FACULTY OF SCIENCE, ENGINEERING AND COMPUTING

School of Computer Science and Mathematics

BSc (Hons) DEGREE

IN

Cyber Security & Computer Forensics

Jason Antwi

K1911178

Analyse And Develop A Secure Webserver

3rd May 2023

Liang Xing

Kingston University London

Declaration

I have read and understood the University regulations on plagiarism, and I understand the meaning of the word *plagiarism*. I declare that this report is entirely my own work. Any other sources are duly acknowledged and referenced according to the requirements of the School of Computer Science and Mathematics. All verbatim citations are indicated by double quotation marks ("…"). Neither in part nor in its entirety have I made use of another student's work and pretended that it is my own. I have not asked anybody to contribute to this project in the form of code, text, or drawings. I did not allow and will not allow anyone to copy my work with the intention of presenting it as their own work.

Date: 3rd May 2023

Name: Jason Antwi

Signature: 

Table of Contents

Table of Figures

Glossary of Terms

Term	Meaning
Agile project management	A project management methodology
Apache	An open-source web server software
Cybercriminal	Someone that uses the internet to commit crimes such as stealing personal data
Cybersecurity	The act of protecting digital systems, networks and data from unauthorised access.
Data breach	Access of unauthorised data such as sensitive data of people.
Fail2ban	A software that can protect web servers from brute-force attacks.
Hotlink protection	A tool that helps prevent direct links of a website's file.
Intrusion Detection	A security system that monitors the traffic of a network and alert users of any suspicious activity spotted on their network.
LAMP Stack	A bundle software that includes Linux, Apache, MySQL and PHP.
Let's Encrypt	A free, automated, and open certificate authority service that helps in enabling HTTPS(SSL/TLS) encryption for websites by providing digital certificates.
OpenVAS	A tool that can scan a web server for vulnerabilities.
Password policy configuration	A software tool that involves setting up specific rules and requirement for password on the server to ensure they are strong enough to not be easily compromised by an attacker.
Secure Shell (SSH)	A network protocol that allows secure communication between two systems.
Secure Socket Layer (SSL)	A protocol for ensure that data transmitted over a network is secure and cannot be intercepted by an attacker by encrypting the communication between the web server and the client
Two Factor Authentication	A security protocol that forces to verify their identity with two different factors before gaining access.

Unattended updates	A tool that ensures a server is always up to date in terms of security patches and updates
Virtual machine	A software emulation of a physical computer system
VirtualBox	A software that allows for multiple virtual machine on a single physical machine
Web application firewall (WAF)	A tool helps protect web applications from malicious attacks.
Webserver	A computer system that provides web pages and other services to users.

Introduction and Background

Ensuring the security of web server applications and data from unauthorised access, data theft, and cyber-attack is very important in today's digital age. Businesses all over the world use web applications to provide services to their consumers; therefore, securing the webserver of such an organisation is a top priority to prevent data loss and many security breaches that could negatively impact the company's reputation or cause severe financial damage. Christoffer et al. (2021) explain that a web server is a software application that provides access to specific functionalities or services through the internet, allowing users to access the web server through a web interface without the need to install any software.

Web servers have become a primary target for cybercriminals due to the increasing dependence on the Internet for communication, commerce, and other important services. Therefore, this project aims to address the security concerns of web servers. It is extremely important to ensure web servers are adequately secured against potential attacks because a successful attack on a web server could have severe consequences such as data breach, theft, service disruption and downtime. This project will explore different tools and software that can enhance and strengthen web servers' security. Each tool and software will be evaluated to identify its effectiveness in mitigating security risks.

A study conducted by Rundle (2022) demonstrated that due to the high stakes and potential consequences of mistakes, defenders experience a high level of stress when fixing web server security issues. This shows the number and frequency of webserver attacks and the scale of the problem. The cost of a successful attack on a web server is significantly high; hence this project will contribute to preventing attacks by exploring and testing the different tools and software that stop attacks on web servers. Traditional security measures may no longer be enough to secure web servers as technologies are advanced and attacks and becoming increasingly sophisticated.

There are several stakeholders that can relate to this project. Website owners and operators may benefit from this project for protecting their users' sensitive data and ensuring their web server security. The end-users may also benefit from this project since they expect web-based services and applications to be protected and secure against attacks and rely on them. Cybersecurity professionals and IT experts that aim to ensure web servers are adequately protected against potentials and are responsible for maintaining and securing web servers are other stakeholders of this project. This project may also benefit from policymakers and regulators that aim to establish and enforce web server security and protection regulations.

Contributing to improving the web server's security is the primary purpose of this project. The aims of the project are as follows:

- Evaluate the most effective tools and software that further secure webservers.
- Provide insights and recommendations of webserver configurations to enhance their security that could be applied to other webservers.
- Contribute to the worldwide effort to protect the sensitive data of users of web-based services and application.
- Emphasise the importance of a secured web server and encourage website owners to ensure they are secure from potential threats.

The objectives were designed to allow this project to deliver the project aims within the time constraints. The objectives are as follows:

- Thoroughly research the current information related to web server security.
- Investigate different tools and software that can be used to enhance the security of web servers.
- Evaluate the most effective tools and software that further secure web servers.
- Develop and design a webserver and implement several security tools and software identified from the literature review.
- Configure the webserver to make it secure to meet industry standards in security practices.
- Test the developed web server and evaluate the effectiveness of the security tools and software implemented.
- Document the development and testing stages of the webserver
- Provide a clear recommendation for enhancing web server security.

The project's scope is to focus on the commonly used webservers for hosting web-based services and applications such as Apache and investigate different tools and software that can be used to further improve the security of the webservers. Webservers being the main target for cyberattacks is one factor that justifies this project's scope, as enhancing security is vital to protect users' sensitive data. Another factor that justifies the project's scope is the wide usage of web servers, representing a considerable portion of the internet infrastructure, making web servers a very important component of our system.

This project did not require ethical approval from the university; however, ethical, legal and social impacts are associated with the project. By enhancing the security of web servers, the project helps to prevent data breaches and cyberattacks, therefore, addressing legal concerns. From a legal point of view, the project will comply with relevant laws and regulations relating to data protection and web server security despite processing dummy data. In terms of social impacts, individuals and organisations can benefit from this project as it highlights ways to enhance the security of web servers.

Project Management Strategy

The agile project management methodology was the technique used to manage this project. Per Chia, Tung and Yong's (2022) discussion about the flexibility of Agile project management and allowing for changes to be made throughout a project lifecycle, this management methodology allowed for flexible and iterative decision making which made it easy to decide between which tool and software will best suit the webserver and improve the security of it throughout the project's lifecycle. Gaborov and Ivetić (2022) elaborated on this and stated that agile provides an iterative approach allowing for continuous improvement and adaptation to improve the project at hand further.

To ensure the webserver was secure, security measures and requirements were incorporated and assessed throughout the development project's lifecycle, and regular testing was conducted during each iteration. In terms of visualising the project timeline, the Gantt chart tool was used with the agile approach. This helped in understanding the project's progress and staying on track with the project goals. Figure I is a Gantt chart that outlines the project timeline. This allowed potential delays or roadblocks to be identified, which could be properly tackled to ensure the project stays on track.

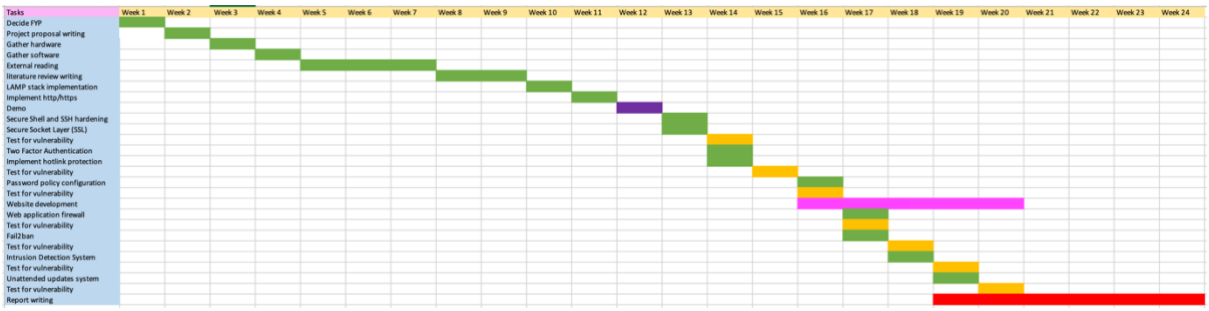

Figure i: Gantt Chart

The Gantt chart illustrates all the project stages, from the hardware and software gathering to the report-writing phase. Implementation phases and testing phases overlapped with one another due to time constraints. Nonetheless, the Gant chart accurately shows the tasks and timeline required to develop this project.

There are alternatives to agile project management; the waterfall methodology could have been used to manage this project. The waterfall is a straightforward and simple methodology, making planning and managing projects easier. According to Nicula and Ghimișu (2019), waterfall provides a well-defined project structure with clear phases and deliveries, which makes projects easy to manage. This allows for better control over the project since each phase is completed before moving on to the next. However, the waterfall can be difficult to make changes once a phase has been completed because it is flexible in its design. This would not have suited the project since the main project goal was to find security tools and techniques and revolved around going back and forth to find ones that better improve the security of the web server. Another alternative methodology that could have been used is Kanban. This methodology focuses on visualising workflow and making work visible, which increases communication and collaboration, according to Shamshurin and Saltz (2022). This allows for identifying and addressing inefficiencies in the process more quickly.

Similar to the methodologies, alternative tools could have been used in this project. According to Evdokimov et al. (2018), the PERT project management tool analyses tasks necessary for project implementation by breaking down a project into smaller tasks and identifying their dependencies to create a network diagram. This provides a clear visualisation of the project's schedule and helps spot problems that may require additional resources.

Literature Review

This section of the report is a literature review that will discuss the numerals systems, techniques, applications and tools that can be used to secure web servers. This literature review will discuss the benefits and limitations of LAMP stack, Secure Shell (SSH) and SSH hardening, Secure Socket Layer (SSL), OpenVAS, Fail2ban, Two Factor Authentication, Password policy configuration, Intrusion Detection System, Unattended updates system, Web application firewall, Hotlink protection and Let's Encrypt as shown in Figure ii. The literature review displayed each benefits and limitations in a table format to make it clear, organised, and easy to read.

Tool/software	**Benefit**	**Limitations**
LAMP Stack	Max (2017) stated that LAMP stack is one of the most well-known open-source software for web development because of its flexibility and user-friendliness. Max (2017) also explained that Linux provides a safe environment for the web application to run on, and the Apache aspect provides the web server with a wide range of security features and modules to enhance the overall security of the web server. To provide a secure to store and retrieve data, MySQL does just that, and for the server-side scripting language of the web application, PHP programming language provides that and more. Matt (2021) went on to state that the LAMP stack provides a stable and reliable platform for developers to develop on will making web development accessible to developers at all levels.	Louridas (2016) talked about one of the major limitations of the LAMP stack: it requires data to be stored as a relational database model, which limits users. Dhuny et al. (2022) also explained that the open-source nature of the LAMP stack makes it a security vulnerability. There is no centralized authority that is responsible for system updates and security patches.
SSH	Lucian (2019) commented SSH provides a high level of security because of its ability to encrypt transmitted data. Lucian (2019) also mentioned SSH allows users to access their servers worldwide through its secure remote access. Wang, Chen and Yu (2022) elaborated on this and discussed that SSH provides a secure channel for accessing and controlling servers and encrypts the data being transmitted, including passwords and other sensitive information, to prevent unauthorised access and cyber-attacks, further giving the server another layer of security.	Lucian (2019) explained that SSH could fall victim to a man-in-the-middle attack when an attacker intercepts data before it gets transmitted.
Secure Socket Layer (SSL)	According to David and Alex (2019), the SSL protocol makes transmitted data unreadable to eavesdroppers. This provides an additional layer of security for web servers.	Owoh and Mahinderjit Singh (2018) studies showed that SSL could be vulnerable to attacks and misuse if it is not

	Rama Devi et al. (2020) also informed that SSL provides privacy and safety in data transmission between web browsers and servers. It ensures that information is encrypted between the two parties, making it difficult for an attacker to eavesdrop and obtain the transmitted data.	implemented correctly. This can compromise the integrity and confidentiality of sensitive data transmitted over a network.
OpenVAS	Caponi and Leuti's (2019) studies showed that OpenVAS is a free, open-source tool that can be used for vulnerability assessment and detect vulnerabilities in all types of systems and applications, such as operating systems, web servers, and other networks. OpenVAS also provides a report of the vulnerabilities it found after scanning a network, which can be used to improve the system's security.	Xia et al. (2020) informed that OpenVas might not be reliable as it cannot detect all vulnerabilities, especially if the attacker is not included in the NVT plugin library. Xia et al. (2020) also stated that OpenVas consumes many system resources, such as memory and CPU, which could slow down the system.
Fail2ban.	Sandra (2022) noted that fail2ban monitors system logs and blocks suspicious IP addresses by placing them in "jail", helping protect the system from potential attacks and unauthorised access. Per Sandra's (2015) statement, fail2ban adds an extra layer of security to the system and helps block addresses that are suspicious of the system's security.	On the other hand, fail2ban relies heavily on log files to detect security threats; therefore, an attacker could exploit this by avoiding generating log entries.
Let's Encrypt.	The automated and cost-free nature of Let’s Encrypt makes it an advantage, according to Sujatanagarjuna, Bochem and Leiding (2022). Let’s Encrypt provides its users with a safer web experience, as sites without SSL certification are listed as unsafe and could potentially hurt a business.	Paul (2016), on the other hand, believes that Let's Encrypt has negative aspects, such as it allows attackers to generate their own legitimate SSL certificates to host malicious HTTPS sites or sign malicious codes. This puts users of such sites at risk of phishing attacks or malware infections. Paul (2016) also stated that certificates made by Let's Encrypt have a lifespan of 90, which means users will have to renew every 90 days making it a tedious process.

Password policy configuration	Gupta et al. (2023) studies showed that pluggable authentication modules can be used for tracking login activities securely. The Pluggable Authentication Modules (PAM) provide a way to authenticate users by configuring and defining various password policies to ensure stronger security against attackers. This can then be examined to look for potential security vulnerabilities. Gupta et al. (2023) mentioned that it could also be used to securely log attempts made to the server, which can then be monitored for security breaches or suspicious activity.	The PAM package might be difficult to set up and configure since it requires technical knowledge, making it challenging for non-technical users to implement. Secondly, PAM is not flexible, which limits it to a particular environment only.
Two Factor Authentication.	Xu et al. (2018) briefly stated that by requiring two different factors, two-factor authentication makes it extremely difficult for a user without unauthorised access to enter the system or gain access to sensitive data. It provides an additional layer of security and reduces the risk of data breaches, phishing and brute-force attacks on the server.	A limitation of two-factor authentication is the risk of being unable to generate TOTP codes when the phone is lost. Two-factor authentication relies on another device to generate TOTP, so it becomes impossible to get into the system if the device is lost or not around.
Intrusion Detection.	According to Kanna and Santhi (2021), an intrusion Detection System helps secure a network by differentiating malicious entries from legitimate ones in network traffic data. Tian, Li and Liu (2019) explained that intrusion detection guarantees secure communication on the server as it monitors the network activities and detects suspicious addresses interacting with the network.	Ashfaq et al. (2016) discussed that intrusion Detection does not necessarily guarantee network protection since attackers always look to disrupt networks with several attacks.
Unattended updates	There are several benefits of Unattended updates. It ensures the server is always up to date with the latest software version and security patches, which reduces the risk of vulnerabilities in the server. K.Buzdar (2017) stated that an unattended installation requires little user intervention since it installs automatically. This saves time and effort that would have been spent checking for weaknesses in the system and manually installing updates.	However, unattended updates come with several limitations. It can overwrite custom configurations or settings since its automatic. It can also disrupt ongoing processes and even cause downtime for users.
Web application firewall	Dawadi, Adhikari and Srivastava (2023) mentioned web application stands between the web application and client,	Dawadi, Adhikari and Srivastava (2023) discussed that some web

	acting as a barrier on the internet and guarding the application's vulnerabilities by detecting anomalous traffics and filtering harmful communications. Shaheed and Kurdy's (2022) studies showed that web application firewalls could help detect and block attacks that would have escaped traditional firewalls. Tekerek and Bay (2019) elaborated on this and stated WAF protect against different types of web-based attacks, such as SQL injection and cross-site scripting.	application firewall only works on specific attack and threats and does not work on zero-day attacks. Dawadi, Adhikari and Srivastava (2023) also stated that they could not properly protect web servers since they do not inspect HTTP packets in the application layer.
Hotlink protection	Implementing hotlink protection protects the website's assets from theft. This ensures the website owners have total control over their content and conserves bandwidth. It helps protect images on the website from image theft and prevent excessive bandwidth usage. Enabling hotlink protection ensures that only users with permission to access content can, and it does this by blocking requests specific for types from unauthorised users.	There are several limitation of this tool. Determined attackers may be able to bypass the hotlink protection, and hotlink protection could potentially block legitimate users, especially if they are not authorised.

Figure ii: Benefits and limitations of the tools and software

The project aims to provide a guide and set of recommendations to individuals and organisations on correctly developing and maintaining a secure web server. Developing a secure webserver by integrating different security tools and software given in this report is the focus and intention. The project will address the outcomes by identifying the specific tools and software based on the relevant literature review. Each tool and software are evaluated by their weakness and strengths to determine the suitable ones for the webserver properly, then implement the selected into the webserver. After implementation, the effectiveness of the tools and software will be tested to ensure they are working as expected.

The focus and intention address the broader problem by providing a practical solution to the security of web servers. A set of recommendations for developing and maintaining a secure web server is proposed. This enables individuals and organisations to enhance the security of their web servers and protect the web-based services from web-based attacks if they choose to implement the recommended security tools and techniques. By providing a comprehensive guide to developing and maintaining a secure web server, this project contributes to the broader research on web security and can be used as a reference for future research.

Methodology

Various security tools and software that help to enhance a web server's security, as per the literature review's findings, will be implemented into this webserver. Developing a secure web server will serve as an example and model for individuals and organisations looking to enhance their own web server's security as they could implement the appropriate security measures to protect against cyber threats and ensure the safety of their data from this project. The web server will also serve as a practice ground to determine which security tools and software are best suited for different types of web servers and for different levels of security needed. The project will provide insights into the most effective measures for securing web servers by utilising the various security tools and software to create a secure web server.

The finding of the literature review has had a significant influence on this project's artefact as it provided valuable insights into the various security tools and techniques that can help improve and enhance the security of web servers. The literature review helped identify the strengths and weaknesses of the different security tools and techniques for web servers. It also influenced identifying the security measures' limitations and potential vulnerabilities so a recommendation could be designed to address each limitation. The literature review informed the importance of maintaining a web server over time through updating and proper maintenance.

To develop a secure web server, there are several tools and software required. Figure iii shows the tools and software used in this project.

Tools/software	**Specification/ reasoning**
Laptop	The laptop will have an Intel of Core i5, RAM of 8GB and 256GB of SSD storage. This high-performance laptop enables it to smoothly handle resource-intensive tasks such as running two or more virtual machines.
Router	For managing network traffic between the server and the client machines and to ensure the machines always remain accessible.
Backup storage device	To ensure that data is backed up and can be quickly restored in case of a hardware failure and to minimise the risk of data loss.
Virtual Box	To host several virtual machines and isolate them while testing numerous security configurations.
Ubuntu_20.4.1_server_amd64	Operating system serving as the server machine to provide a stable, secure and versatile platform to develop the server.
Kali machine	Serving as the client machine for testing, evaluation and identifying potential security weakness.
LAMP stack	For the web server and database management, build a highly scalable and performant server that can be used to meet the aims and objectives.

Secure Shell tool	Ensure secure access to the server and manage the sever remotely without needing physical or direct access.
Secure Socket Layer security protocol	To encrypt transmitted data over the network and ensure sensitive information remains confidential and secure.
Fail2ban	Protect the server against brute-force attacks and other unauthorised access by proactively identifying and blocking potential security threats.
Two-factor Authentication	Authenticating users and adding an extra layer of security by protecting against common cyber threats.
Password policy configuration	To enforce strong password requirements and protect against common password guessing and brute-force attacks.
PAM package	To force users to create strong passwords and ensure accounts remains secure and protected against unauthorised access.
Snort	For monitoring the network traffic and detecting malicious activity to minimise data breaches and other cyber-attacks.
Unattended updates system	Ensure the server is always up to date with security updates and patches.
Web Application Firewall tool	Protect the web server from malicious attacks by monitoring and blocking suspicious requests to the server.
Hotlink protection software	To prevent direct links to the website's assets and protect assets from data theft and misuse.

Figure iii: Tools and Software used

Implementation Of Artefact

This implementation section will provide a step-by-step guide describing the necessary process a user must follow to reproduce a successfully secured web server. This involves the software and hardware needed, setup guidelines, and additional relevant information. This section will be written clearly and concisely while outlining each stage of the process in detail for readers to replicate efficiently. This will provide the information required to be enabled to set up and run a secured webserver without encountering any significant issues.

1. Environment set-up

For a secure webserver implementation, a laptop (pc), VirtualBox software and two virtual machines installed within the VirtualBox are needed. The Virtual box allows for creation of multiple isolated environments, called virtual machines, for testing and implementation. Each virtual machine created operates independently, and any configuration done on them does not affect the primary system. This means different configurations can be done and tested to find the best settings that increase the web server's security with zero risks of affecting the main system.

On the laptop or pc, install Virtual Box, and create two virtual machines, which is ubuntu_20.4.1_server_amd64.ovf as the server and Kali Linux as the client. Configure the virtual machines according to the system requirement for smooth operation.

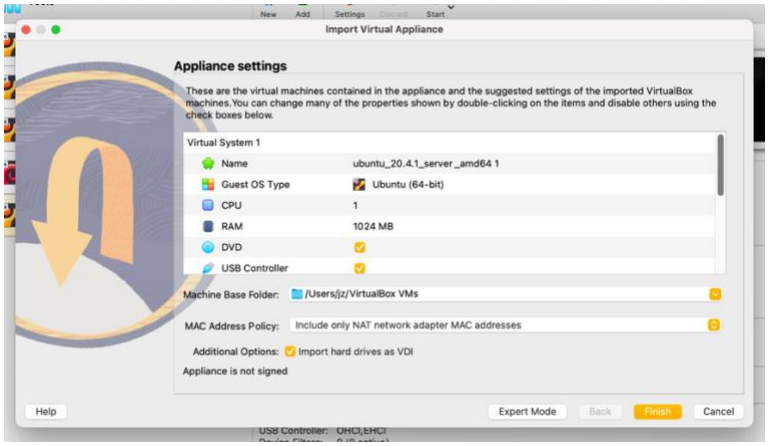

Figure iv: Ubuntu vm configuration

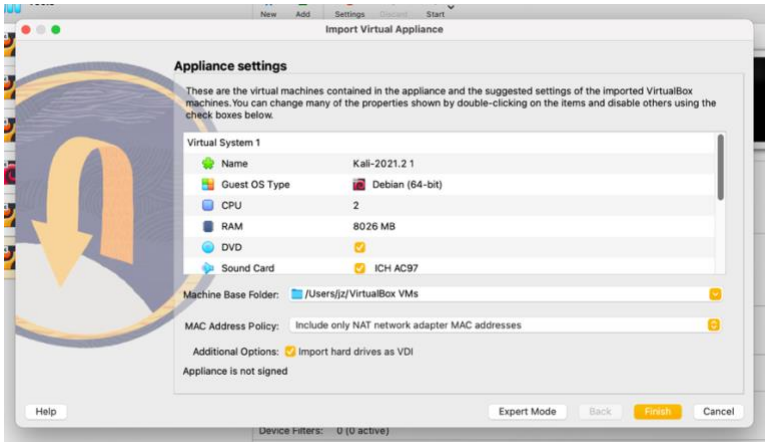

Figure v: Kali vm configuration

Now we have to set a static IP address on both virtual machines. Starting with the Ubuntu virtual machine, navigate to the netplan directory by entering “cd /etc/netplan” in the terminal. Then, open the configuration file “00-installer-config.yaml” with the nano text editor and enter using “sudo nano 00-installer-config.yaml” command. Set the static IP address inside the configuration file as shown in Figure iv, with the IP address preferred. After setting up the static IP, enter “sudo netplan apply” in the terminal to apply these changes.

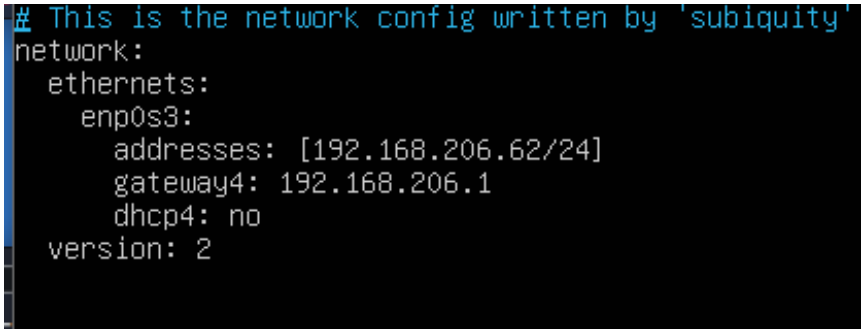

Figure vi: Ubuntu IP configuration

On the Kali machine, right-click the rectangular icon on the top bar, click on the bottom plus sign and select “Create”. Navigate to “IPv4 settings”, then add an IP address and save.

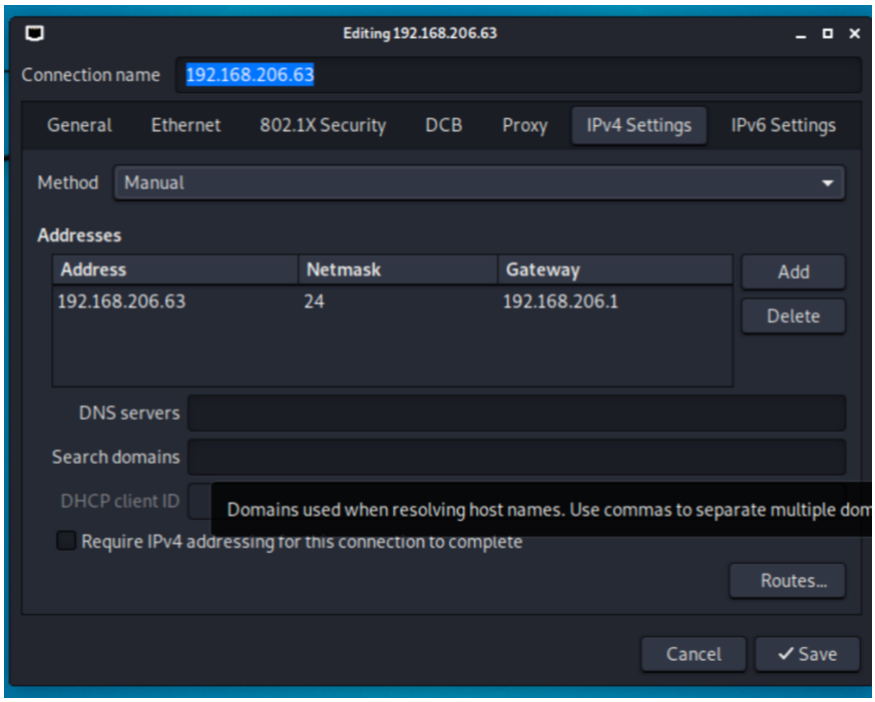

Figure vii: kali IP address configuration

2. Apache installation

There are several benefits to installing Apache on the server. Not only is Apache a free and open-source software, it also includes numerous modules that enhance the web server's functionality and

security of the web server. In terms of security, Apache is one of the most popular software that guarantees that. Apache provides several modules to protect communications between clients and the server and allow for access control to the server.

Before installing Apache on the server, the virtual machine must have access to the internet by configuring the server on "Nat" and disabling the static IP address. To confirm internet accessibility, enter the terminal "ping 8.8.8.8". To install Apache, first update the local package index with "sudo apt update", then install the apache2 package by entering "sudo apt install apache2". If a ufw firewall is set up, enter "sudo ufw allow "Apache Full" to open up the HTTP and HTTPS ports.

```
Enabling module mpm_event.
Enabling module authz_core.
Enabling module authz_host.
Enabling module authn_core.
Enabling module auth_basic.
Enabling module access_compat.
Enabling module authn_file.
Enabling module authz_user.
Enabling module alias.
Enabling module dir.
Enabling module autoindex.
Enabling module env.
Enabling module mime.
Enabling module negotiation.
Enabling module setenvif.
Enabling module filter.
Enabling module deflate.
Enabling module status.
Enabling module reqtimeout.
Enabling conf charset.
Enabling conf localized-error-pages.
Enabling conf other-vhosts-access-log.
Enabling conf security.
Enabling conf serve-cgi-bin.
Enabling site 000-default.
Created symlink /etc/systemd/system/multi-user.target.wants/apache2.service → /lib/systemd/system/ap
ache2.service.
Created symlink /etc/systemd/system/multi-user.target.wants/apache-htcacheclean.service → /lib/syste
md/system/apache-htcacheclean.service.
Processing triggers for ufw (0.36-6) ...
Processing triggers for systemd (245.4-4ubuntu3.2) ...
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for libc-bin (2.31-0ubuntu9) ...
user@jason:~$ sudo ufw allow "Apache Full"
Rules updated
Rules updated (v6)
user@jason:~$
```

Figure viii: Apache installation, http and https ports open up

3. MySQL installation and configuration

MySQL database management system provides several security benefits on web servers. MySQL allows for creating and managing user accounts and permissions on the server, and it also provides a secure authentication mechanism by verifying users before allowing access to the data. To install MySQL, first update the system with “sudo apt update”, then install MySQL with “sudo apt install mysql-server”. After installing MySQL, root can be used to enter, which is not very secure; therefore, the authentication method for root must change with a preferred password. To do this, enter MySQL with “sudo mysql” then in MySQL enter “alter user ‘root’@’localhost’ identified with mysql_native_password by ‘[preferred password here]’;” Now exit MySQL with “exit” and enter “mysql_secure_installation” to run the MySQL secure installation to make the installation more

secure. MySQL will prompt to enter the [preferred password] set in the previous step, then choose yes to use the validate password component, which makes it more secure.

Figure ix: MySql secure installation

Users will be prompt to choose how strong passwords must be, choose the strong option to make it secure.

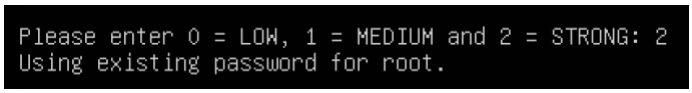

Figure x: MySQL password strength

MySQL will ask a series of questions such as “remove anonymous users”, “Disallow root login remotely”, and “Remove test database and access to it?” to strengthen the system. Answering yes to them increases security.

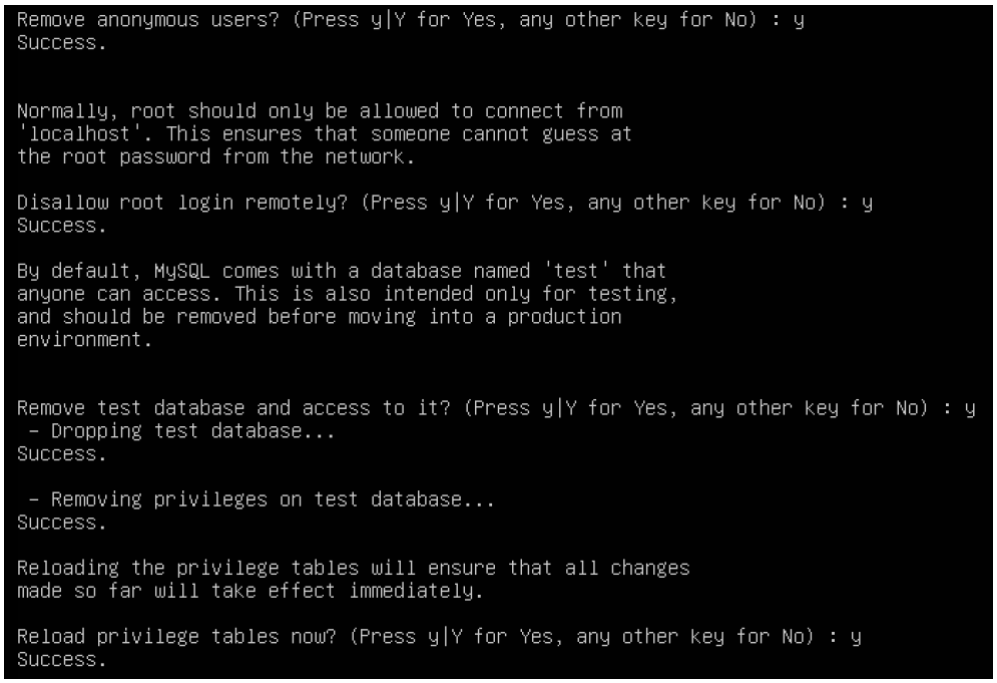

Figure xi: MySQL secure installation configurations

After the mysql secure installation configurations, trying to log into mysql with root by using “sudo mysql” will not work, instead this command “mysql -u -p” works by prompting users to enter password. This further makes the system secure since users are authenticated before allowing access into the database.

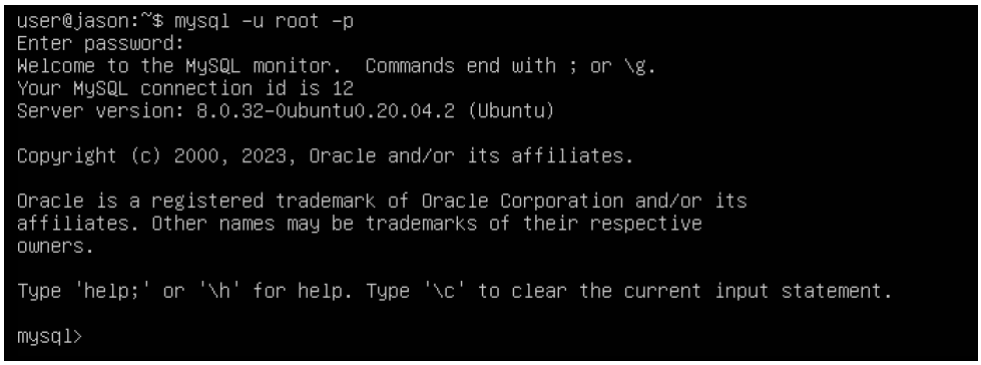

Figure xii: Secure MySQL access

4. PHP installation

PHP helps maintain the server's security as it is compatible with numerous security tools such as web application firewalls and intrusion detection systems. PHP being an open-source language means security patches and updates are received the moment they are released, guaranteeing the server's security.

To install PHP on the server use “sudo apt install php” then install phpMyAdmin as well with “sudo apt install phpmyadmin”. Users will be prompt to choose the webserver that should automatically be configured to run phpMyAdmin, apache2 will be selected here.

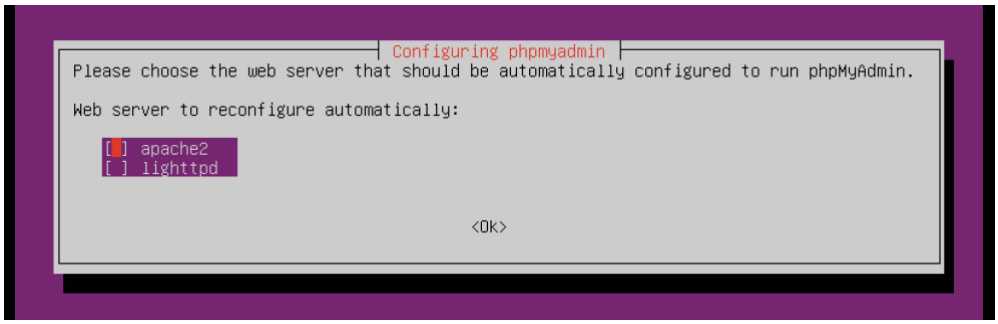

Figure xiii: PHP Installation configuration

The installation will ask about auto configuration to the databases, “no” will be selected here.

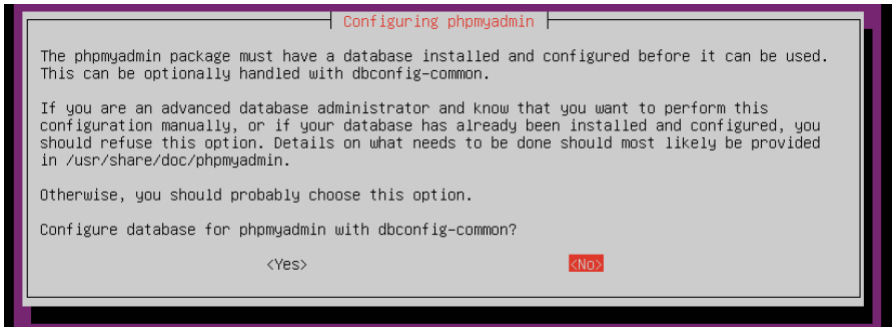

Figure xiv: phpamyadmin web server auto configuration

5. Self-Signed Certificate

SSL improves the security of the web server. SSL enables HTTPS encryption, ensuring that communications between the clients and the server are encrypted and secure. SSL also provides a level of trust as it ensures users communicate with the intended server. To create a Self-Signed Certificate on the Apache webserver and secure communication between the client and the webserver, an Apache module called “mod_ssl” will be used to enable SSL encryption. This module is enabled by using the a2enmod command and then restarting Apache.

```
user@jason:~$ sudo a2enmod ssl
[sudo] password for user:
Considering dependency setenvif for ssl:
Module setenvif already enabled
Considering dependency mime for ssl:
Module mime already enabled
Considering dependency socache_shmcb for ssl:
Enabling module socache_shmcb.
Enabling module ssl.
See /usr/share/doc/apache2/README.Debian.gz on how to configure SSL and create self-signed certifica
tes.
To activate the new configuration, you need to run:
  systemctl restart apache2
user@jason:~$ sudo systemctl restart apache2
user@jason:~$
```

Figure xv: a2enmod command and Apache restart

After enabling mod_ssl, a self-signed SSL certificate needs to be generated with the help of the openssl command. This command will create a key file and a certificate file containing the website information such as the hostname and IP address. Thew certificate uses DSA algorithm with a key length of 2048 bits and is valid for one year. The certificate creation process will prompt users to enter hostname or IP address to access the website. For this example, we will use “gojogym.com” as hostname.

```
user@jason:~$ sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/ssl/private/apac
he-selfsigned.key -out /etc/ssl/certs/apache-selfsigned.crt
Generating a RSA private key
........+++++
.....................................................+++++
writing new private key to '/etc/ssl/private/apache-selfsigned.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:UK
State or Province Name (full name) [Some-State]:london
Locality Name (eg, city) []:london
Organization Name (eg, company) [Internet Widgits Pty Ltd]:gojogym.com
Organizational Unit Name (eg, section) []:gojogym.com
Common Name (e.g. server FQDN or YOUR name) []:gojogym.com
Email Address []:k1911178@kingston.ac.uk
user@jason:~$
```

Figure xvi: SSL key file and certificate file creation

The next step is to configure Apache to use the certificate and key files created. In the sites-available directory, create a new Apache configuration file specifying the ServerName and DocumentRoot directories.

```
user@jason:~$ sudo nano /etc/apache2/sites-available/gojogym.com.conf
```

Figure xvii: creating a new file in sites-available

Add the necessary SSL options to point to the certificate and key files.

```
  GNU nano 4.8                /etc/apache2/sites-available/gojogym.com.conf
<VirtualHost *:443>
   ServerName gojogym.com
   DocumentRoot /var/www/gojogym.com

   SSLEngine on
   SSLCertificateFile /etc/ssl/certs/apache-selfsigned.crt
   SSLCertificateKeyFile /etc/ssl/private/apache-selfsigned.key
</VirtualHost>
```

Figure xviii: specifying the ServerName and DocumentRoot with necessary SSL options

Next is to enable the configuration file with the a2ensite tool

```
user@jason:~$ sudo a2ensite gojogym.com.conf
[sudo] password for user:
Enabling site gojogym.com.
To activate the new configuration, you need to run:
  systemctl reload apache2
```

Figure xix: using a2ensite tool to configuration file

Test for any error in the configuration then reload Apache

```
user@jason:~$ sudo apache2ctl configtest
AH00112: Warning: DocumentRoot [/var/www/gojogym.com] does not exist
AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 192.1
68.1.153. Set the 'ServerName' directive globally to suppress this message
Syntax OK
user@jason:~$ sudo systemctl reload apache2
user@jason:~$
```

Figure xx: error testing and Apache reload

To redirect requests from HTTP to HTTPS since the current configuration only responds to HTTPS requests on port 443, another VirtualHost block needs to be added to match requests on port 80 in the .conf file created in sites-available directory.

```
  GNU nano 4.8                /etc/apache2/sites-available/gojogym.com.conf
<VirtualHost *:443>
   ServerName gojogym.com
   DocumentRoot /var/www/gojogym.com

   SSLEngine on
   SSLCertificateFile /etc/ssl/certs/apache-selfsigned.crt
   SSLCertificateKeyFile /etc/ssl/private/apache-selfsigned.key
</VirtualHost>
<VirtualHost *:80>
        ServerName gojogym.com
        Redirect / https://gojoym.com/
</VirtualHost>_
```

Figure xxi: redirecting HTTP to HTTPS

Test the configuration syntax again and reload apply to apply the new changes.

For testing purposes, create a DocumentRoot and put a php file in.

```
user@jason:/etc/netplan$ sudo mkdir /var/www/gojogym.com
```

Figure xxii: creating a DocumentRoot

```
user@jason:/etc/netplan$ sudo nano /var/www/gojogym.com/test.php
```

Figure xxiii: creating test php file

In the php file write a simple html script.

```
  GNU nano 4.8                          /var/www/gojogym.com/test.php
<h1>testing gojogym</h1>
```

Figure xxiv: sample simple html script in test.php file

Now configure the netplan by enabling the static IP and switch from Nat to Bridge network for the server. On the kali virtual machine navigate to the terminal and ping the server by entering “ping 192.168.206.62”. If successful, configure the “/etc/hosts/” to associate the hostname with the server’s IP address. This allows access to the site using the hostname instead of the IP address. Add the server’s IP address followed by the hostname as shown in the figure xix

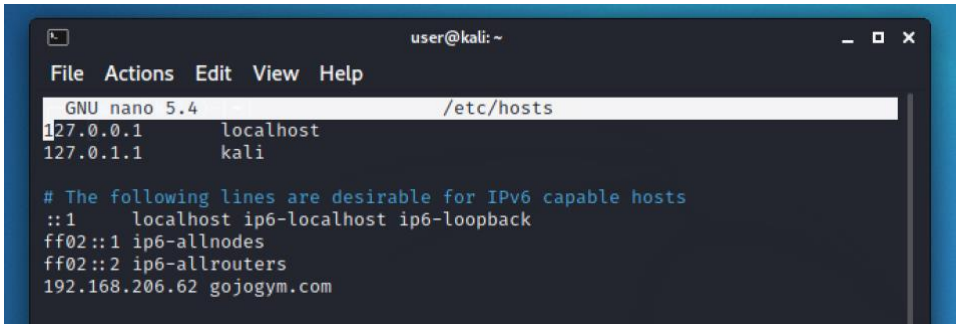

Figure xxv: configuring “/etc/hosts/”

Visit the php site on a browser to confirm if it worked

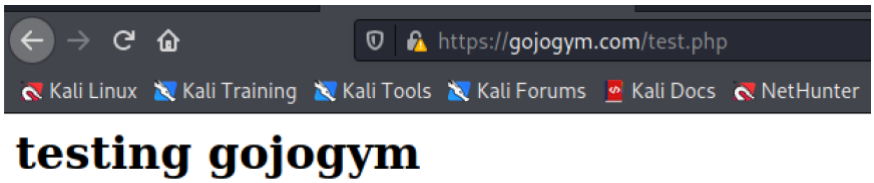

Figure xxvi: test.php on browser

6. SSH enabling, connecting, and hardening

By default, SSH is not enabled and to enable it, first make sure to be on Nat and netplan configured to access the internet then update the server with "sudo apt update" and install the openssh-server

package with “sudo apt install openssh-server”. Verify that SSH is running by entering "sudo systemctl status ssh".

With SSH now enabled, it ensures that connection to the server from another computer is secure and helps prevent eavesdropping and data interception from attackers as it uses encryption to protect transmitted data. SSH also allows for remote access control as it allows users with permission to connect to the server to do so remotely.

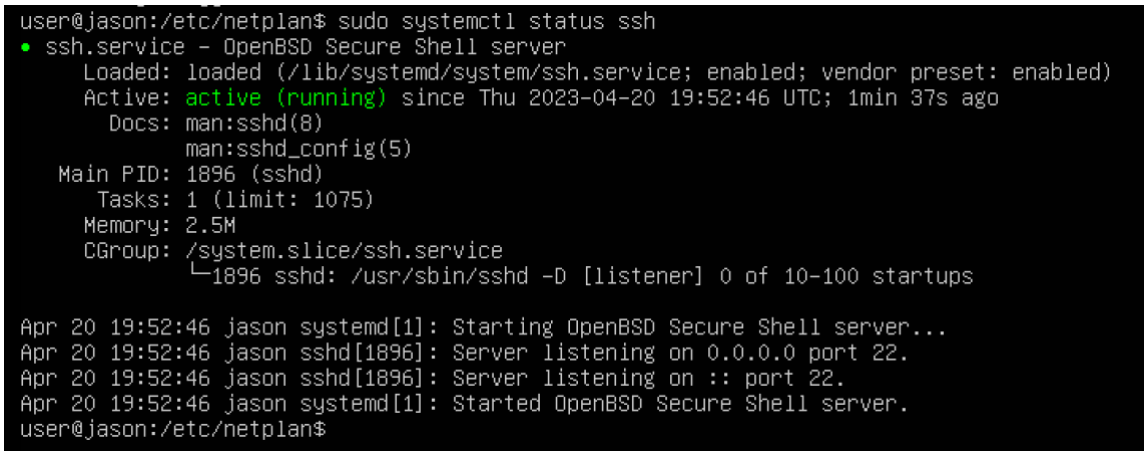

Figure xxvii: verifying if SSH activate

To connect to the SSH, switch to bridge on both machines and ping each other. Then in the terminal of client machine, enter “ssh user@192.168.206.62”, choose yes in when prompted and enter the server’s password to connect to the server through the client’s machine.

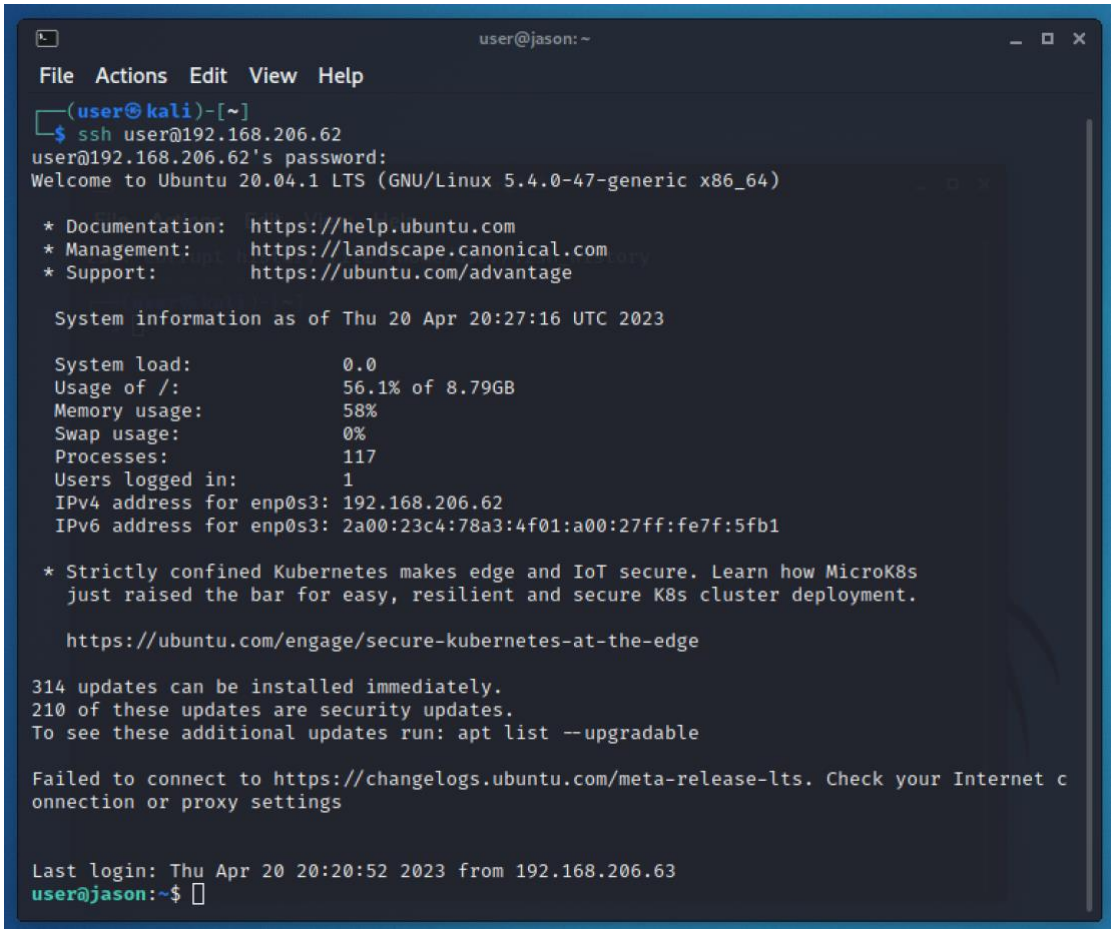

Figure xxviii: successful SSH connection

Hardening the SSH increases the security of the server. To do this, navigate to the SSH directory by entering “cd /etc/ssh/” then using nano text editor open “sshd_config” file. Inside the sshd_config file, uncomment lines to add layers of security to the SSH. After enabling the rules, save the file and restart SSHD with “sudo systemctl restart ssh”.

There are several configurations done on the sshd_config and they are as follows.

- “AllowUsers”: This will restrict SSH access to certain users, reducing the risk of unauthorised access by specifying which users can log into the server through SSH. In this example only “user” is allowed SSH log in.

```
user@jason:/etc/ssh$ grep AllowUsers /etc/ssh/sshd_config
AllowUsers user
user@jason:/etc/ssh$ _
```

- “PermitTunnel no”: This turned on will prevent users from bypassing security measures such as firewalls.

```
user@jason:/etc/ssh$ grep "PermitTunnel no" /etc/ssh/sshd_config
PermitTunnel no
user@jason:/etc/ssh$ _
```

- “MaxSessions”: This turned on will protect the SSH from denial of service by preventing overwhelming the daemon.

```
MaxSessions 10

#PubkeyAuthentication yes
```

- “ClientAliveInterval”: This will set a timeout interval (300 in this example) for inactivity and send a message to the client for a response.

```
#Compression delayed
ClientAliveInterval 300
```

- “ClientAliveCountMax”: This disconnects the client and terminates the session after sending the specified number for ClientAliveCountMax messages. In this example after 3 ClientAliveCountMax messages, users will be disconnected from the SSH

```
ClientAliveCountMax 3
#UseDNS no
```

- “PermitEmptyPasswords”: This will disallow logins to accounts with empty passwords on the SSH

```
# To disable tunneled clear text passwords, change to no here!
#PasswordAuthentication yes
PermitEmptyPasswords no
```

- “MaxAuthTries”: This will minimise the risk of successful brute force attacks on the SSH server.

```
user@jason:/etc/ssh$ grep MaxAuthTries /etc/ssh/sshd_config
MaxAuthTries 4
user@jason:/etc/ssh$ _
```

7. Installing and configuring fail2ban

Fail2Ban intrusion prevention software provides an extra layer of security to the server. It protects the server from brute-force attacks by detecting repetitive failed login attempts and blocks the IP address of the source. Fail2Ban is flexible and easy to customise, allowing the security configurations to be customised.

To install fail2ban, configure the server to have access to the internet. In the terminal update the system then enter, “sudo apt install fail2ban”. This will install fail2ban on the server, confirm if it installed by entering “sudo systemctl status fail2ban”.

```
user@jason:/etc/netplan$ sudo systemctl status fail2ban
● fail2ban.service - Fail2Ban Service
     Loaded: loaded (/lib/systemd/system/fail2ban.service; enabled; vendor preset: enabled)
     Active: active (running) since Thu 2023-04-20 23:46:08 UTC; 31s ago
       Docs: man:fail2ban(1)
   Main PID: 4401 (f2b/server)
      Tasks: 5 (limit: 1075)
     Memory: 14.2M
     CGroup: /system.slice/fail2ban.service
             └─4401 /usr/bin/python3 /usr/bin/fail2ban-server -xf start

Apr 20 23:46:08 jason systemd[1]: Starting Fail2Ban Service...
Apr 20 23:46:08 jason systemd[1]: Started Fail2Ban Service.
Apr 20 23:46:09 jason fail2ban-server[4401]: Server ready
user@jason:/etc/netplan$ _
```

Figure xxix: confirmation of successful fail2ban installation

Fail2ban has four configuration files. Create a jail.local file by running “sudo cp /etc/fail2ban/jail.conf /etc/fail2ban/jail.local” then use nano text editor to open the newly created file “sudo nano /etc/fail2ban/jail.local”.

```
  GNU nano 4.8                          /etc/fail2ban/jail.local
#
# WARNING: heavily refactored in 0.9.0 release.  Please review and
#          customize settings for your setup.
#
# Changes:  in most of the cases you should not modify this
#           file, but provide customizations in jail.local file,
#           or separate .conf files under jail.d/ directory, e.g.:
#
# HOW TO ACTIVATE JAILS:
#
# YOU SHOULD NOT MODIFY THIS FILE.
#
# It will probably be overwritten or improved in a distribution update.
#
# Provide customizations in a jail.local file or a jail.d/customisation.local.
# For example to change the default bantime for all jails and to enable the
# ssh-iptables jail the following (uncommented) would appear in the .local file.
# See man 5 jail.conf for details.
#
# [DEFAULT]
# bantime = 1h
#
# [sshd]
# enabled = true
#
# See jail.conf(5) man page for more information



# Comments: use '#' for comment lines and ';' (following a space) for inline comments


[INCLUDES]
                                [ Read 953 lines ]
^G Get Help  ^O Write Out  ^W Where Is  ^K Cut Text  ^J Justify   ^C Cur Pos    M-U Undo
^X Exit      ^R Read File  ^\ Replace   ^U Paste Text ^T To Spell  ^_ Go To Line M-E Redo
```

Figure xxx: jail.local file

- Several configurations can be done to increase the security of the server. Setting the “maxretry” value to “4” will strengthen the SSH as after 4 unsuccessful attempts, the IP address will be banned.

```
# "maxretry" is the number of failures before a host get banned.
maxretry = 4
```

- Modifying the “Find time” makes brute force attack on the server an unviable attack method as it may take specified minutes to be allowed to retry to log into the server.

```
# A host is banned if it has generated "maxretry" during the last "findtime"
# seconds.
findtime  = 10m
```

- Editing the “Ban time” will indicate how long an IP is banned. To make it very secure, setting the “Ban time” to a negative number will permanently ban the IP address. This prevents repetitive attempts from an attacker attempting to gain access to the server.

```
# "bantime" is the number of seconds that a host is banned.
bantime  = -1
```

- To avoid banning a specific IP address, for instance the client IP address, add the IP to the “ignoreip”

```
# "ignoreip" can be a list of IP addresses, CIDR masks or DNS hosts. Fail2ban
# will not ban a host which matches an address in this list. Several addresses
# can be defined using space (and/or comma) separator.
ignoreip = 127.0.0.1/8 ::1 192.168.206.63
```

After modifying the jail.conf file, save it then restart fail2ban with “sudo systemctl restart fail2ban” to activate these changes.

8. Installing and configuring Pluggable Authentication Modules (PAM) Package

On Nat, update the server then install the pam package with “sudo apt install libpam-pwquality”. Navigate to “/etc/security/pwquality.conf” to configure password policies. This pwquality configuration provides several password policies that can be customised to further harden the server. Note, uncomment the policy to be enforced. The policies implemented in this server are as follows.

- Password minimum length: Increasing the length of the password ensures that users’ passwords are strong and not easily guessable. “minlen” was set to 14.

```
user@jason:/$ grep '^\s*minlen\s*' /etc/security/pwquality.conf
minlen = 14
user@jason:/$ _
```

- Require at least one digit, one uppercase, one special character, one lowercase character: Setting “dcredit” to -1 will force users to have at least one digit, “ucredit” to -1 to have at least one upper case character, “ocredit” to -1 to have at least one special character, and “lcredit” to -1 to have at least one lowercase character in their password which makes it harder for an attacker to guess.

```
user@jason:/$ grep '^\s*dcredit\s*' /etc/security/pwquality.conf
dcredit = -1
user@jason:/$ grep '^\s*ucredit\s*' /etc/security/pwquality.conf
ucredit =-1
user@jason:/$ grep '^\s*ocredit\s*' /etc/security/pwquality.conf
ocredit = -1
user@jason:/$ grep '^\s*lcredit\s*' /etc/security/pwquality.conf
lcredit =-1
user@jason:/$ _
```

- To prevent users from reusing their old passwords, which could be a security threat as an attacker might be able to guess it, open the “common-password” configuration file from “/etc/pam.d”. and include “password required pam_pwhistory.so remember=5” where appropriate.

```
password        required                        pam_pwhistory.so remember=5
```

- In the “common-password” configuration file, ensure the password hashing algorithm being used Is SHA-512 as it provides a much stronger hashing compared to MD5. This provides an additional protection to the system by increasing the difficult an attacker will have to experience to determine passwords. Add “password [success=1 default=ignore] pam_unix.so sha512” in the file where appropriate.

```
password        [success=1 default=ignore]      pam_unix.so sha512
```

- Configure the “login.defs” file to add several security hardening features on server. Modify the “PASS_MIN_DAYS” parameter to restrict the frequent password changes and prevent repetitively changing of password in attempt to bypass password reuse controls. “PASS_MIN_DAYS” was set to no less than 1 day.

```
user@jason:/etc/security$ grep PASS_MIN_DAYS /etc/login.defs
#       PASS_MIN_DAYS   Minimum number of days allowed between password changes.
PASS_MIN_DAYS   1
user@jason:/etc/security$ _
```

- Inside the “login.defs” modify the “PASS_MAX_DAYS” parameter to set an expiration day for the password. This is done to reduce an attacker’s window to comprise credentials. “PASS_MAX_DAYS” was set to 90 days therefore, after every 90 days a new password will have to be set.

```
user@jason:/etc/security$ grep PASS_MAX_DAYS /etc/login.defs
#       PASS_MAX_DAYS   Maximum number of days a password may be used.
PASS_MAX_DAYS   90
user@jason:/etc/security$
```

- To warn a user about their password expiration before it expires, modify the “PASS_WARN_AGE” parameter. It was set to 7 days so a week before a user’s password expires, they will be warned.

```
user@jason:/etc/security$ grep PASS_WARN_AGE /etc/login.defs
#       PASS_WARN_AGE   Number of days warning given before a password expires.
PASS_WARN_AGE   7
user@jason:/etc/security$ _
```

9. Hotlink protection

To implement a hotlink protection on the server, navigate to the root directory of the website, which in my case was “/var/www/gojogym.com” and create a “.htaccess” file using nano. Inside this newly created file add figure xxxi.

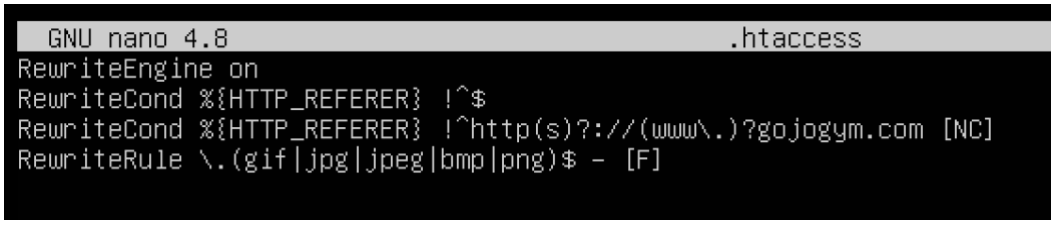

Figure xxxi: : .htaccess

10. Implementing an Intrusion Detection System (Snort)

Intrusion Detection System helps in monitoring the server for suspicious activities such as security breaches and enables users to respond quickly to such activity effectively. The Intrusion Detection System implemented was Snort. To install snort the server must have access to the internet and in the terminal enter “sudo apt-get install snort”. Snort will ask for the interface to listen on, find this by entering “ifconfig” on the terminal.

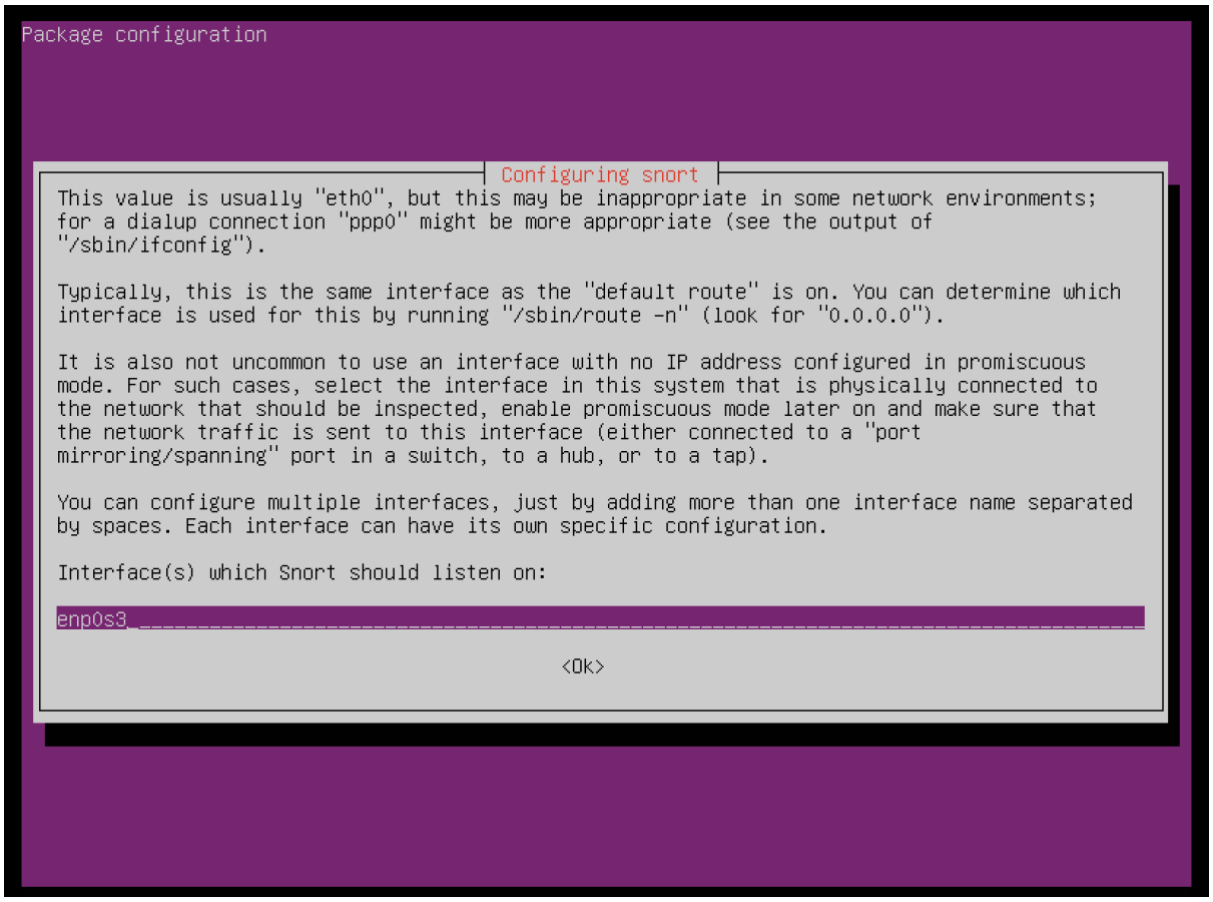

Figure xxxii: Specifying interface Snort should listen on

After installation, navigate to snort configuration file and open by entering “sudo nano /etc/snort/snort.conf”. Inside the file, uncomment lines to enable specific rules. Enable default snort rule.

```
# site specific rules
include $RULE_PATH/local.rules
```

Figure xxxiii: Snort default rule

Save the file then in the terminal enter “sudo snort -T -c /etc/snort/snort.conf -I enp0s3” to validate the configuration.

```
Snort successfully validated the configuration!
Snort exiting
user@jason:/var/www/gojogym.com$
```

Figure xxxiv: successful snort configuration validation

Restart snort to apply the changes.

11. Implementing an unattended updates system

Using unattended updates system minimises downtime or security breaches due to outdated software or vulnerabilities. Install the unattended-upgrades packages by running “sudo apt install unattended-upgrades” in the terminal. Open the configuration file for unattended upgrades by entering “sudo nano /etc/apt/apt.conf.d/50unattended-upgrades” after installation. Inside this file, remove the slashes on “Unattended-Upgrade::Allowed-Origins” to allow the rule.

```
// pocket these get automatically pulled in.
Unattended-Upgrade::Allowed-Origins {
        "${distro_id}:${distro_codename}";
```

Figure xxxv: Allowing unattended update rule

Uncomment “Unattended-Upgrade::Automatic-Reboot” and change the value to “true” to allow the system to automatically reboot after installing updates.

```
// Automatically reboot *WITHOUT CONFIRMATION* if
//  the file /var/run/reboot-required is found after the upgrade
Unattended-Upgrade::Automatic-Reboot "true";
```

Figure xxxvi: Allowing system reboot after updates

Save the unattended upgrades configuration file and open the package update manager configuration file by running “sudo nano /etc/apt/apt.conf.d/20auto-upgrades” command. Ensure the line “APT::Periodic::Update-Package-Lists” is uncommented to allow the system to update package list automatically. Uncomment “APT::Periodic::Unattended-Upgrade” and set the value to “true” to allow the server to automatically install security updates to ensure the system is always up-to-date.

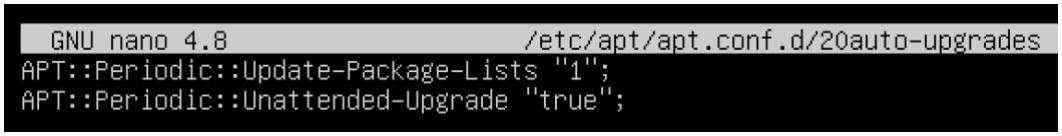

Figure xxxvii: Allowing security updates and package installation

Save the package update manager configuration file and restart the unattended-upgrades with “sudo systemctl restart unattended-upgrades” command to apply the changes.

12. Web application firewall (WAF) implementation

Before setting up a web application firewall, update the system. Web application firewall provides advanced security capabilities and protects the server from common web-based attacks such as SQL injection, cross-site scripting.

In the terminal, install ModSecurity and its Apache module by running “sudo apt-get install libapache2-mod-security2” command. The next step to enable the ModSecurity module then restart apache2. This is done by entering “sudo a2enmod security2” followed by “sudo systemctl restart apache2”.

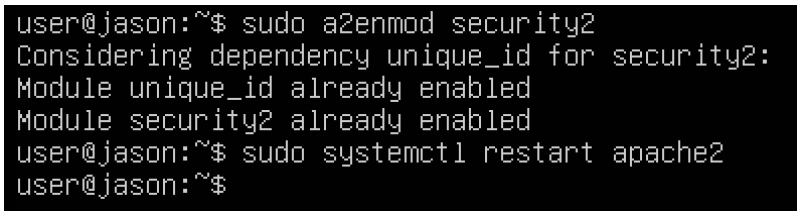

Figure xxxviii: Enabling ModSecurity module

Create the configuration file for ModSecurity by running “sudo cp /etc/modsecurity/modsecurity.conf-recommended /etc/modsecurity/modsecurity.conf”. then open the ModSecurity configuration file and modify the rules for the website with “sudo nano /etc/modsecurity/modsecurity.conf”.

One modification that was done inside the ModSecurity configuration file was “SecRequestBodyLimit”. A limit was set on “SecRequestBodyLimit” to help prevent an attacker from sending a large amounts of data to the server to consume the server resourses and cause a denial-of-service attack.

```
#
SecRequestBodyLimit 13107200
SecRequestBodyNoFilesLimit 131072
```

Figure xxxix: Enabling ModSecurity rule 1

Another modification that was done was the “SecRequestBodyLimitAction”. This option was set to “Reject” to prevent an attacker from exceed the request body’s specified limit.

```
#
SecRequestBodyLimitAction Reject
```

Figure xl: Enabling ModSecurity rule 2

After implementing the changing in the file, save the ModSecurity configuration file and then on the terminal, reload apache2 with “sudo systemctl reload apache2”

13. Two-Factor Authentication implementation

Install Two-Factor Authentication by running “sudo apt-get install libpam-google-authenticator“ in the terminal. After a successful installation, enter “google-authenticator” to start Google Authenticator setup. Several questions will be asked and a QR code given. When asked to disallow multiple uses of same authentication code, select yes as it Is more secure since a token will belong to one and only one user.

```
Do you want to disallow multiple uses of the same authentication
token? This restricts you to one login about every 30s, but it increases
your chances to notice or even prevent man-in-the-middle attacks (y/n)
```

Figure xli: Two-Factor Authentication installation question 1

When ask about rate limit, select yes as it makes brute force attacks an unviable attack choice since it prevents not more than 3 logins attempts in 30s.

```
If the computer that you are logging into isn't hardened against brute-force
login attempts, you can enable rate-limiting for the authentication module.
By default, this limits attackers to no more than 3 login attempts every 30s.
Do you want to enable rate-limiting? (y/n)
```

Figure xlii: Two-Factor Authentication installation question 2

Install google authenticator on mobile device and scan the given QR code. Continue answering the question and when done, edit the PAM configuration by entering “sudo nano /etc/pam.d/sshd” to open the file then add “auth required pam_google_authenticator.so” at the top of the file.

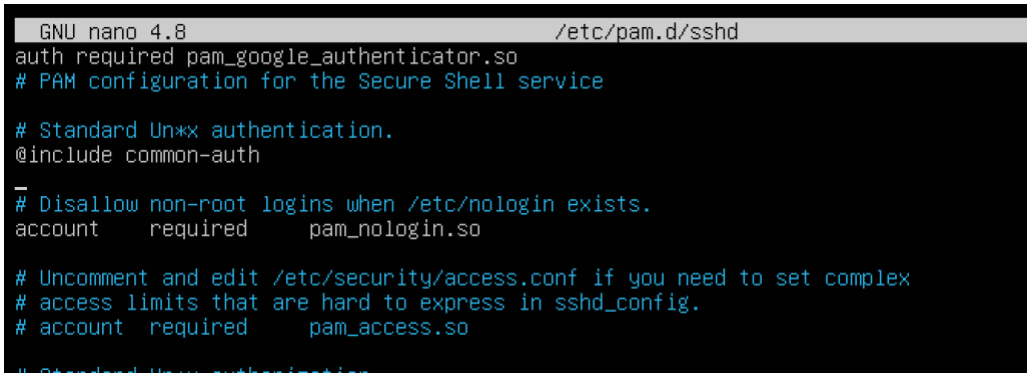

Figure xliii: Allowing google authentication in PAM configuration file

Afterward, restart the SSH service with “sudo systemctl restart sshd” command to apply the changes.

TESTING

When it comes to server management and security, testing is an important aspect. Testing ensures that security measures that are implemented work effectively and helps in identifying vulnerabilities or weaknesses on the server. Regular testing of the server also helps prevent potential security breaches and reduces the risk of data loss and system downtime. Several test was done on selected security features on the ubuntu server.

1. Two-Factor Authentication testing

The two-Factor Authentication was tested to ensure it worked correctly. To do this, ensure the client machine and the server can ping each other by configuring the networks. Inside the terminal of the client machine, enter the server with SSH by running “ssh user@192.168.206.62” command. If SSH asks for a verification, which is the six-digit number on the google authenticator app, then the two-Factor Authentication works correctly.

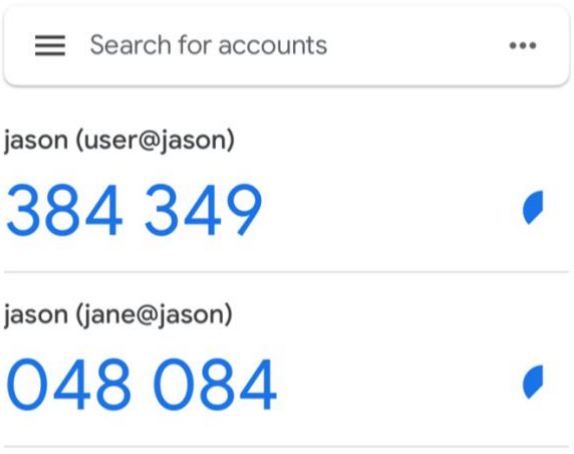

Figure xliv: 6 digit code for google authentication

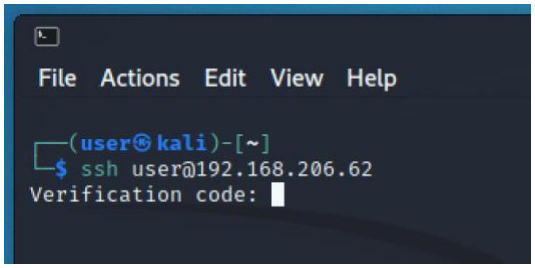

Figure xlv: SSH prompting for verification code

Another security feature that was implemented was “rate limit”. This was tested by intentionally entering the wrong password more than 3 times within 30 seconds to confirm if the server blocks login attempt.

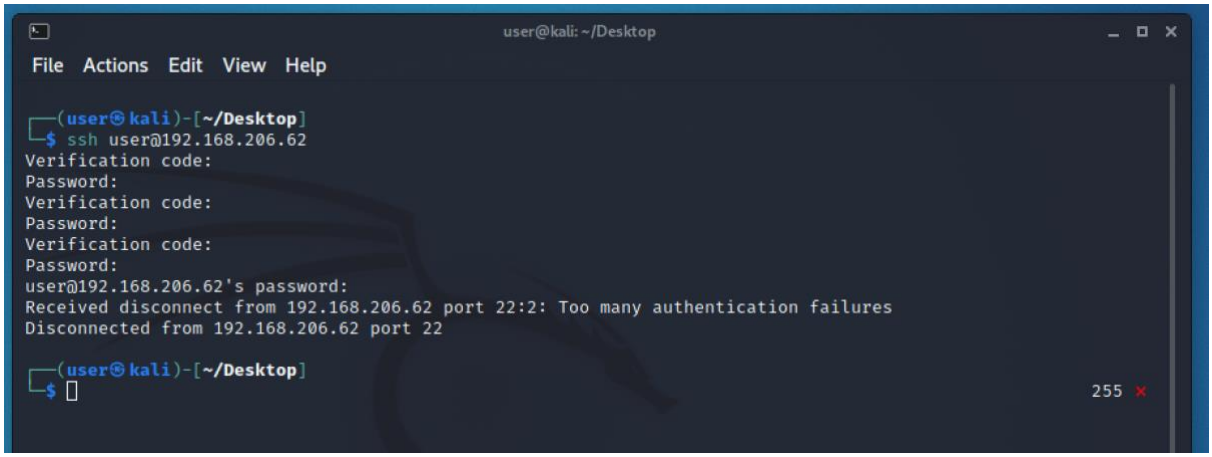

Figure xlvi: SSH showing verification failure

2. Snort testing

To test the snort intrusion detection system, Nmap tool was installed on a different virtual machine to scan the server’s network ports and generate network traffic which should trigger the Snort’s rules. Before running the scans, turn snort on by entering “sudo snort -i enp0s3 -c /etc/snort/snort.conf -A console” command on the server’s terminal.

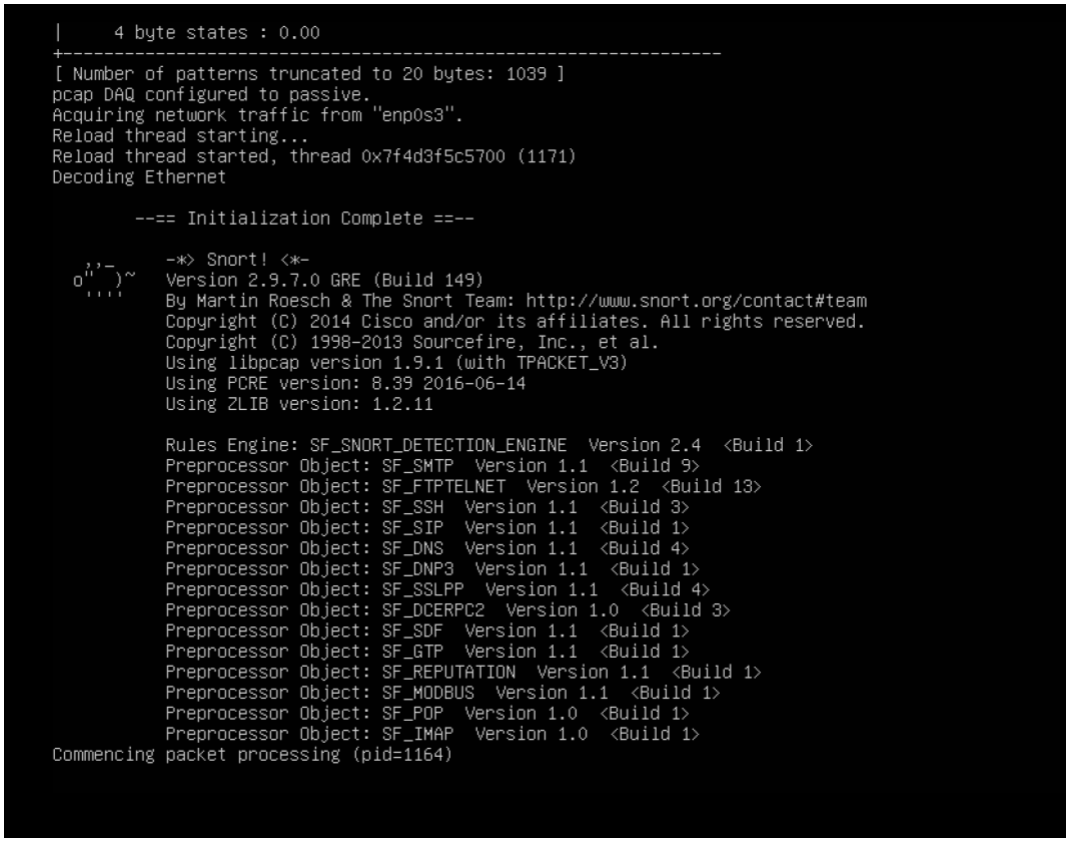

Figure xlvii: Snort activation

On the attacker machine with Nmap installed, run “nmap 192.168.206.62” to perform a basic scan on the server to confirm Nmap is active and can scan the server.

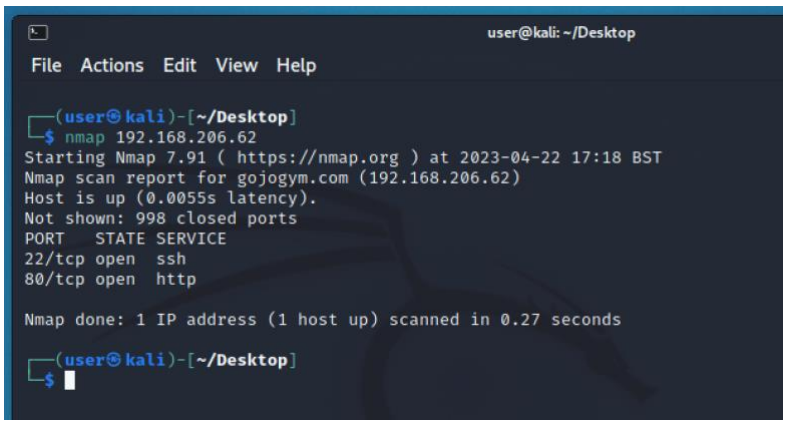

Figure xlviii: Confirming if nmap is activate

Perform a SYN scan on the server’s network ports to generate suspicious activity alert with “sudo nmap -sS 192.168.206.62”

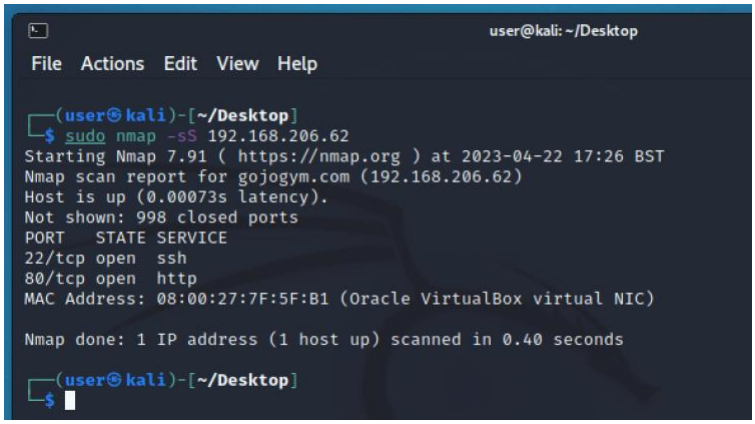

Figure xlix: Performing nmap commands to generate suspicious alert 1

Use nmap to scan for common vulnerabilities on the server to trigger the snort rules with “sudo nmap --script vuln 192.168.206.62”.

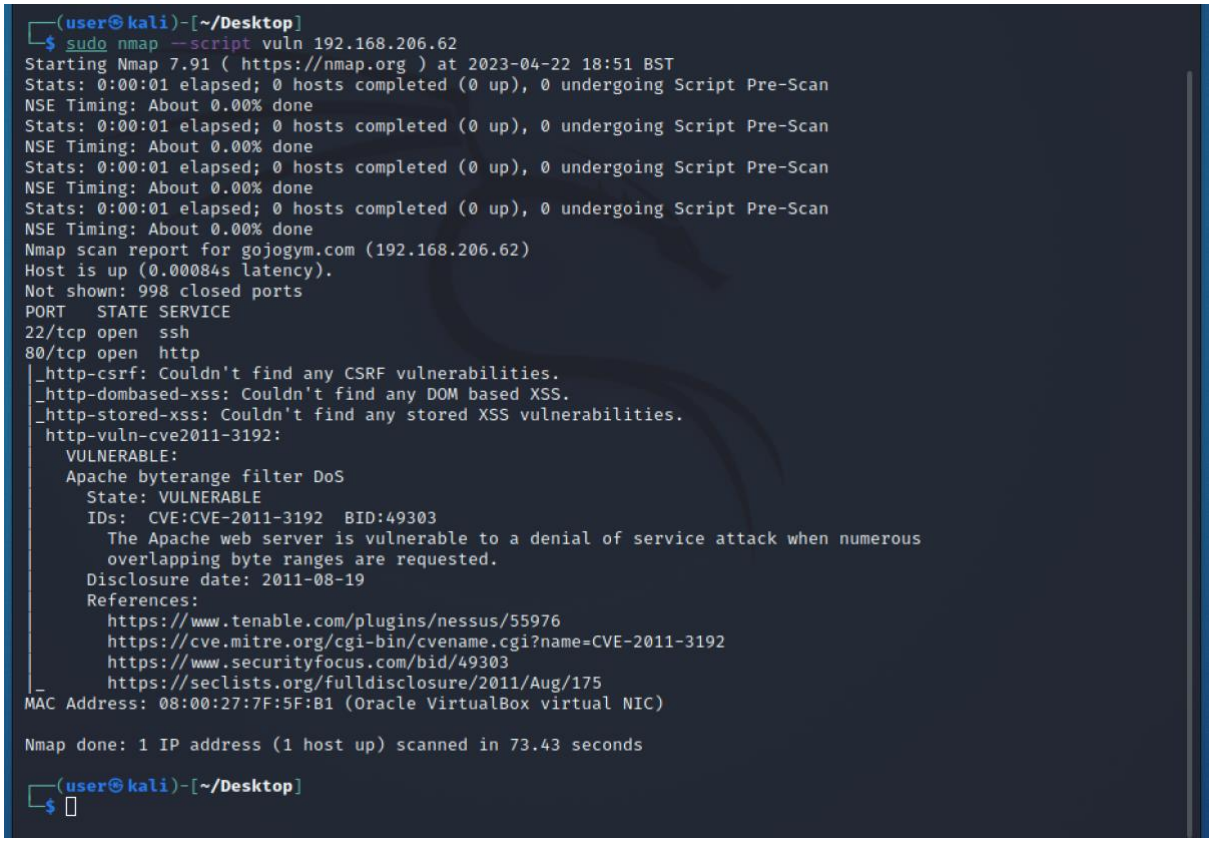

Figure l: Performing nmap commands to generate suspicious alert 2

On the server with snort activated, an “Attempted Information Leak” message is shown with the attack’s IP address meaning snort detected a suspicion activity.

```
04/22-17:51:34.436646  [**] [1:1421:11] SNMP AgentX/tcp request [**] [Classification: Attempted Info
rmation Leak] [Priority: 2] {TCP} 192.168.206.63:52576 -> 192.168.206.62:705
04/22-17:51:34.461219  [**] [1:1418:11] SNMP request tcp [**] [Classification: Attempted Information
 Leak] [Priority: 2] {TCP} 192.168.206.63:52576 -> 192.168.206.62:161
```

Figure li: Snort displaying attempted information leak

Exit snort and view all alerts associated with the server by running “grep "192.168.206.62" /var/log/snort/snort.log”.

3. Pluggable Authentication Modules (PAM) Package testing

To test the password policies that were set with the PAM package, run “passwd” to activate the password change.

```
user@jason:~$ passwd
Changing password for user.
Current password:
```

Figure lii: Password change for server

Enter current password to verify the user and when prompted to enter new password, enter one without a digit to test “dcredit”. If “BAD PASSWORD: The password contains less than 1 digit” message appears, then “dcredit” rule was set correctly.

```
New password:
BAD PASSWORD: The password contains less than 1 digits
New password: _
```

Figure liii: PAM Package test on dcredit

Test the “ucredit” rule by entering a password without an uppercase letter.

```
New password:
BAD PASSWORD: The password contains less than 1 uppercase letters
```

Figure liv: PAM Package test on ucredit

Test the “lcredit” rule by entering a password without a lowercase letter.

```
New password:
BAD PASSWORD: The password contains less than 1 lowercase letters
New password:
```

Figure lv: PAM Package test on lcredit

Test the “ocredit” rule by entering a password without a special letter.

```
New password:
BAD PASSWORD: The password contains less than 1 non-alphanumeric characters
```

Figure lvi: PAM Package test on ocredit

Test the “minlen” rule by entering a password less than 14 characters.

```
New password:
BAD PASSWORD: The password is shorter than 14 characters
New password: _
```

Figure lvii" PAM Package test on minlen

4. SSH hardening testing

Several tests were conducted on the SSH hardening implemented on the server. First was to test the “AllowUsers” rule in the sshd_config file. To do this, a new user was created on the server by running “sudo adduser jane” on the server’s terminal, setting a password for the user then following the additional prompts to add more information about this new user.

```
user@jason:~$ sudo adduser jane
[sudo] password for user:
Adding user `jane' ...
Adding new group `jane' (1002) ...
Adding new user `jane' (1002) with group `jane' ...
Creating home directory `/home/jane' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for jane
Enter the new value, or press ENTER for the default
        Full Name []: Jane Bones
        Room Number []: 1
        Work Phone []: 0792581273
        Home Phone []: 0248738096
        Other []: 1
Is the information correct? [Y/n] y
user@jason:~$
```

Figure lviii: Creating new user on server

After creating the user, configure two-factor authenticator on this new user. Switch to the client machine and try to log into server through SSH with a user that is allowed to test if SSH is working correctly. Afterwards, try using this newly created user to log into the server through SSH. Since this new user is not listed in the “AllowUsers”, SSH should not let them in.

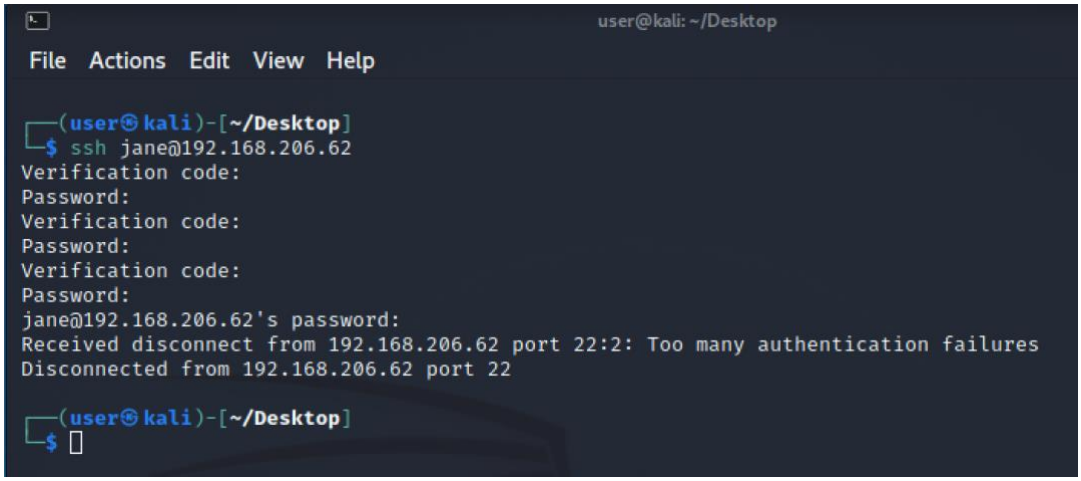

Figure lix: Unsuccessful SSH attempt for new user

Now, add the new user to the allowed users and try to log into the SSH as the new user. The SSH should let them in now since they are in the "AllowUsers" list

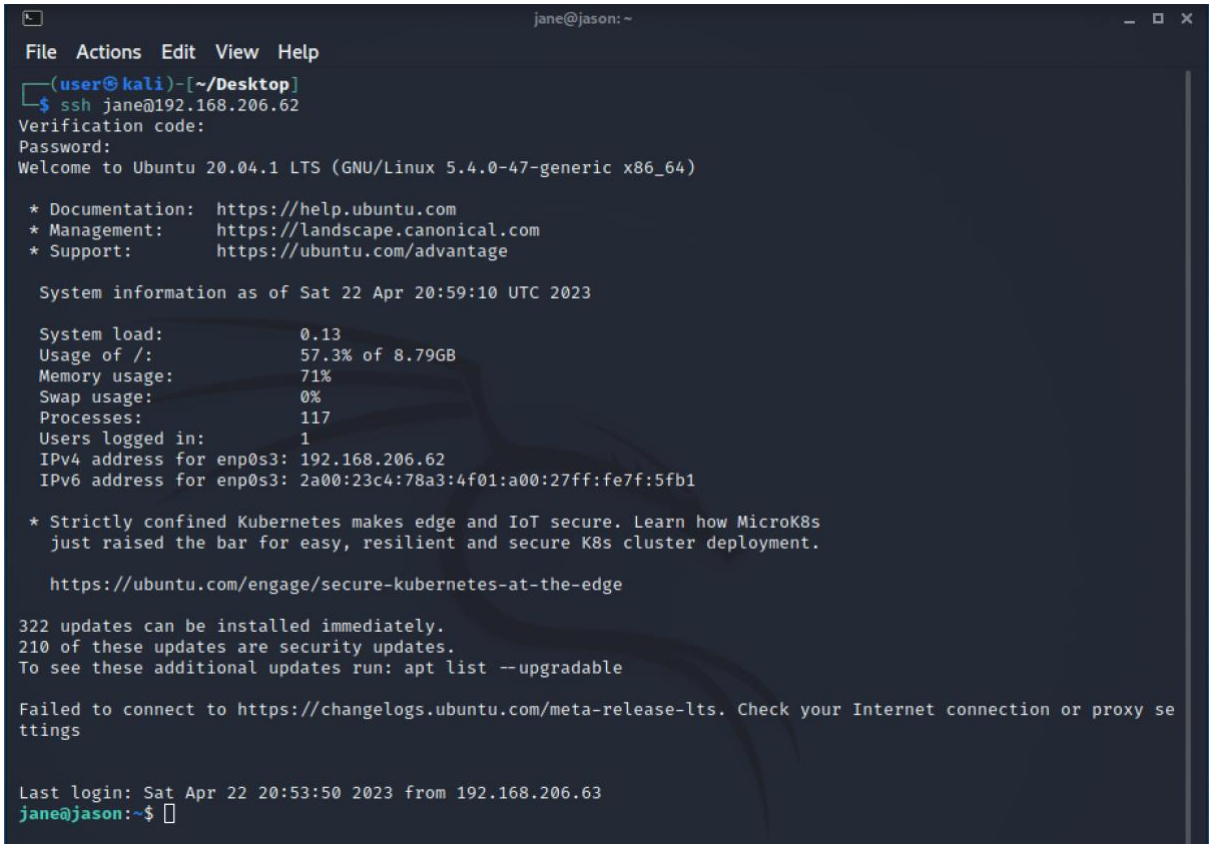

Figure lx: Successful SSH attempt for new user

The "PermitTunnel" rule in the sshd_config was tested as well. This was done by running "ssh -v -w 0:0 user@192.168.206.62" command on the client machine. Since "PermitTunnel" is set to "no", no tunnel will be established.

```
debug1: sys_tun_open: failed to configure tunnel (mode 1): Operation not permitted
Tunnel device open failed.
```

Figure lxi: Unsuccessful tunnel creation

5. Fail2ban testing

Testing the fail2ban configuration requires another virtual machine that has not been set on the "ignoreip" in the fail2ban configuration file. Configure the new virtual machine to ping the server and call it "attacker". In the terminal of the attacker machine, try to log into the SSH with the wrong credentials. If after 4 incorrect attempts and the SSH disconnects, the "maxretry" rule is active.

```
┌──(user㉿kali)-[~/Desktop]
└─$ ssh user@192.168.206.62
Password:
Password:
Password:
user@192.168.206.62's password:
Received disconnect from 192.168.206.62 port 22:2: Too many authentication failures
Disconnected from 192.168.206.62 port 22
```

Figure lxii: Fail2Ban test on maxretry

To test the "bantime" rule, try to login using SSH. SSH should refuse connection which indicate the host address Is banned

```
┌──(user㉿kali)-[~/Desktop]
└─$ ssh user@192.168.206.62
ssh: connect to host 192.168.206.62 port 22: Connection refused

┌──(user㉿kali)-[~/Desktop]
└─$
```

Figure lxiii: Fail2Ban test on bantime

Add the attacker IP address to the fail2ban configuration file and retry to login. SSH should allow relogin even after exceeded "maxretry".

```
┌──(user㉿kali)-[~/Desktop]
└─$ ssh user@192.168.206.62
Password:
Password:
Password:
user@192.168.206.62's password:
Received disconnect from 192.168.206.62 port 22:2: Too many authentication failures
Disconnected from 192.168.206.62 port 22

┌──(user㉿kali)-[~/Desktop]
└─$ ssh user@192.168.206.62
Password:
```

Figure lxiv: Fail2Ban test on maxretry 2

6. MySQL testing

The first security feature that was implemented into the MySQL was disabling root logins without authentication. To test this, run "sudo mysql". Access to the database should be denied.

```
user@jason:~$ sudo mysql
ERROR 1045 (28000): Access denied for user 'root'@'localhost' (using password: NO)
user@jason:~$ _
```

Figure lxv: MySQL disallowing entry by defaul root access

Afterwards, run “mysql -u root -p”. This command will prompt for the password to authenticate the user.

```
user@jason:~$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 9
Server version: 8.0.32-0ubuntu0.20.04.2 (Ubuntu)

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

Figure lxvi: MySQL test on root plus password access

Website Development Documentation

As part of this project, a website was created as a case study for analysing and developing a secure web server. The website, gojo.com, was designed for a fictional gym site that provides personal training services. The site's main goal was to offer users a convenient and secure way to book training sessions with a personal trainer. Using PHP, gojogym.com was created on the Ubuntu server. The gym website allows users to log in and book a training session with several personal trainers. Personal trainers can also log in to see whom their booked with and which day they have been booked. The website is linked with the web server and demonstrates the importance of web server security with a showcase of some of the best practices for developing a secure website.

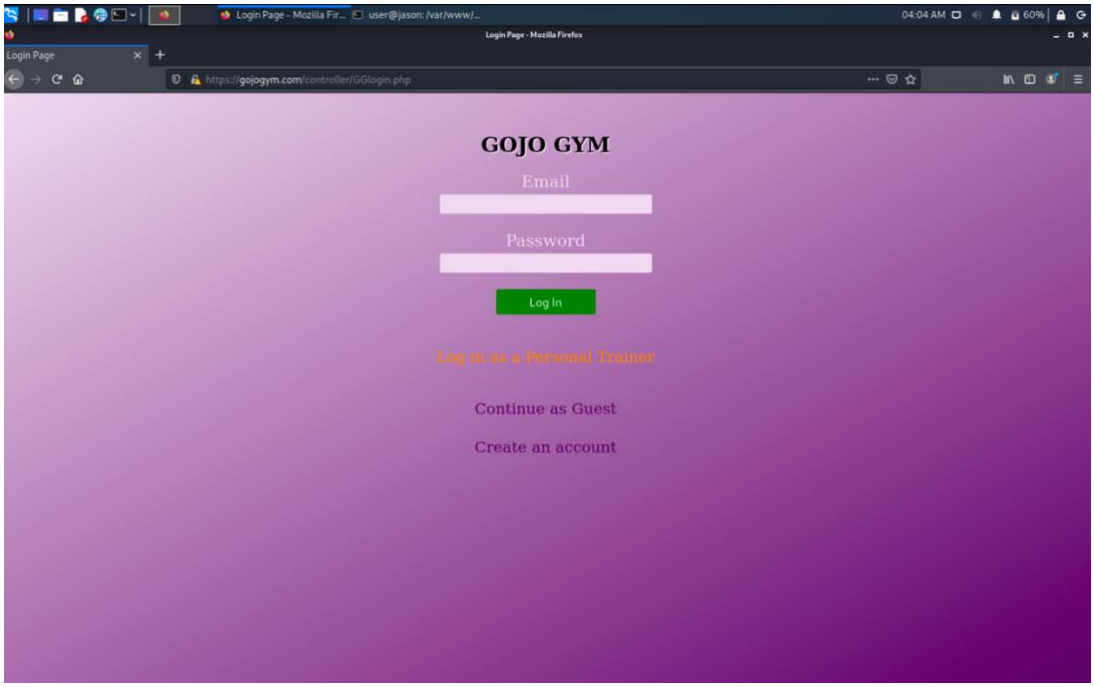

Figure lxvii: Client log in page

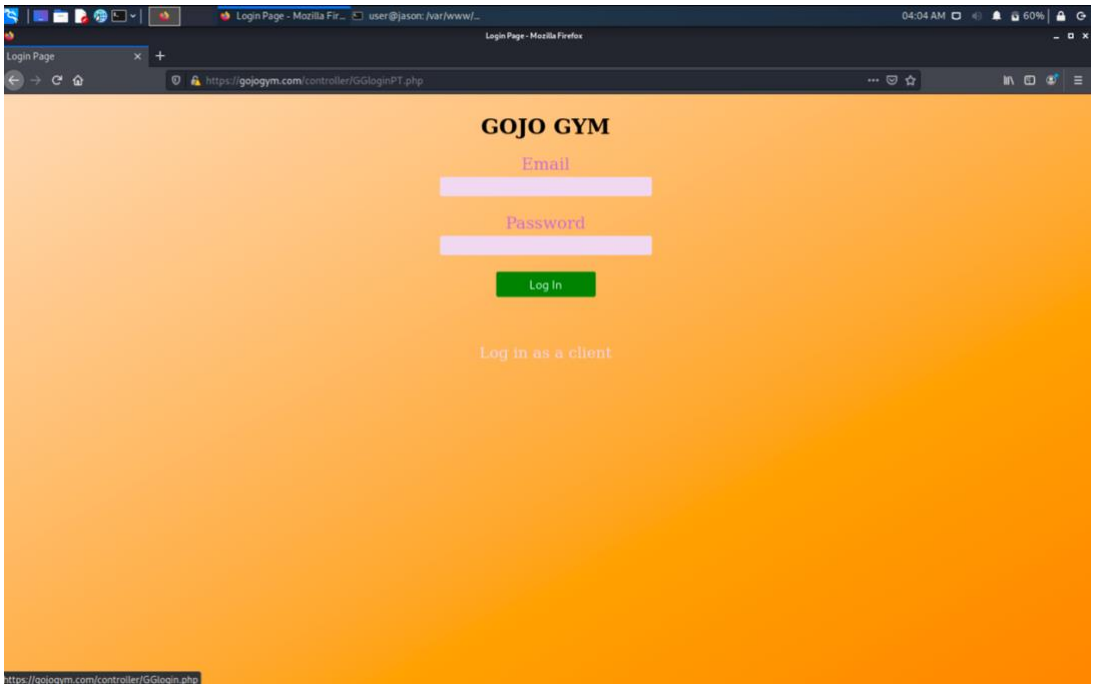

Figure lxviii: Personal Trainer log in page

On the user's dashboard page, users can see the opening hours for the gym and also book sessions with their preferred personal trainer.

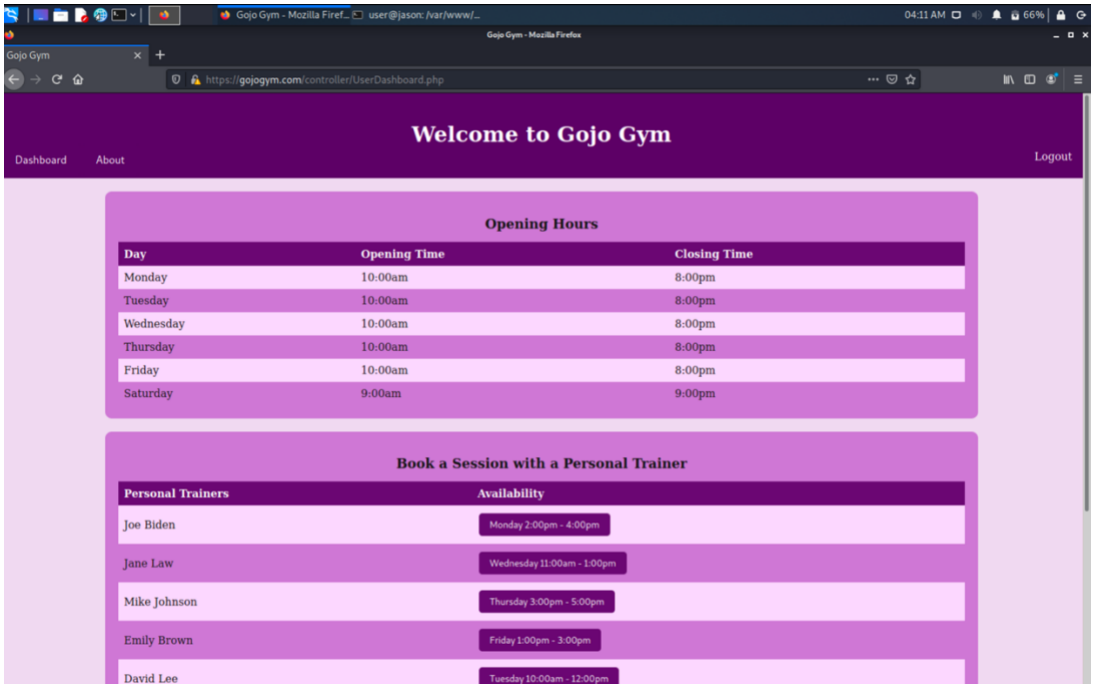

Figure lxix: Client Dashboard page

The About section for clients gives more information about gojogym.com and presents the team with a little information about each.

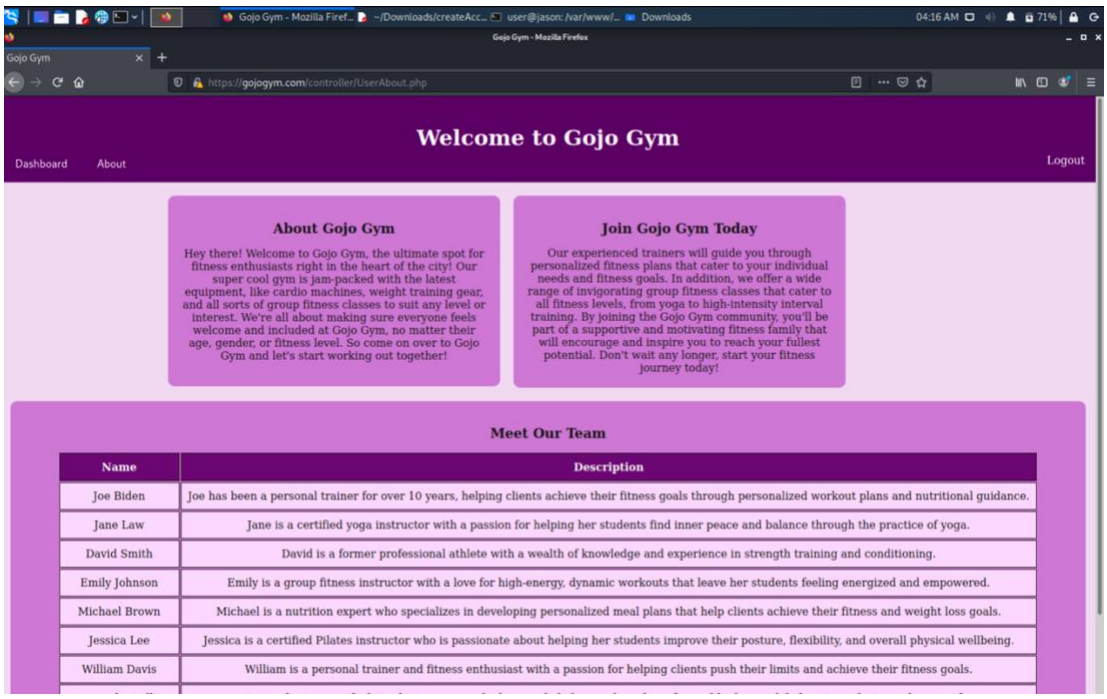

Figure lxx: Client About page

The personal trainer dashboard shows the PT who they are booked with and the time they have been scheduled for. It also has information about the importance of personal trainers.

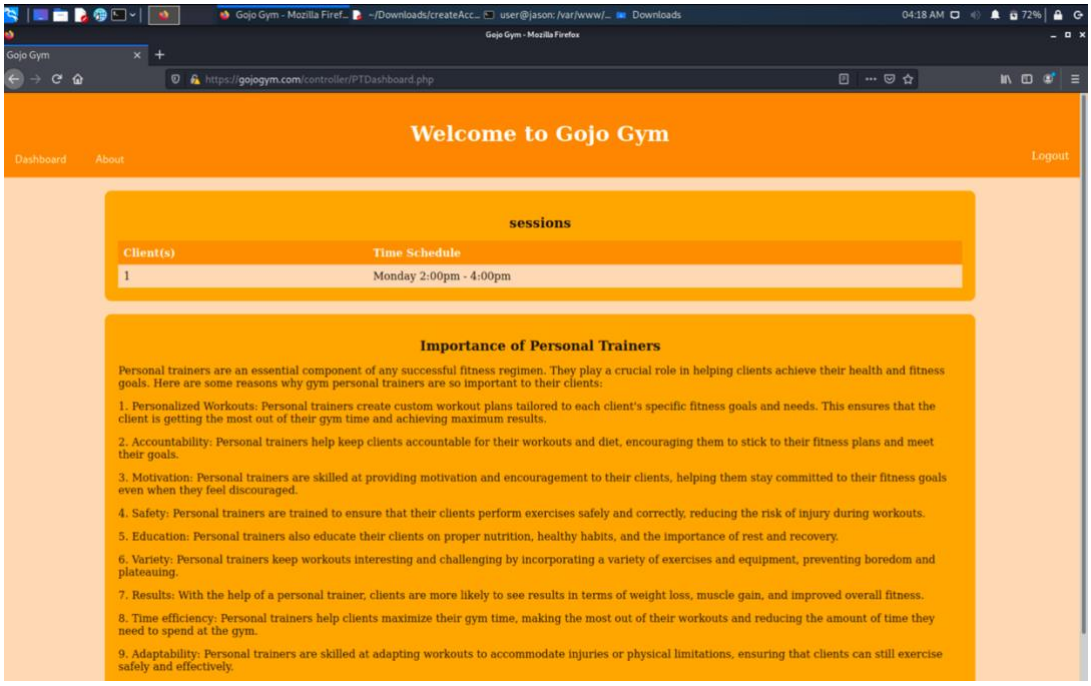

Figure lxxi: Personal Trainer Dashboard page

The About page for personal trainers has information about why gojo gym may be suited for the personal trainers and the benefit personal trainers are for clients.

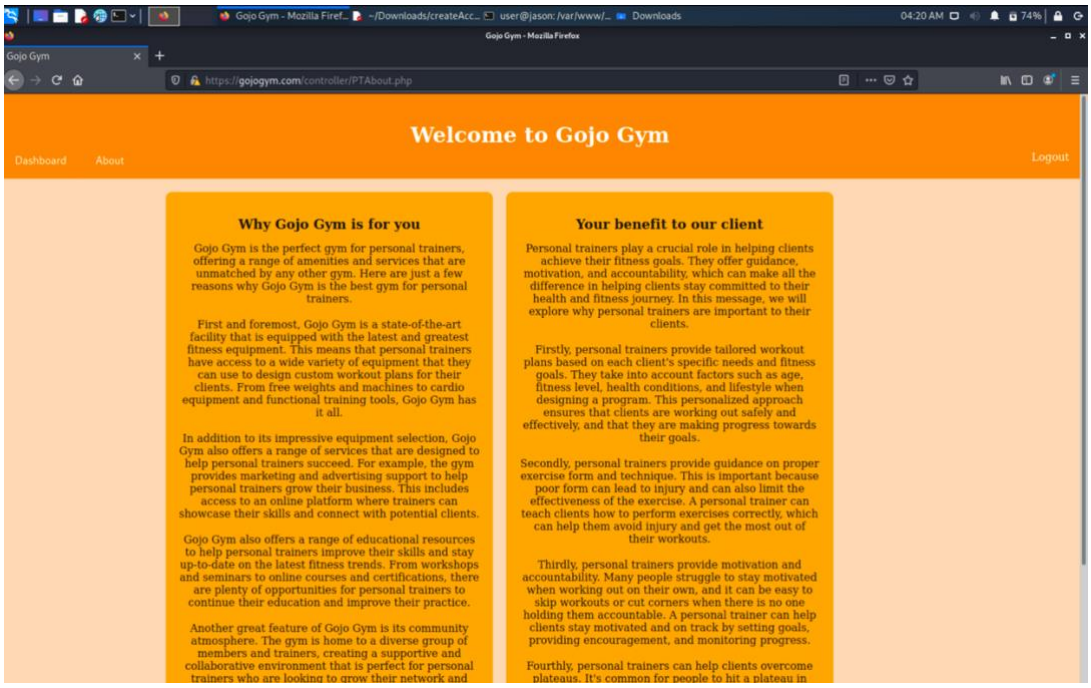

Figure lxxii: Personal Trainer About page

Guest can also access the website by clicking on the “Continue as a guest” option on the client log in page. The guest page only have useful information about gojo gym such as the Opening Hours, why they should join us and the team at gojo gym.

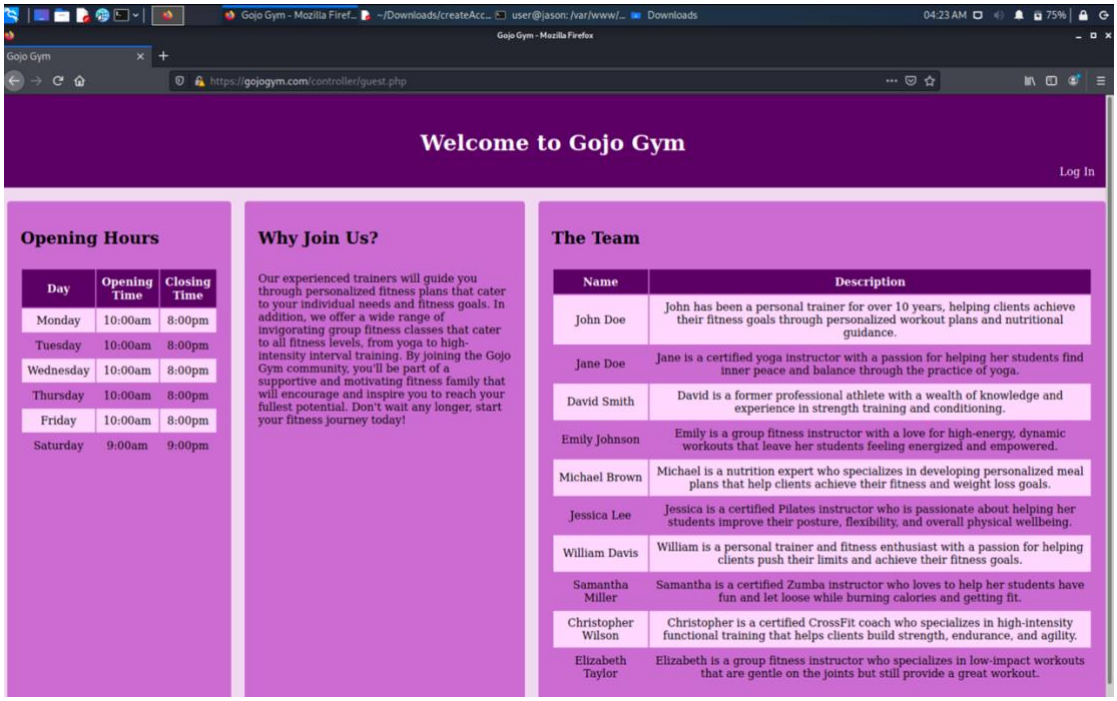

Figure lxxiii: Guest page

There is a create an account feature through the client log in page and here, the data entered gets stored in PHPadmin database.

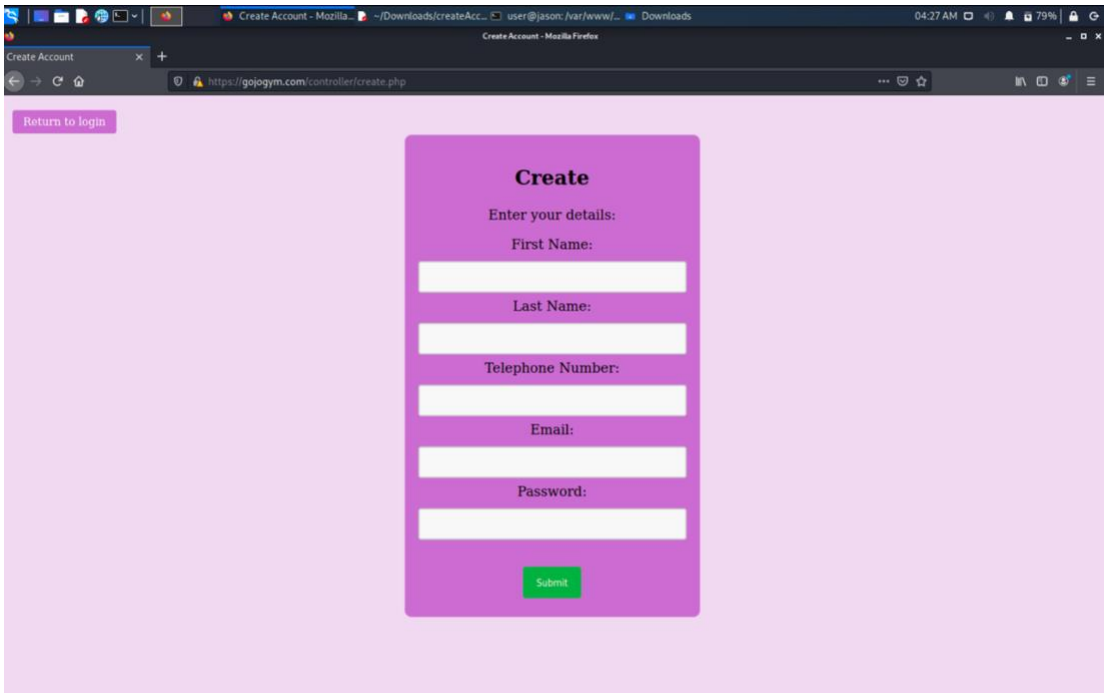

Figure lxxiv: Create an account page

Conclusion And Critical Review

This section of the report will critically review the outcome, aims and objectives and evaluate if the project succeeded in achieving the expected goals. The project completed all of the intended aims stated. A thorough investigation was conducted into different tools and systems that can b used to strengthen the security of web servers. A secure webserver was developed using information gained from research, and the developed web server was tested for vulnerabilities. Regarding the project's objectives, deep research was successfully conducted and provided a literature review about all the different ways to secure web servers.

This project was well-managed in terms of milestones, deliveries, and time management. The project's timeline was clear, and tasks were divided to make them manageable, allowing for effective monitoring. The project was reviewed regularly to ensure that project was on track and that any issues encountered were adequately addressed. Due to minor delays and challenges during the project, not all the established milestones were met on time. However, the project was successfully completed within the given timeline. Development, testing and documentation were carefully planned and executed on schedule. In terms of deliverables, the project achieved all stated deliverables, from the literature review to the documentation.

The aspect of the project that worked well was the implementation and testing of the LAMP stack, Secure Shell (SSH) and SSH hardening, Secure Socket Layer (SSL), Fail2ban, Two Factor Authentication, PAM Password policy configuration, Snort Intrusion Detection System, Unattended updates system, Web application firewall, creating a website using PHP and PHPadmin for the database and hotlink protection. These tools and software were successfully implemented and tested for effectively successfully.

The project succeeded in achieving its intended outcome. However, there are always areas that could have been changed to improve the project. The scope of the project could have been changed to allow for a more targeted and focused approach. The scope could have been refined to focus on a certain aspect of web server security that needs protection, such as protection from DDoS attacks. A more refined scope would have also helped in looking at other commonly used web servers such as Google Cloud Platform. Regarding the testing, a wide range of attacks could have been tested on the implemented security tools and software to ensure further the effectiveness of the web server developed. The project could have been extended to discuss the cost of security tools and software to give a more practical and realistic analysis of ways to secure web servers.

There are several potential future development paths to build on based on the findings and outcome of this project. One possible future development path is using sophisticated security tools and software designed specifically for web servers. This path revolves around developing new software or using machine learning and artificial intelligence algorithms to secure web servers. Another future development path is further research and development of industry-standard security practices and regulations regarding web server security. This will include new security frameworks and standards development, such as the NIST cybersecurity framework for web servers.

References

1. Ashfaq, R.A.R., Wang, X.-Z., Huang, J.Z., Abbas, H. and He, Y.-L. (2016). Shibboleth Authentication Request. *login.ezproxy.kingston.ac.uk*, [online] 378. doi:https://doi.org/10.1016.
2. Buzdar, K. (2017). Configuring Windows Deployment Services on Server 2012 R2 with DHCP Running on Ubuntu 14.04.5 LTS Server. Windows IT Pro (Online), Feb 15, 2017. Available at: https://www.proquest.com/magazines/configuring-windows-deployment-services-on-server/docview/1868547583/se-2?accountid=14557.
3. Chia, F.C., Tung, Y.H. and Yong, F.Y.Y. (2022). Examining the Agile Project Management Practices in the Malaysian Construction Industry. *IOP Conference Series: Earth and Environmental Science*, 1101(4), p.042041. doi:https://doi.org/10.1088/1755-1315/1101/4/042041.
4. Christoffer, C., Chen, S., Bharadwaj, V., Aderinwale, T., Kumar, V., Hormati, M., & Kihara, D. (2021). LZerD webserver for pairwise and multiple protein–protein docking. Nucleic Acids Research, 49(Web Server issue), W359-W365. https://doi.org/10.1093/nar/gkab336
5. Caponi, A. and Leuti, M. (2019). A Web Platform for Integrated Vulnerability Assessment and Cyber Risk Management. *Information*, 10(7), p.242. doi:https://doi.org/10.3390/info10070242.
6. David, M. and Alex, C. (2019). *Why You Need An SSL Certificate Now - ProQuest*. [online] www.proquest.com. Available at: https://www.proquest.com/docview/2306440934?accountid=14557&parentSessionId=r5oqmLDnS4G%2BZ%2BhdVqsFkQIELZqT5DdvK9IW17ZXjEs%3D&pq-origsite=primo [Accessed 23 Jan. 2023].
7. Dawadi, B.R., Adhikari, B. and Srivastava, D.K. (2023). Deep Learning Technique-Enabled Web Application Firewall for the Detection of Web Attacks. *Sensors*, 23(4), p.2073. doi:https://doi.org/10.3390/s23042073.
8. Dhuny, R., Peer, A.A.I., Mohamudally, N.A. and Nissanke, N. (2022). Performance evaluation of a portable single-board computer as a 3-tiered LAMP stack under 32-bit and 64-bit Operating Systems. *Array*, 15, p.100196. doi:https://doi.org/10.1016/j.array.2022.100196.
9. Evdokimov, I.V., Tsarev, R.Y., Yamskikh, T.N. and Pupkov, A.N. (2018). Using PERT and Gantt charts for planning software projects on the basis of distributed digital ecosystems. *Journal of Physics: Conference Series*, [online] 1074(1), p.012127. doi:https://doi.org/10.1088/1742-6596/1074/1/012127.
10. Gupta, R.K., Chawla, V., Pateriya, R.K., Shukla, P.K., Mahfoudh, S. & Syed Bilal, H.S. 2023, "Improving Collaborative Intrusion Detection System Using Blockchain and Pluggable Authentication Modules for Sustainable Smart City", *Sustainability,* vol. 15, no. 3, pp. 2133.
11. Gaborov, M. and Ivetić, D. (2022). The importance of integrating Thinking Design, User Experience and Agile methodologies to increase profitability. *Journal of Applied Technical and Educational Sciences*, [online] 12(1), pp.286–286. doi:https://doi.org/10.24368/jates286.
12. Lucian, C. (2019). *OpenSSH to protect keys in memory against - ProQuest*. [online] www.proquest.com. Available at: https://www.proquest.com/docview/2244432286/E2E07C33A77B4DC4PQ/64?accountid=14557 [Accessed 3 Feb. 2023].

13. Louridas, P. (2016). *Shibboleth Authentication Request*. [online] login.ezproxy.kingston.ac.uk. Available at: https://ieeexplore-ieee-org.ezproxy.kingston.ac.uk/stamp/stamp.jsp?tp=&arnumber=7420497&tag=1 [Accessed 1 Dec. 2022].
14. Max, E. (2017). *What makes LAMP stack important for your web - ProQuest*. [online] www.proquest.com. Available at: https://www.proquest.com/docview/1917371457?accountid=14557&parentSessionId=yO6eLb7Ks2Xf4r3WRImbg2vrbUmuPm8vwo%2Bvo%2Ffi0f8%3D&pq-origsite=primo [Accessed 18 Jan. 2023].
15. Matt, A. (2021). *How the cloud and big compute are remaking HPC - ProQuest*. [online] www.proquest.com. Available at: https://www.proquest.com/docview/2548473400?pq-origsite=primo [Accessed 9 Jan. 2023].
16. Nicula, D. and Ghimișı, S.S. (2019). Command and Control vs self Management. *IOP Conference Series: Materials Science and Engineering*, 514(1), p.012039. doi:https://doi.org/10.1088/1757-899x/514/1/012039.
17. Owoh, N.P. and Mahinderjit Singh, M. (2018). Security analysis of mobile crowd sensing applications. *Applied Computing and Informatics*. doi:https://doi.org/10.1016/j.aci.2018.10.002.
18. Paul, N. (2016). *Let's encrypt - but let's also decrypt and - ProQuest*. [online] www.proquest.com. Available at: https://www.proquest.com/docview/1764220510/7901BCD03A2E43B9PQ/2?accountid=14557 [Accessed 10 Feb. 2023].
19. Rama Devi, O., Parvathi Vallabhaneni, S., Hussain, M.A. and Kumar, T.K. (2020). Security Analysis on Remote Authentication against Man-in-the-Middle Attack on Secure Socket Layer. *IOP Conference Series: Materials Science and Engineering*, 981, p.022015. doi:https://doi.org/10.1088/1757-899x/981/2/022015.
20. Rundle, J. (2022). Rise in Cyberattacks Stretches and Stresses Defenders. *Wall Street Journal*. [online] 5 Oct. Available at: https://www.wsj.com/articles/rise-in-cyberattacks-stretches-and-stresses-defenders-11664962202 [Accessed 31 Dec. 2022].
21. Rajesh Kanna, P. and Santhi, P. (2021) "Unified deep learning approach for efficient intrusion detection system using integrated spatial–temporal features," Knowledge-Based Systems, 226, p. 107132. Available at: https://doi.org/10.1016/j.knosys.2021.107132.
22. Sandra, H.-S. (2015). *Keeping the bad guys out with fail2ban - ProQuest*. [online] www.proquest.com. Available at: https://www.proquest.com/docview/1655189068/69FA6C02E174F20PQ/9?accountid=14557 [Accessed 17 Mar. 2023].
23. Sandra, H.-S. (2022). *Using fail2ban on Fedora: Fail2ban can detect - ProQuest*. [online] www.proquest.com. Available at: https://www.proquest.com/docview/2641600158?accountid=14557&parentSessionId=2v4YUImaQMQ0Tv57gnqUtRbgiIlketeLC3cRT2oTksI%3D&pq-origsite=primo [Accessed 22 Feb. 2023].
24. Shaheed, A. and Kurdy, M.H.D.B. (2022). Web Application Firewall Using Machine Learning and Features Engineering. *Security and Communication Networks*, 2022, pp.1–14. doi:https://doi.org/10.1155/2022/5280158.

25. Shamshurin, I. and Saltz, J.S. (2022). Using a coach to improve team performance when the team uses a Kanban process methodology. *International Journal of Information Systems and Project Management*, 7(2), pp.61–77. doi:https://doi.org/10.12821/ijispm070204.
26. Sujatanagarjuna, A., Bochem, A. and Leiding, B. (2022). Formalizing and Safeguarding Blockchain-Based BlockVoke Protocol as an ACME Extension for Fast Certificate Revocation. *Cryptography*, 6(4), p.63. doi:https://doi.org/10.3390/cryptography6040063.
27. Tian, Q., Li, J. and Liu, H. (2019). A Method for Guaranteeing Wireless Communication Based on a Combination of Deep and Shallow Learning. *IEEE Access*, [online] 7, pp.38688–38695. doi:https://doi.org/10.1109/ACCESS.2019.2905754.
28. Tekerek, A. and Bay, O.F. (2019). DESIGN AND IMPLEMENTATION OF AN ARTIFICIAL INTELLIGENCE-BASED WEB APPLICATION FIREWALL MODEL. *Neural Network World*, 29(4), pp.189–206. doi:https://doi.org/10.14311/nnw.2019.29.013.
29. Wang, B.-X., Chen, J.-L. and Yu, C.-L. (2022). An AI-Powered Network Threat Detection System. *IEEE Access*, 10, pp.54029–54037. doi:https://doi.org/10.1109/access.2022.3175886.
30. Xia, Y., Wang, jin, Liu, C. and Yu, K. (2020). Design and Implementation of Vulnerability Scanning Tools for Intelligent Substation Industrial Control System Based on Openvas. *IOP Conference Series: Earth and Environmental Science*, 440, p.042031. doi:https://doi.org/10.1088/1755-1315/440/4/042031.
31. Xu, G., Qiu, S., Ahmad, H., Xu, G., Guo, Y., Zhang, M. and Xu, H. (2018). A Multi-Server Two-Factor Authentication Scheme with Un-Traceability Using Elliptic Curve Cryptography. *Sensors*, 18(7), p.2394. doi:https://doi.org/10.3390/s18072394.