



As of this implementation, it is highly recommended that the backup files are stored remotely on another secured server instead of storing it on the local machine. This is to ensure that the backup sources are not modified or deleted when the web server is compromised. If the attacker were to gain access and remove the backups, the server would have no restoration point to fall back on. Hence, for future implementations, the backup files should be relocated and stored on a server which is logically and physically separated from the web server.

5.4 Secure Shell (SSH) Hardening

Although SSH has been configured to be secured and mitigated from possible attacks with encrypted communication, more can be done to harden the configuration of this service to reduce the attack surface. Firstly, port forwarding is disabled by setting AllowTCPForwarding to “no” as there is a substantial risk that users will use SSH tunneling to open backdoors into the server. Secondly, specifying the maximum number of authentication attempts permitted per connection to 3. Lastly, modify the logging level to verbose to capture more detailed remote authentication logs. As seen below, an attacker is unable to perform brute-force attempts using automated tools such as Hydra to gain unauthorized access due to disabling password authentication.

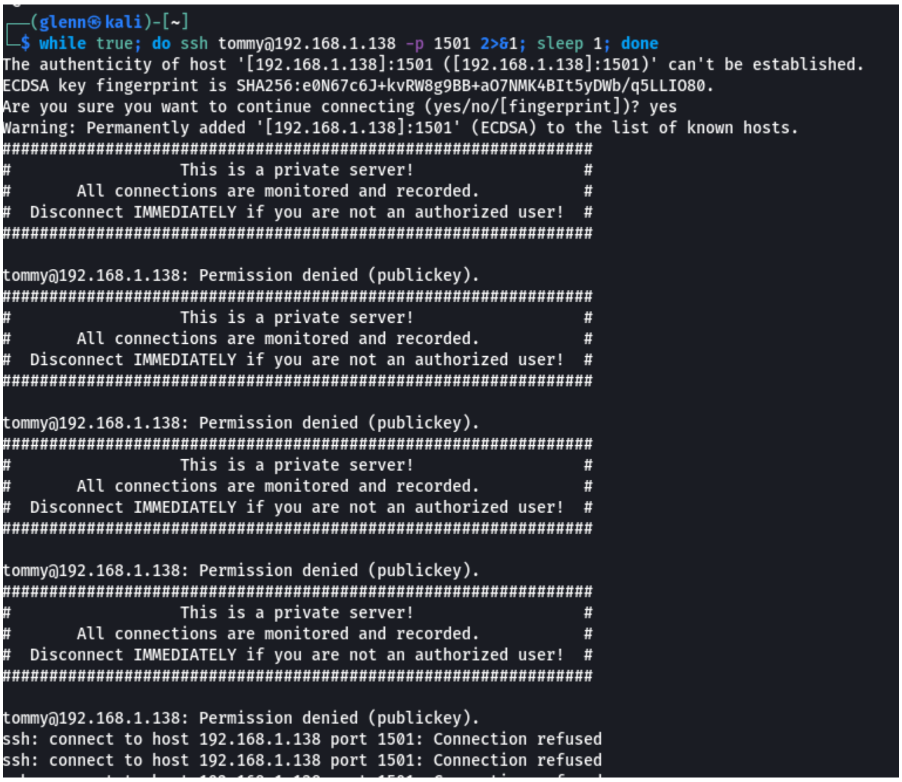


5.4.1 Intrusion Prevention System

Intrusion Prevention System (IPS) is a network security solution that examines network traffic and detects malicious inputs to prevent various attacks. It not only detects malicious inputs but also protects the network from harmful attacks. It can guard against brute-force attacks such as DoS (Denial of Service), DDoS (Distributed Denial of Service), exploits, worms, viruses, and other typical threats. When an alarm is triggered, malicious packets can be dropped, and suspicious IP addresses can be blocked by the IPS.

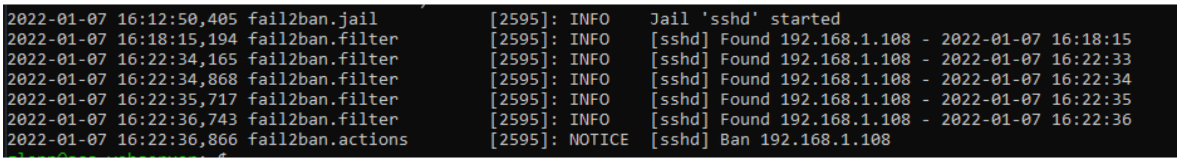
In this server, Fail2Ban, which is an Intrusion Prevention Software package, will be implemented on the server to add a security layer against different brute-force attacks. Despite password authentication for SSH being disabled, there is still a possibility of brute-force attempts using private keys. This would mitigate attacks, for example, in a scenario where the attacker managed to get a hold of a list of private keys used on the server. In that case, the attacker can utilize that list to test all private keys to gain access to a specific user account on the server. Hence, Fail2Ban would mitigate this threat by blocking that specific IP address after multiple failed attempts.

Below is an attempt to simulate an attack by repetitively generating multiple failed logons onto the server via SSH.



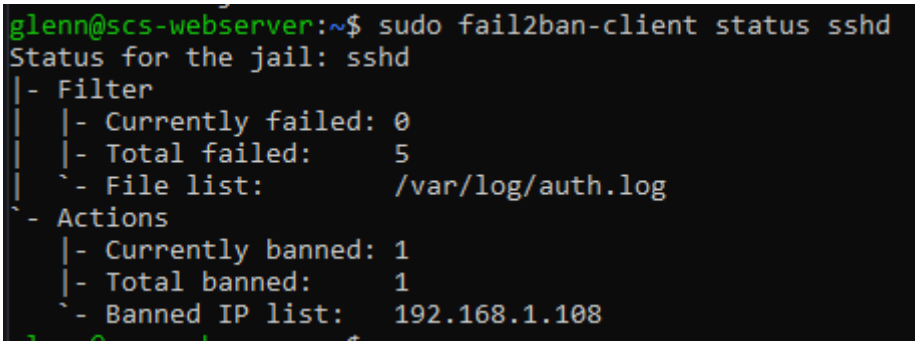
```
┌──(glenn㉿kali)-[~]
└─$ while true; do ssh tommy@192.168.1.138 -p 1501 2>&1; sleep 1; done
The authenticity of host '[192.168.1.138]:1501 ([192.168.1.138]:1501)' can't be established.
ECDSA key fingerprint is SHA256:e0N67c6J+kvRW8g9BB+aO7NMK4BIt5yDWb/q5LLIO80.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[192.168.1.138]:1501' (ECDSA) to the list of known hosts.
###############################################################
#                  This is a private server!                  #
#        All connections are monitored and recorded.          #
#  Disconnect IMMEDIATELY if you are not an authorized user!  #
###############################################################

tommy@192.168.1.138: Permission denied (publickey).
###############################################################
#                  This is a private server!                  #
#        All connections are monitored and recorded.          #
#  Disconnect IMMEDIATELY if you are not an authorized user!  #
###############################################################

tommy@192.168.1.138: Permission denied (publickey).
###############################################################
#                  This is a private server!                  #
#        All connections are monitored and recorded.          #
#  Disconnect IMMEDIATELY if you are not an authorized user!  #
###############################################################

tommy@192.168.1.138: Permission denied (publickey).
###############################################################
#                  This is a private server!                  #
#        All connections are monitored and recorded.          #
#  Disconnect IMMEDIATELY if you are not an authorized user!  #
###############################################################

tommy@192.168.1.138: Permission denied (publickey).
ssh: connect to host 192.168.1.138 port 1501: Connection refused
ssh: connect to host 192.168.1.138 port 1501: Connection refused
```

As seen below, these are the logs generated from Fail2Ban when an attempted intrusion was detected. The IP address of the attacker was 192.168.1.108 and after 5 failed attempts within a certain timeframe, the IP address was blocked. This IPS can also be further utilized by manually blocking known suspicious IP addresses along with detecting new sources of threat.

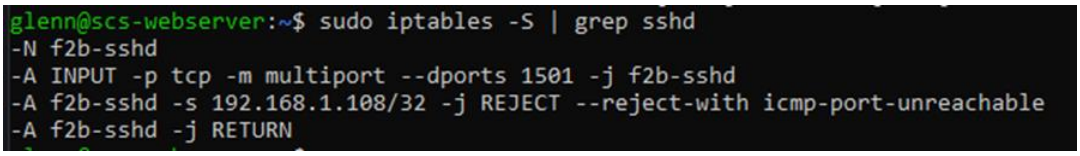


```
2022-01-07 16:12:50,405 fail2ban.jail      [2595]: INFO    Jail 'sshd' started
2022-01-07 16:18:15,194 fail2ban.filter    [2595]: INFO    [sshd] Found 192.168.1.108 - 2022-01-07 16:18:15
2022-01-07 16:22:34,165 fail2ban.filter    [2595]: INFO    [sshd] Found 192.168.1.108 - 2022-01-07 16:22:33
2022-01-07 16:22:34,868 fail2ban.filter    [2595]: INFO    [sshd] Found 192.168.1.108 - 2022-01-07 16:22:34
2022-01-07 16:22:35,717 fail2ban.filter    [2595]: INFO    [sshd] Found 192.168.1.108 - 2022-01-07 16:22:35
2022-01-07 16:22:36,743 fail2ban.filter    [2595]: INFO    [sshd] Found 192.168.1.108 - 2022-01-07 16:22:36
2022-01-07 16:22:36,866 fail2ban.actions   [2595]: NOTICE  [sshd] Ban 192.168.1.108
```

Fail2ban was configured with a jail specifically for detecting and responding to intrusions regarding the SSH service. The screenshot below shows that the IP address of the attacker was banned and placed in the SSH jail. This shows that Fail2Ban is effective in mitigating any brute force attacks towards remote access via SSH.



Also, as seen below, fail2ban accomplishes this prevention by automatically creating a rule in the IP tables to reject traffic involving the suspicious IP address when an intrusion attempt is detected.



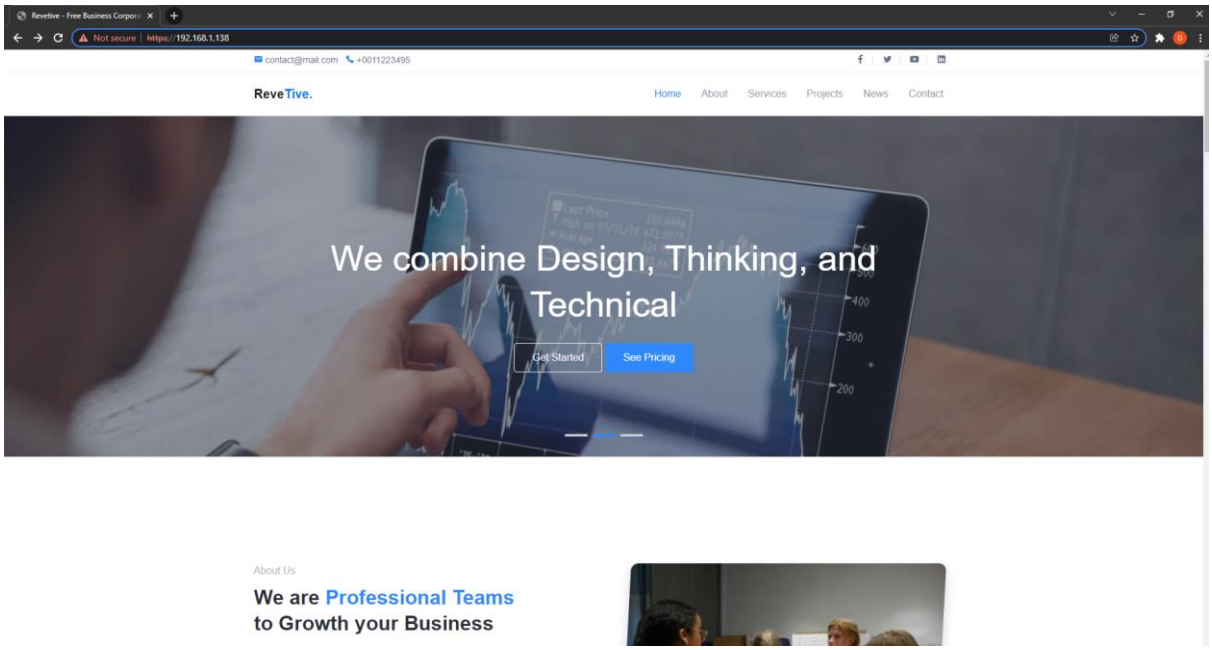
5.5 Application Layer Hardening

Since the server is to be deployed as a web server, Apache service was installed. In addition to securing the system, this section would discuss securing the server from application-layer attacks targeting the website.

5.5.1 Secure Socket Layer (SSL)

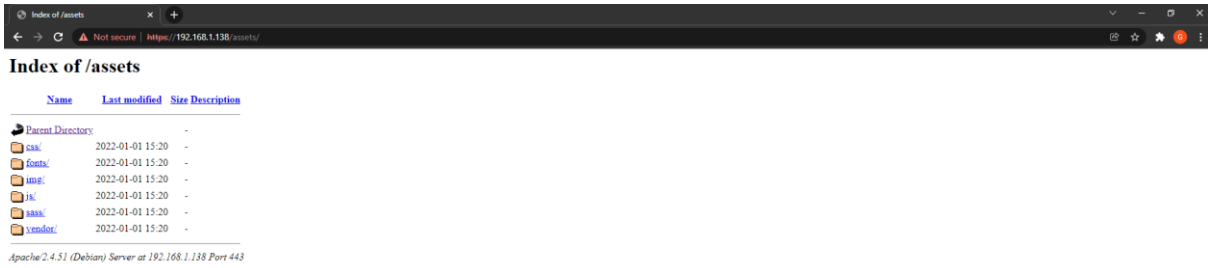
Websites need SSL certificates to keep user data secure, to verify ownership of the website, to prevent attackers from creating a fake version of the site, and to convey trust to users. Hence, SSL must be implemented on the web server. If the website requires user authentication or entering personal details such as credit card numbers or the viewing of sensitive information such as health benefits or financial information, then it is essential to keep the data confidential. SSL certificates help keep online interactions private and assure users that the website is authentic and safe to share private information. HTTPS is the secure form of HTTP, which therefore means that HTTPS websites have their traffic encrypted by SSL and ensure secure communication between client and server.

In this case, a full web application was not within the server but instead a static placeholder website. Also, in generating the SSL certificate, the certificate generated and used is a self-signed certificate and not generated by a Certificate Authority (CA). However, it still encrypts data and has an acceptable level of security. In the future, a CA issued certificate would be preferred as it proves that the identity of the certificate holder is trusted and verified by a third-party entity. It is also configured to redirect HTTP request to HTTPS to prevent users from accessing the website in an insecure manner.



5.5.2 Directory Listing

When directory listing is enabled, it poses a potential threat as it allows access to complete directory contents. If this option is enabled, an attacker can simply discover and view any files within the webserver directory. This could potentially lead to the attacker decompiling and reverse engineering an application in order to obtain the source code. They can then analyze the source code for possible security flaws or obtain more information about an application, such as database connection strings and passwords to other systems. As seen below, since directory listing is enabled by default, folders and files can be viewed without authorization by modifying the URL with a common subfolder name like “assets”.



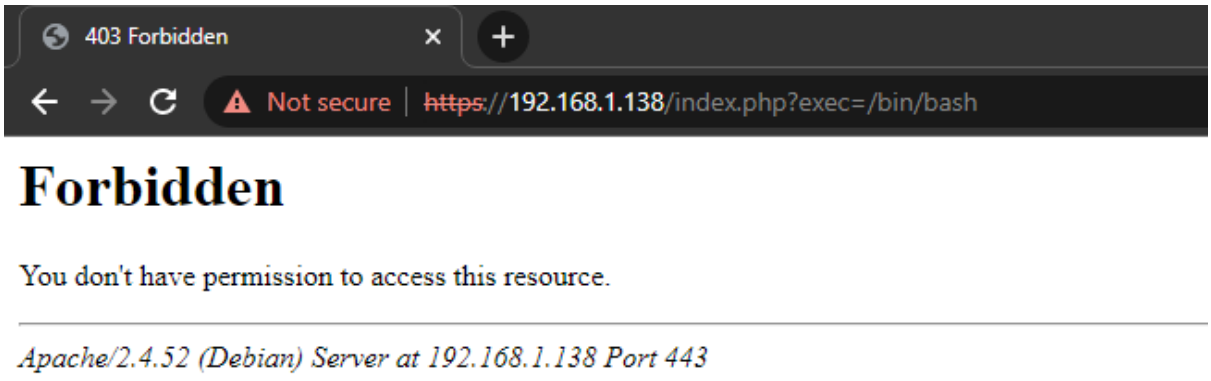
To mitigate this threat, directory listing is disabled from the web root folder onwards. This would prevent attackers from modifying the URL and attempt to access files within the system by force-browsing techniques which attempts to enumerate system resources.

5.5.3 Web Application Attack Protection

To protect the system from web application attacks, a Web Application Firewall (WAF) is implemented. Mod Security is a free WAF that works with Apache. It supports flexible rule engines to perform simple and complex operations. It works by inspecting requests sent to the webserver in real-time against a predefined rule set, preventing typical web application attacks like Cross-Site Scripting (XSS) and SQL Injection from occurring.

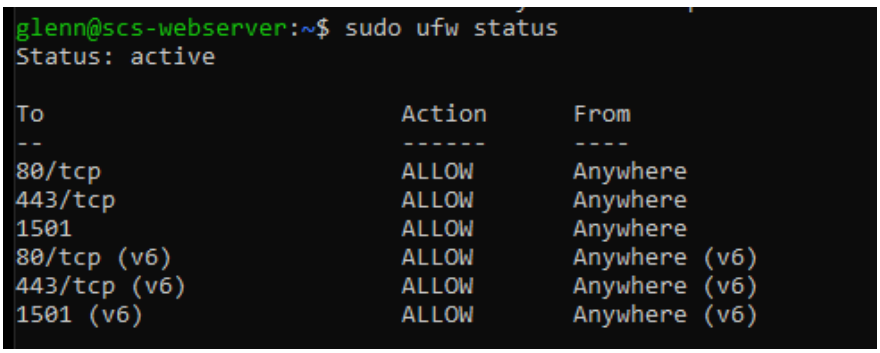
For this server, it would be configured with the OWASP ModSecurity Core Rule Set (CRS). It is a set of generic attack detection rules for the use with ModSecurity or compatible web application firewalls. The CRS aims to protect the web server from a wide range of attacks, including the OWASP Top Ten, with minimal false alerts.

As seen below, a malicious payload was appended to the URL and the requests were blocked as a command injection attack to execute a shell on the server-side was detected by the ModSecurity. In this case, PHP was not used on the server, but if it was and without the presence of this WAF, an attacker would be able to gain remote access to the web server and launch remote code execution attacks. However, based on the result, it is shown that ModSecurity is effective in detecting and preventing malicious attacks targeting the application layer.



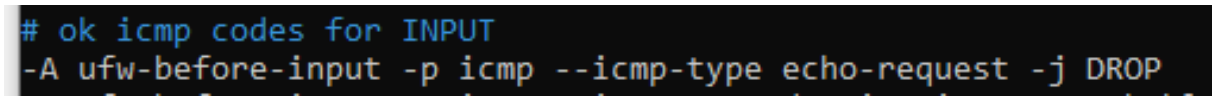
5.6 Host-Based Firewall

To enhance the security of the server, host-based firewall is used to prevent unauthorized access. Using Linux iptables to keep track of incoming, outgoing, and forwarded practices can secure the server by configuring “allow” and “deny” rules to accept or send traffic from specific IP addresses. This restricts the unchecked traffic movement on the server. Uncomplicated Firewall (UFW) is a netfiltering firewall designed to interact with iptables for configuration. For the server, only TCP port 80 (HTTP), TCP port 443 (SSL), and port 1501 (SSH) are opened. UFW was implemented to only allow specific ports to be opened and implicitly deny every other port to reduce the attack surface and prevent threat actors from gaining access to unnecessarily opened ports.

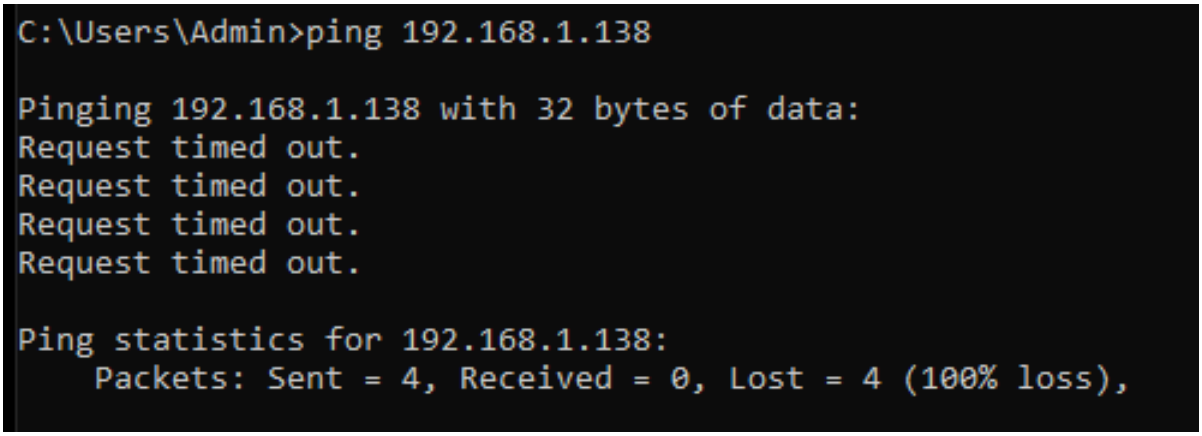


As seen below, since the server was configured with limited services, only the needed ports are active and listening. Unnecessary protocols and services such as SMTP and FTP are not present on the web server.

Moreover, it is also good practice to block Internet Control Message Protocol (ICMP) traffic. Thus, an explicit rule is created on UFW to drop ICMP echo requests. ICMP allows internet hosts to notify other hosts about errors and helps system administrators in troubleshooting. However, ICMP can also be exploited by adversaries to gain information about attacked networks. When ICMP is enabled, malicious attacks including network discovery, covert communication channels, and network traffic redirections can be executed. Hence, it is blocked to prevent attackers from potentially causing Denial of Service to the web server and prevent attackers from knowing the status of the web server.



As seen below, Ping request to the web server is unsuccessful due to this protocol being explicitly blocked on UFW.



5.7 Other Controls Implemented

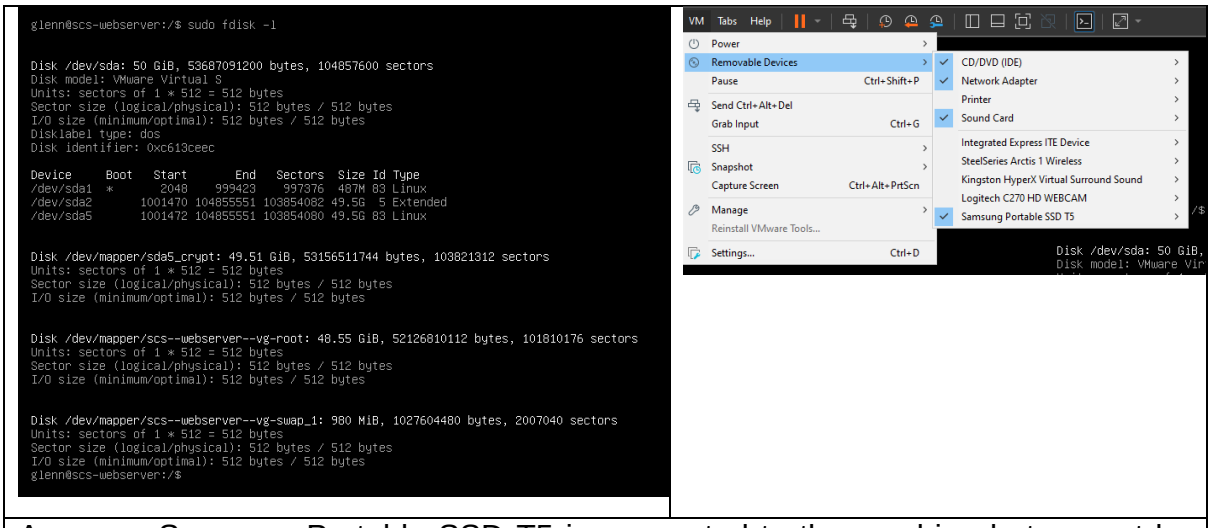
Linux Debian is built to be stable and secure. But further configurations must be applied to the default configurations in addition to installing additional security to harden the server and enhance security. This section will address the configurations performed to further secure the web server.

5.7.1 Boot Loader Security

In addition to encrypting the hard drive, the inclusion of a GRUB loader password adds an extra layer of security from physical threats. With this implementation, the server cannot be booted without first entering a password at the GRUB menu. However, there is a risk with this that is similar to disk encryption. The secrecy of the password and strength is a factor in securing the physical drive. Nonetheless, this provides multi-layered security as in order to fully compromise and access the contents of the drive, both the GRUB loader password and decryption password must be obtained. Moreover, this also prevents attackers from editing any GRUB entries or passing arguments to the kernel from the GRUB command line without entering the password. For this server, the GRUB password is “vUVyM?2!”.

5.7.2 Disable External Media

Booting from unauthorized external devices can allow attackers to bypass the security of the system by booting the Operating System from their external device. Allowing external media can also result in confidential data being extracted from the server if an attacker manages to gain physical access to the server. To prevent potential threats from occurring, drivers like USB storage and firewire storage should be disabled when not used. It can also prevent unauthorized storage or data theft. This can be achieved by creating a file /etc/modprobe.d/ to blacklist the stated modes.



As seen, Samsung Portable SSD T5 is connected to the machine but cannot be detected by the server when drives are listed.

5.7.3 Increase Hashing Rounds

When a user is created with a password, the password is hashed and stored in the /etc/shadow file. To enhance the security of the password in the event that the shadow file is compromised, the number of hashing rounds can be modified and varied. In this case, the rounds are set to between 5000 to 99999 rounds. This makes the password harder to crack for an attacker. If this configuration is not hardened and an attacker managed to obtain the shadow file, passwords can be easily cracked and disclosed due to being only hashed once. Hence, the number of rounds the password is hashed should be a large number of times and varied among different users.

```
# Define the number of SHA rounds.
# With a lot of rounds, it is more difficult to brute forcing the password.
# But note also that it more CPU resources will be needed to authenticate
# users.
#
# If not specified, the libc will choose the default number of rounds (5000).
# The values must be inside the 1000-999999999 range.
# If only one of the MIN or MAX values is set, then this value will be used.
# If MIN > MAX, the highest value will be used.
#
SHA_CRYPT_MIN_ROUNDS 5000
SHA_CRYPT_MAX_ROUNDS 99999
```

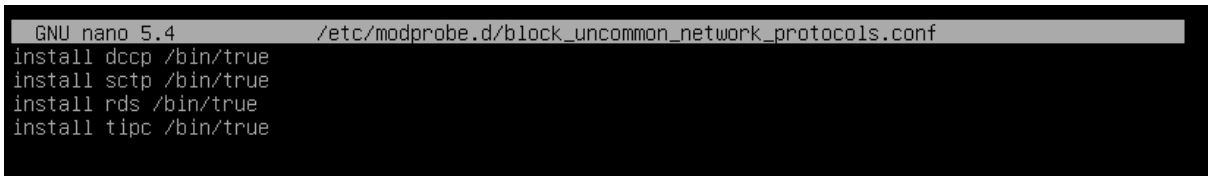
5.7.4 Logon Banner

One security control type that can be implemented to the server is a deterrent. This control type psychologically discourages an attacker from attempting an intrusion. Logon banners have been a common feature of operating systems and applications for many years. Organizations have adopted logon banners to threaten unauthorized users with severe repercussions. Thus, a logon banner was included in the local logon as well as the SSH logon.

```
C:\Users\Admin>ssh glenn@192.168.1.138 -p 1501
###############################################################
#                 This is a private server!                   #
#       All connections are monitored and recorded.           #
#  Disconnect IMMEDIATELY if you are not an authorized user!  #
###############################################################

Linux scs-webserver 5.10.0-10-amd64 #1 SMP Debian 5.10.84-1 (2021-12-08) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sat Jan  1 15:41:51 2022 from 192.168.1.214
glenn@scs-webserver:~$
```

5.7.5 Blacklist Uncommon Network Protocols

There are some network protocols supported by the Linux kernel modules that are rarely used and are likely to contain unknown vulnerabilities. These protocols are unreliable and have shown to contain severe vulnerabilities and hence should be disabled to reduce the attack surface. Blocking of these protocols will be done through creating a file in modprobe to blacklist the protocols as seen below.



6 Logging and Monitoring

Keeping detailed logging and auditing enabled for servers is crucial. These logs can later be used to detect any attempted intrusions. Also, in case of intrusion, these logs will help to gauge the extent of the breach and offer insight into the intent of the attacker. In the future, log files from the /var/log/ can be forwarded to a SIEM or Log Manager for ease of monitoring. However, due to the absence of a centralized log collector, logs must be reviewed manually. Below is the file path to each log file the log monitoring personnel must review regularly. It is critical that these files are routinely checked and analyzed as without proper monitoring, logging is futile.

Type	Path
Authentication	/var/log/auth.log
Apache Access	/var/log/apache2/access.log
Apache Error	/var/log/apache2/error.log
Audit Files	/var/log/audit/audit.log
Fail2Ban	/var/log/fail2ban.log

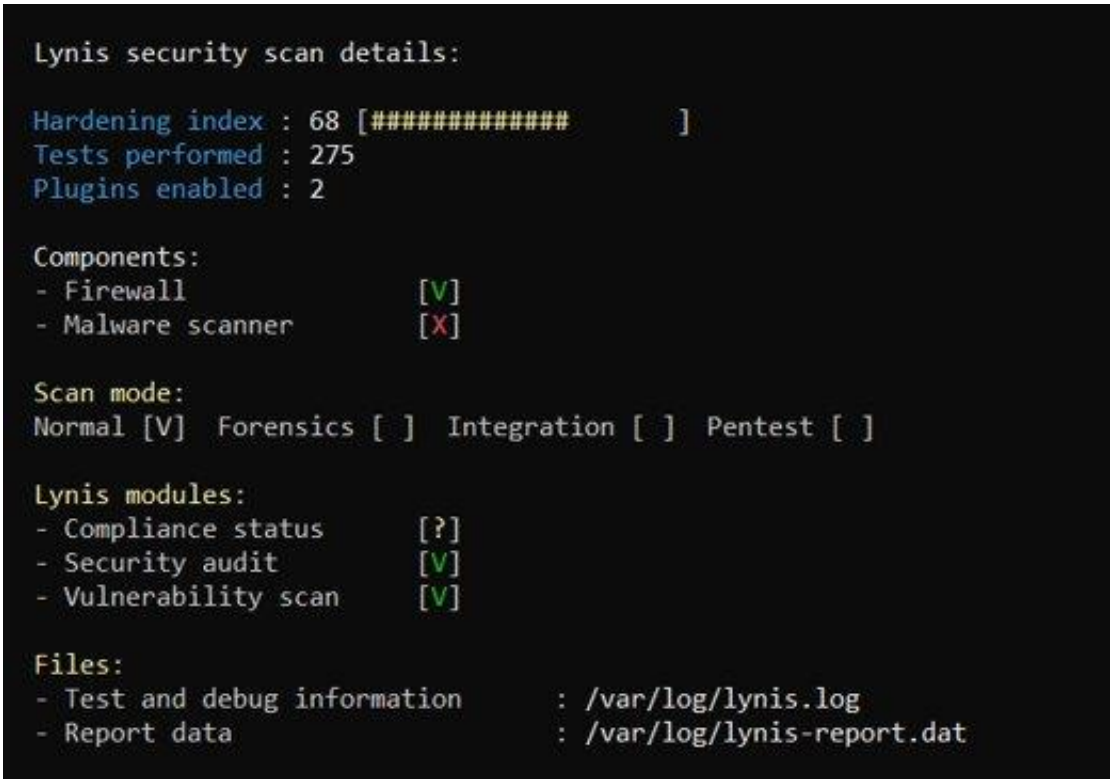
7 Security Testing

Vulnerabilities and misconfigurations on the server can be detected using automated tools. These tools are used to verify whether the existing security measures are effective as well as detect potential vulnerabilities on the server.

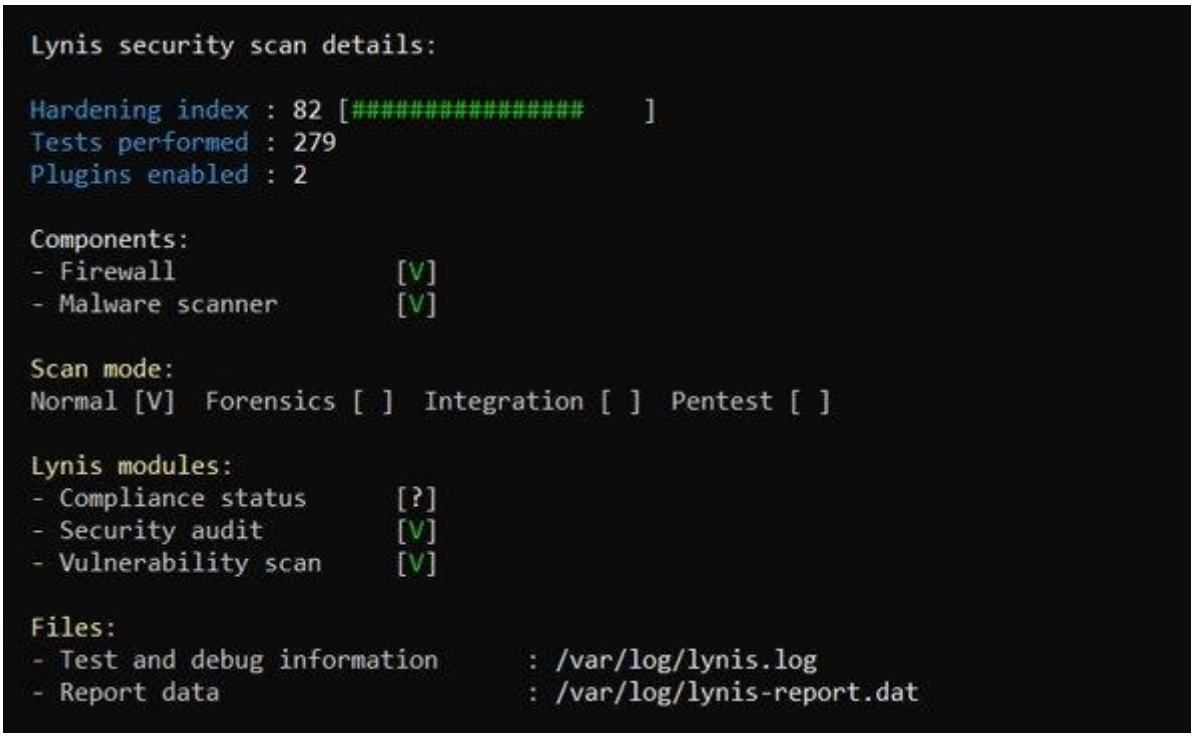
7.1 Security Auditing

Lynis is a security tool to perform an extensive “health” scan of the server to support system hardening compliance testing. It ensures that the best practices for implementation are in place by checking the available modules and their program details. It is able to offer a customized checking system in this sense, as it does not follow any particular guideline, but instead focuses on the best practices for each of the tools implemented.

As seen below, it is the result of the first scan after initial implementations are configured such as installation of Apache, SSH, and password policies. Based on the results, the hardening index indicates the security scale of the server which is 68%. However, based on the suggested guidelines provided in this scan, the recommended actions were taken and discussed in this report such as the inclusion of GRUB loader password, disable external media, and blacklisting of uncommon network protocols.

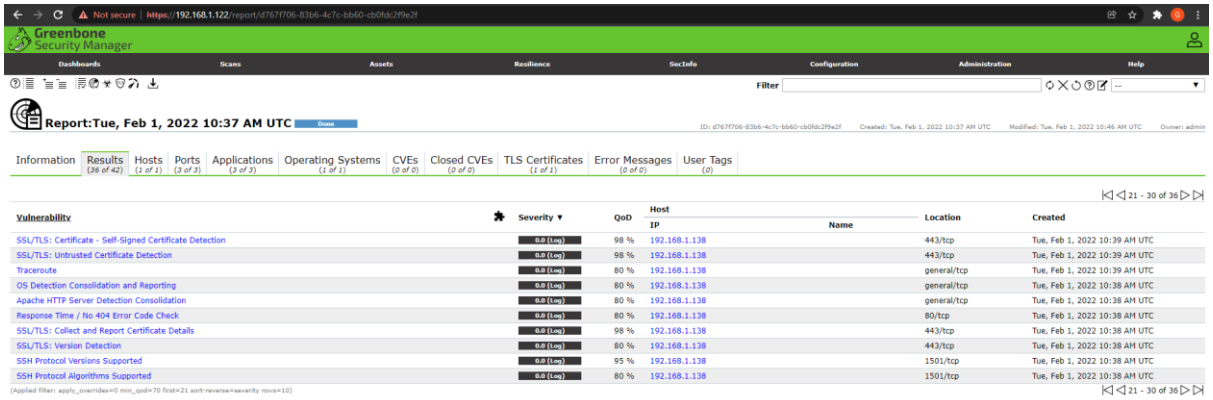


After the necessary recommended actions by Lynis are implemented, a second scan was conducted to verify if the misconfigurations are corrected. Based on the scan result, the server hardening index was significantly improved to 82% and hardened to a certain extent. However, there are still existing concerns that may be potential issues in the future and should be monitored regularly.



7.2 Vulnerability Scanner

Another tool used for vulnerability scanning was Greenbone Security Manager (GSM). GSM will be used for further vulnerability scanning of the web server. Its capabilities include unauthenticated and authenticated testing, various high-level and low-level internet and industrial protocols, performance tuning for large-scale scans and a powerful internal programming language to implement any type of vulnerability test. Based on the result, there are no potential threats and vulnerabilities detected on the server. However, the server should be regularly scanned at least once every week with GSM updated to detect new vulnerabilities that were not detected as of the moment of performing the scan.



7.3 System Scanning

System scanning with nmap was performed to check if the respective firewalls and ports were set correctly earlier on. As this is a web server, port 80 and 443 is open for HTTP and HTTPS respectively. As mentioned earlier, the port for SSH has been reassigned to the non-standard port which is 1501 and is reflected by nmap accordingly. This scan verified that there are no unintentionally opened ports that an attacker can use to leverage an attack.

```
┌──(glenn㉿kali)-[~]
└─$ sudo nmap -A 192.168.1.138
Starting Nmap 7.91 ( https://nmap.org ) at 2022-02-05 14:32 +08
Nmap scan report for 192.168.1.138
Host is up (0.00016s latency).
Not shown: 997 filtered ports
PORT     STATE SERVICE  VERSION
80/tcp   open  http     Apache httpd 2.4.52
|_http-server-header: Apache/2.4.52 (Debian)
|_http-title: Did not follow redirect to https://192.168.1.138/
443/tcp  open  ssl/http Apache httpd 2.4.52 ((Debian))
|_http-server-header: Apache/2.4.52 (Debian)
|_http-title: Revetive - Free Business Corporate Template By MACode ID
| ssl-cert: Subject: commonName=192.168.1.138/organizationName=NP/stateOrProvinceName=Singapore/countryName=SG
| Not valid before: 2022-01-14T03:05:09
|_Not valid after:  2023-01-14T03:05:09
| tls-alpn:
|_  http/1.1
1501/tcp open  ssh      OpenSSH 8.4p1 Debian 5 (protocol 2.0)
| ssh-hostkey:
|   3072 d0:71:9c:3a:9a:2d:fc:01:a3:a1:b6:c5:d3:09:52:72 (RSA)
|   256 aa:10:f7:2c:db:c8:43:a3:ff:c8:84:0d:6c:32:6d:f8 (ECDSA)
|_  256 40:ae:41:69:6b:c8:08:29:86:a5:f5:d2:02:ad:eb:08 (ED25519)
MAC Address: 00:0C:29:FB:F3:2E (VMware)
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Device type: general purpose
Running: Linux 4.X|5.X
OS CPE: cpe:/o:linux:linux_kernel:4 cpe:/o:linux:linux_kernel:5
OS details: Linux 4.15 - 5.6, Linux 5.0 - 5.4
Network Distance: 1 hop
Service Info: Host: 127.0.1.1; OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

8 Conclusion

Overall, through the security controls implemented discussed in this report, the attack surface of the Linux web server would be drastically reduced. Throughout the server, security principles such as defense-in-depth for multi-layered security, least privilege in account management, and leveraging existing components have been enforced to ensure that security within server is as strong enough to mitigate and withstand against most attacks. In addition, the installation and configuration of security tools enhances the security level such as responding to suspected intrusions, detection of malware, and with the inclusion of a firewall to restrict unauthorized access or misuse through unnecessary ports. Thus, the web server has been hardened as well as at the application level.

However, after the web server has been deployed, continuous and regular maintenance, security auditing, and vulnerability scanning must be performed as there might still be vulnerabilities that slipped through and may be discovered long after the server is deployed. Hence, these security issues need to be patched and corrected as soon as they are detected and verified again to further increase the security of the server. Despite the efforts made to ensure the server is secure, there is always a need for constant checking and updates to maintain the highest level of security. Additionally, new tools can always be implemented if needed to adapt to new technologies.