5Graphical user interface, application

Description automatically generated

**Staffordshire University London**

**Penetration Testing (COCS71151)**

**Assignment Specification**

**Weighted at 50% of the module mark**

Student Number: D013868L

Student Name: Salvatore Di Salvo

Due DATE: 21st March 2023

WEIGHTAGE: 50%

# TABLE OF CONTENTS

[TABLE OF CONTENTS 2](#_Toc135463672)

[LIST OF FIGURES 4](#_Toc135463673)

[Part A 5](#_Toc135463674)

[Introduction 5](#_Toc135463675)

[Kali Linux OS 5](#_Toc135463676)

[Kali Linux Networking Tools 5](#_Toc135463677)

[Parrot OS 6](#_Toc135463678)

[Parrot’s Tools 6](#_Toc135463679)

[Pen-Testing Methodology 8](#_Toc135463680)

[Penetration Testing in Practice 9](#_Toc135463681)

[Information Gathering & Scanning 9](#_Toc135463682)

[Exploitation & Post-Exploitation 12](#_Toc135463683)

[Executive Summary 22](#_Toc135463684)

[Part B 23](#_Toc135463685)

[Introduction 23](#_Toc135463686)

[Securing the Vulnerabilities 23](#_Toc135463687)

[FTP 23](#_Toc135463688)

[NFS 23](#_Toc135463689)

[SSH 24](#_Toc135463690)

[HTTP 24](#_Toc135463691)

[RPCBIND 24](#_Toc135463692)

[Securing the Apache Server 25](#_Toc135463693)

[Security Baseline 26](#_Toc135463694)

[Executive Summary 27](#_Toc135463695)

[REFERENCES 28](#_Toc135463696)

**LIST OF FIGURES**

[Figure 1 - NMap scanning the IP address and expressing all the open ports. 7](#_Toc135554062)

[Figure 2 - The vulnerabilities with the IP address after OpenVAS has completed it's scan of the target host. 8](#_Toc135554063)

[Figure 3 - In-depth information of the vulnerabilities identified by OpenVAS. 8](#_Toc135554064)

[Figure 4 - OpenVAS expressing all the vulnerabilities with the target host after scanning it. 10](#_Toc135554065)

[Figure 5 - In-depth information regarding the vulnerabilities discovered by OpenVAS after scanning the target host 11](#_Toc135554066)

[Figure 6 - NMap's array of scanning commands being utilized to express all the open port/vulnerabilities with the target host. 11](#_Toc135554067)

[Figure 7 - A custom Python script scanning and expressing the target host's open ports. 12](#_Toc135554068)

[Figure 8 - The code for the custom script in which ports are scanned from the target IP. 12](#_Toc135554069)

[Figure 9 - Expresses the infiltration of the FTP server by using the username 'anonymous' (as expressed by OpenVAS), and downloading the files found on the server into the host machine. 13](file:///C:\Users\salva\OneDrive%20-%20Staffordshire%20University\School\Year%202\Ethical%20Hacking%20(EH)\Sem2\ASSIGNMENTS\Assignment\EthicalHacking%20Assignment%20DRAFT.docx#_Toc135554070)

[Figure 10 – A new directory was created in order to connect it to the target IP. By using the NFS command, the attacker could connect to the target IP and mount all of the data to the host machine, allowing the attacker to then explore the target IP's files. 14](#_Toc135554071)

[Figure 11 - As the attacker looks further into the files, a message is discovered expressing the username and password for the remote machine, otherwise known as the SSH server. However the password is encrypted by MD5 hash. 15](#_Toc135554072)

[Figure 12 - By utilizing hashcat brute forcer, the hash was eventually cracked into the password 'toortoor1'. 16](#_Toc135554073)

[Figure 13 - The hash was inputted into an online de-crypter, which expressed the same 'toortoor1' as the password. 16](#_Toc135554074)

[Figure 14 - The username and password found by using NFS was used to log into the SSH server and it was successful. 17](#_Toc135554075)

[Figure 15 - By using Burp Suite, and attempting to log into the Apache HTTP server on the site hosted by Burp, the 'GET REQUEST' was captured. 18](#_Toc135554076)

[Figure 16 - The 'GET REQUEST' was saved into a file so it could be read by the SQLmap 19](#_Toc135554077)

[Figure 17 - SQLmap is being utilized by having it read the 'GET REQUEST' file and discover any exploits to bypass the server login. 20](#_Toc135554078)

[Figure 18 - SQLmap has discovered all the exploits and has produced payloads which can be used to allow the attacker to bypass the login page and access the HTTP Apache server. 20](#_Toc135554079)

[Figure 19 - The payload is being injected into the 'username' parameter and any random character can be inputted into the 'password' parameter. 21](#_Toc135554080)

[Figure 20 - The HTTP server was successfully bypassed and infiltrated. 22](#_Toc135554081)

[Figure 21 - Metasploit is being used with a specific module in order to spam Port 111 with a surplus of packets. 23](#_Toc135554082)

[Figure 22 - After being overwhelmed with data packets, a DoS effect occurs, causing the target host to become unresponsive temporarily. 24](#_Toc135554083)

# Part A

## Introduction

The field of ethical hacking and cybersecurity will be investigated in this report and the knowledge and abilities from this field will be put to the test. From a specific IP address , this being 172.17.4.55, an attempt to infiltrate, hack, and access the system will be made and any vulnerabilities will be discovered and expressed in depth. Through this exercise, many methods and equipment that threat actors employ to compromise systems, gain unauthorized access, and take advantage of vulnerabilities will be researched and utilized. To guarantee that our system or the system is not negatively impacted during this project, ethical and controlled procedures will be employed and exercised.

## Kali Linux OS

Cybersecurity experts, ethical hackers, and penetration testers frequently utilize Kali Linux as their preferred operating system for conducting security audits and vulnerability testing. The Kali Linux Operating system will be utilized for its plethora of cybersecurity and hacking tools which can be used to gain access to the target system. (Lu, H. J., & Yu, Y., 2021)

### Kali Linux Networking Tools

Every penetration testing or hacking operation requires the use of networking tools. These techniques enable a security professional to map out the network topology, identify network services, and identify network infrastructure vulnerabilities. A threat actor can adopt these same tools for the same reasons but for an attack; this being a reconnaissance attack. (Asaad, R. R., 2021)

Networking software’s such as Nmap and Wireshark can be used to scan a network for open ports, services, and operating systems. This data can then be used to assess which devices are vulnerable to compromise. Once vulnerable systems have been found, the next step is to exploit those vulnerabilities to get access to the system. Metasploit and other networking tools can be used to exploit known vulnerabilities in operating systems and applications. Metasploit is a sophisticated tool that includes a large library of exploits and payloads for exploiting vulnerabilities in target systems. (Khawaja, G., 2021)

A picture containing text, screenshot, software, display

Description automatically generated

Figure - NMap scanning the IP address and expressing all the open ports.

## Parrot OS

Parrot OS is a security-focused operating system based on Debian Linux that is intended for penetration testing, digital forensics, and privacy protection due to its pre-installed tools for network analysis, vulnerability evaluation, and exploitation. The tools that come pre-installed with Parrot OS range from network scanners and vulnerability assessment tools to password crackers and exploit frameworks, giving users a comprehensive arsenal for testing and securing computer systems and networks. As a result, it is an excellent resource for penetration testing, allowing testers to uncover and attack vulnerabilities in a variety of situations. (ul Hassan, S. Z., Muzaffar, Z., et al., 2021)

### Parrot’s Tools

OpenVAS is a robust vulnerability scanning tool that comes pre-installed with Parrot OS and can be used to thoroughly scan IP addresses. It is a vulnerability screening tool for detecting security flaws in computer systems and networks. It operates by scanning IP addresses and servers for known vulnerabilities and producing reports outlining any security threats discovered. Security professionals can use OpenVAS to do in-depth vulnerability scans, detect potential security concerns, and take steps to mitigate them. The program is adaptable and powerful for securing computer systems and networks since it is customizable and provides a variety of scan choices. (Qureshi, S., He, J., Tunio, S., Zhu, N., et al., 2022)

A screenshot of a computer

Description automatically generated

Figure - The vulnerabilities with the IP address after OpenVAS has completed it's scan of the target host.

A screenshot of a computer

Description automatically generated

Figure - In-depth information of the vulnerabilities identified by OpenVAS.

## Pen-Testing Methodology

A pen-testing methodology is a structured strategy used by cybersecurity professionals to do penetration testing, which is the process of simulating an attack on a computer system or network in order to identify security weaknesses. (Velu, V. K., & Beggs, R., 2019)

1. Information Gathering:

* Gather information about the target host's infrastructure, network architecture, and operating systems.
* Use tools such as Nmap, Recon-ng, and Google dorks for scanning the target's IP ranges, DNS information, WHOIS data, and open ports.
* Collect information from publicly available sources such as social media, online forums, and search engines to gather information about the target's employees, technologies, and potential vulnerabilities.

1. Scanning:

* Conduct vulnerability scans using tools like Nessus, OpenVAS, and Qualys to identify potential vulnerabilities in the target's systems and applications.
* Perform network scans using tools like Nmap and Netcat to identify open ports, services, and potential attack vectors.
* Conduct web application scans using tools like Burp Suite, OWASP ZAP, and Nikto to identify vulnerabilities in web applications and APIs.

1. Vulnerability Classification:

* Classify the identified vulnerabilities based on their severity and impact on the target's systems and applications.
* Use Common Vulnerability Scoring System (CVSS) to assess the severity of vulnerabilities and prioritize them for further exploitation.
* Categorize vulnerabilities into critical, high, medium, and low severity levels based on their potential impact and likelihood of exploitation.

1. Exploitation:

* Exploit identified vulnerabilities using tools like Metasploit, ExploitDB, and custom scripts to gain unauthorized access to the target's systems and applications.
* Attempt to escalate privileges, gain access to sensitive data, and pivot to other systems within the target's network.
* Document and report any successful exploits and unauthorized access to the target's systems and applications.

1. Post-Exploitation:

* Conduct further reconnaissance and enumeration to gather additional information about the target's systems, users, and applications.
* Use tools like PowerShell Empire, Cobalt Strike, and BloodHound for post-exploitation activities such as lateral movement, privilege escalation, and persistence.
* Identify and exploit any misconfigurations, weak passwords, or other vulnerabilities to gain further access and control over the target's systems.

1. Reporting:

* Document all findings, including vulnerabilities, exploits, and unauthorized access, in a detailed report with clear explanations, evidence, and recommendations.
* Include screenshots, logs, and other supporting materials to provide a comprehensive report to the target's management and IT team.
* Clearly outline the potential impact and severity of the vulnerabilities.

Steps 1 to 5 will be expressed in Part A of this document, whilst step 6 will be expressed in Part B.

## Penetration Testing in Practice

### Information Gathering & Scanning

The first steps of the pen-testing methodology were used to scan and identify any vulnerabilities with the IP’s network. The software’s and tools that were utilized to perform this were; OpenVAS, NMap and a personal Python script which scanned for open ports. (Wang, J., Gao, Y., Liu, W., et al., 2019)

A screenshot of a computer

Description automatically generated

Figure - OpenVAS expressing all the vulnerabilities with the target host after scanning it.

A screenshot of a computer

Description automatically generated

Figure - In-depth information regarding the vulnerabilities discovered by OpenVAS after scanning the target host

A computer screen shot of a program

Description automatically generated with low confidence

Figure - NMap's array of scanning commands being utilized to express all the open port/vulnerabilities with the target host.

A screenshot of a computer program

Description automatically generated with medium confidence

Figure - A custom Python script scanning and expressing the target host's open ports.

A screenshot of a computer program

Description automatically generated with medium confidence

Figure - The code for the custom script in which ports are scanned from the target IP.

Figure 7 express that the same open ports; 21, 22, 23, 53, 80, 111, 2049, 3128 were found to be open. These ports being open mean they are vulnerable to potential threat actors and hence, for the sake of the practical, these ports were exploited and infiltrated.

### Exploitation & Post-Exploitation

After discovering the open ports, numerous methods and software’s were employed to exploit them, and the exploitation of the particular ports and services will be expressed below.( Harper, A., Linn, R., Sims, et al., 2022)

#### FTP / Port 21

As expressed in Figure 5, FTP could be infiltrated by logging in as ‘anonymous’ and gaining entry to the IP’s server. This will offer the opportunity to steal sensitive data or upload malware that can be used to hack other network systems or use the server to conduct subsequent attacks against other targets.

**A screenshot of a computer screen

Description automatically generated with medium confidence**

Figure - Expresses the infiltration of the FTP server by using the username 'anonymous' (as expressed by OpenVAS), and downloading the files found on the server into the host machine.

Figure 9 is expressing how the IP’s FTP was breached, allowing the cyber security specialist to not only enter the FTP server, but to explore it and even download files from the server.

#### NFS / Port 2049

NFS is a protocol that allows users to access files and directories on remote servers as if they were on their own computer, but the misconfiguration of this can result in threat actors exploiting the service to gain access to sensitive information. (Lever, C., & Noveck, D., 2019)

A picture containing text, screenshot, software, multimedia software

Description automatically generated

Figure – A new directory was created in order to connect it to the target IP. By using the NFS command, the attacker could connect to the target IP and mount all of the data to the host machine, allowing the attacker to then explore the target IP's files.

A screenshot of a computer screen

Description automatically generated with medium confidence

Figure - As the attacker looks further into the files, a message is discovered expressing the username and password for the remote machine, otherwise known as the SSH server. However the password is encrypted by MD5 hash.

After successfully exploiting the NFS service and gaining unauthorized access to the IP’s files, investigative work could be done and the user and password for a shared machine was discovered. This means that the login details for the SSH had been discovered. However, the password must have been encrypted by being hashed by MD5, since it’s consisting of 32 hex digits.

#### Hashcat for Password Decryption (MD5)

Since the password needed to be decrypted by MD5, hashcat was installed and used to do this.

A picture containing text, electronics, screenshot, software

Description automatically generated

Figure - By utilizing hashcat brute forcer, the hash was eventually cracked into the password 'toortoor1'.

Hashcat expressed that the decrypted password was ‘toortoor1’, however, the password was also inserted into an online MD5 hasher in order for it to be decrypted so that the password could be deemed reliable.

A screenshot of a computer

Description automatically generated

Figure - The hash was inputted into an online de-crypter, which expressed the same 'toortoor1' as the password.

The online MD5 hasher expresses the same information as hashcat, meaning that the password for SSH is truly ‘toortoor1’.

#### SSH / Port 22

Although neither a tool or software was utilized in order to breach the SSH server, it was still infiltrated due to the breach of NFS and the investigative efforts of a cyber security specialist.

A screenshot of a computer

Description automatically generated

Figure - The username and password found by using NFS was used to log into the SSH server and it was successful.

Furthermore, simply logging in with the username and password acquired in the earlier steps, SSH was successfully breached, allowing a potential threat actor to gain unauthorized access to the IP server’s files.

A "man-in-the-middle" (MitM) attack is another way in which gaining access to the SSH server could result in subsequent malicious attacks. This attack occurs when an attacker intercepts and alters network traffic between the SSH client and the server. A MitM attack can be used by an attacker to intercept or change sensitive data sent between the client and the server, or to introduce malicious code into a system. (Ylonen, T., 2019)

#### HTTP / Port 80 – SQL Injection

HTTP is the protocol used to transmit web pages and other web content over the internet, and a common HTTP-related issue is known as a "SQL injection" attack. An attacker conducts this form of attack by exploiting vulnerabilities in a web application to insert malicious SQL commands into the application's database. An SQL injection attack can be used to steal sensitive information, modify or remove data, or carry out other malicious actions. (Kim, J., 2020)

Before this attack can occur, a ‘GET REQUEST’ for the specific IP database must be created. This was successfully created by using Burp Suite on Kali Linux and stored into a text file.

A screenshot of a computer

Description automatically generated

Figure - By using Burp Suite, and attempting to log into the Apache HTTP server on the site hosted by Burp, the 'GET REQUEST' was captured.

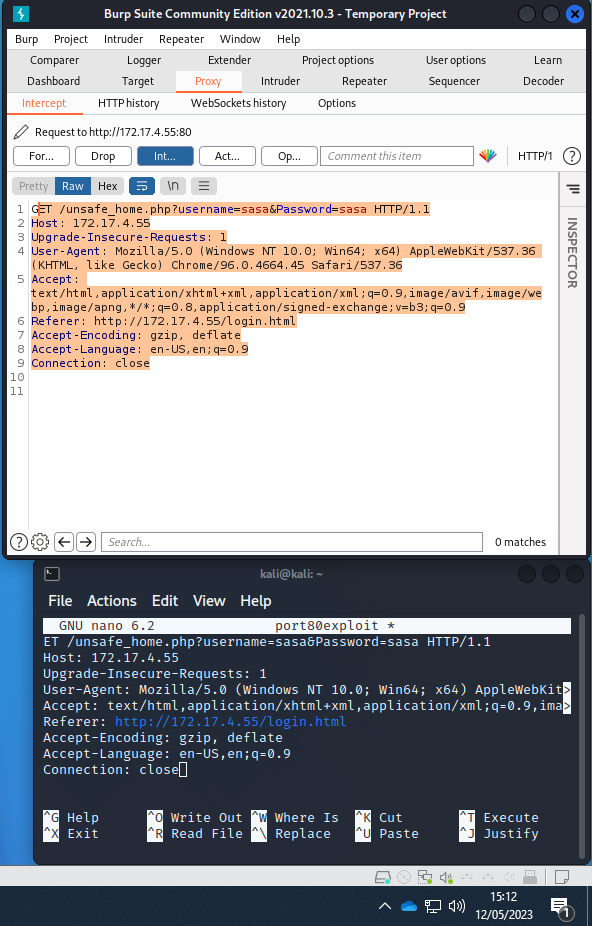


Figure - The 'GET REQUEST' was saved into a file so it could be read by the SQLmap

A screenshot of a computer screen

Description automatically generated with medium confidence

Figure - SQLmap is being utilized by having it read the 'GET REQUEST' file and discover any exploits to bypass the server login.

A screenshot of a computer

Description automatically generated

Figure - SQLmap has discovered all the exploits and has produced payloads which can be used to allow the attacker to bypass the login page and access the HTTP Apache server.

A screenshot of a computer

Description automatically generated

Figure - The payload is being injected into the 'username' parameter and any random character can be inputted into the 'password' parameter.

Once the file had been created, a unique and specific SQL-injection payload had to be created by using SQLMap on Kali Linux. With the creation of the payloads, they could be inserted into the ‘username’ section, allowing us to completely bypass the login process and gain unauthorized access to the private part of the IP’s website. (Baklizi, M., Atoum, I., Abdullah, N., et al., 2022)

A screenshot of a computer

Description automatically generated

Figure - The HTTP server was successfully bypassed and infiltrated.

#### RPCBIND / Port 111 – DoS Attack

RPC is a protocol that allows one program on a remote system to request a service from another program on the same system. It's often used in networked contexts for things like file sharing and remote system administration. A "buffer overflow" attack is another RPC-related exploit. When an attacker gives more data to a program or service than it can process, the program or service crashes or executes arbitrary code. Metasploit was used with the particular module ‘rpcbomb’ to encumber the port with an extreme amount of data and subsequently create a Denial of Service (DoS). However, an attacker can even exploit an RPC service that is vulnerable to buffer overflow attacks to execute malicious code on the server and potentially acquire control of the system. (Hara, K., Sato, T., Imamura, M., et al., 2020)

A screenshot of a computer

Description automatically generated

Figure - Metasploit is being used with a specific module in order to spam Port 111 with a surplus of packets.

A screen shot of a computer

Description automatically generated with medium confidence

Figure - After being overwhelmed with data packets, a DoS effect occurs, causing the target host to become unresponsive temporarily.

As expressed in Figure 22, the IP was encumbered with data, so it could not respond to any more data for a temporary amount of time, showing a DoS attack being created from an rpcbind attack.

## Executive Summary

Infiltration into the IP’s servers were successful, meaning that the IP was successfully compromised. The blend of reconnaissance, port exploitations and various other attacks through the use of software’s and tools from varying operating systems allowed this outcome to be achieved. The fact that the research was applied into a genuine scenario, with the figures being proof of this, express the power of these methods as well as the importance of configuring and securing servers; a concept that will be thoroughly researched in Part B of this document.

# Part B

## Introduction

This paper investigates the techniques and best practices for efficiently securing a target host, which include strong access restrictions, robust authentication mechanisms, frequent patching and updates, advanced monitoring and detection systems, user education, and developing a security culture. Organizations and people may reinforce their defenses against cyber attacks, preserve key assets, and ensure a safer digital future by employing a comprehensive strategy that includes both technological measures and human behavior. Additionally, the C.I.A (confidentiality, integrity and availability respectively) security policies will be discussed.

## Securing the Vulnerabilities

### FTP

An array of methods will need to be performed to avoid FTP exploits and improve the security of FTP servers. First, secure authentication should be enforced through the use of strong, unique passwords and the use of multi-factor authentication (MFA). It is critical to use secure FTP protocols that encrypt data transfer, such as FTPS or SFTP. Limiting access and permissions, upgrading and patching the FTP server software on a regular basis, and using a firewall and network segmentation all assist to avoid attacks. Continuous security is ensured via intrusion detection and monitoring systems, as well as frequent vulnerability assessments and penetration testing. Finally, user education and training on safe FTP practices are critical in limiting risks and creating a secure workplace. The C.I.A security principle that is violated is confidentiality, since it was not configured properly, FTP could be exploited and infiltrated by inputting ‘anonymous’, which is a default password. (Singh, S. P., 2019)

### NFS

Several safeguards may be put in place to avoid NFS (Network File System) attacks. First, safeguard the NFS server by setting authentication measures to ensure only authorized access. Implement tight access restrictions and limit the rights allowed to NFS clients. Use firewall rules and network segmentation to regulate NFS traffic and limit access to trustworthy computers. Encrypt data in transit by utilizing NFS over a secure protocol like NFSv4 with Kerberos. Regularly update and patch the NFS server software to resolve any known vulnerabilities. Implement monitoring and logging to detect and respond to any suspicious activities or unauthorized access attempts. Finally, do frequent security assessments and penetration testing to discover and repair any flaws or vulnerabilities in the NFS server setup. NFS’ C.I.A principle that was violated is Integrity, when the threat actor can simply connect to the target IP with a ‘mount’ command, it expresses that the NFS was never configured appropriately. (Riasetiawan, M., & Amien, N., 2022)

### SSH

Multiple steps should be applied to avoid SSH (Secure Shell) attacks. First, enforce robust authentication by requiring complicated, unique passwords or pass-based authentication. SSH keys, rather just passwords, can give even greater authentication. It is critical to deactivate or limit SSH access to any unused accounts. Update the SSH server software on a regular basis to fix any known vulnerabilities. Set up firewall rules to restrict SSH access and only allow connections from trusted sources. Monitoring SSH logs can aid in the detection of unusual activity and potential attacks. Finally, teaching users on safe SSH practices such as not disclosing keys or credentials and adopting secure connection protocols helps to prevent SSH vulnerabilities generally. Likewise to FTP, the C.I.A principle that’s violated is Confidentiality since the SSH was exploited by discovering a password in a file that should have been deleted altogether.

### HTTP

Several preventive measures may be applied to avoid HTTP exploits and improve the security of online applications. To begin, make certain that web servers and apps are frequently updated with the most recent security patches and upgrades. Use safe coding practices to protect against common vulnerabilities like cross-site scripting (XSS) and SQL injection attacks. To avoid fraudulent data entry, enforce tight input validation. Use secure communication protocols, such as HTTPS, to encrypt data in transit and prevent interception. To restrict access to critical resources, utilize access restrictions and user authentication techniques. To identify and stop malicious traffic, use web application firewalls (WAFs). Monitor and analyze web server logs on a regular basis for evidence of unusual behavior. To detect and mitigate vulnerabilities, conduct frequent security assessments and penetration testing. The C.I.A principle violated here is Integrity, since infiltration into the HTTP server was committed by tampering with the server’s username and password database via SQL Injection. ((Nance, B. P., & Bengston, A. S., 2020))

### RPCBIND

Several precautions can be made to avoid RPCbind exploitation. To begin, it is critical to maintain the RPCbind service and associated software up to current with the most recent security patches and upgrades. Patching on a regular basis helps to fix known vulnerabilities and guard against possible attacks. It is also critical to configure correct firewall rules to restrict access to the RPCbind service. The attack surface can be limited by restricting access to only trustworthy networks or specified IP addresses. To reduce possible vulnerabilities, it is also advised to eliminate superfluous RPC services and protocols. Monitoring and recording RPCbind activity can aid in the detection of any unusual behavior or unauthorized access attempts. Unlike the others, the C.I.A principle that was violated here is Availability, because the exploit committed was a DoS attack, which renders server’s temporarily unresponsive.

## Securing the Apache Server

The following methods can be applied to harden and secure an Apache server:

* Maintain the Apache server software with the most recent security patches and upgrades. Check for and deploy updates to address known vulnerabilities on a regular basis
* Disable or delete any superfluous Apache modules: Disable or remove any unnecessary Apache modules. The smaller the attack surface and possible weaknesses, the fewer modules deployed.
* Secure configuration: In order to ensure secure settings, review and alter the server configuration files. Setting suitable rights and ownership for files and directories, deactivating directory listing, and enabling secure SSL/TLS protocols are all part of this.
* Strong access restrictions should be implemented to restrict server access. Authentication techniques such as username/password and certificate-based authentication should be used. Access limits can be set via utilizing tools such as .htaccess files or Apache configuration directives.
* Use a firewall and network segmentation to manage incoming and outgoing traffic to the Apache server. Only allow access to the appropriate ports and protocols. Consider network segmentation to keep the server apart from other critical systems.
* Logging and monitoring: Enable logging for Apache server operations and check the logs on a regular basis for any suspicious or malicious behaviour. To identify and respond to prospective threats, use intrusion detection and prevention systems (IDS/IPS).
* DDoS mitigation: Take steps to reduce Distributed Denial of Service (DDoS) assaults. This includes the use of load balancers, rate restriction, and the deployment of specialized DDoS protection services.
* Backup the Apache server settings, website files, and databases on a regular basis. This guarantees that data may be recovered in the case of a security compromise or system failure.
* Security headers: To give further protection, include security headers in Apache's HTTP response headers. Cross-site scripting (XSS) attacks, clickjacking, and other online vulnerabilities can be mitigated by using these headers.
* Security audits on a regular basis: Conduct periodic security audits and vulnerability assessments on the Apache server. This aids in identifying and addressing any security flaws or vulnerabilities in the system.

(Piantadosi, V., Scalabrino, S., & Oliveto, R., 2019) (Olenčin, M., & Perháč, J., 2019)

By employing these techniques, an Apache server's security posture may be greatly enhanced, lowering the danger of attacks and unauthorized access. However, it is critical to be cautious, follow best practices in security, and adjust security measures when new threats develop.

## Security Baseline

This is a rather simple security baseline that can be employed by any individual and organization to secure their network/devices.

* Strong Password Policy: Implement a strong password policy that forces users to establish complicated passwords, update them on a regular basis, and prevent reusing passwords across several platforms.
* Role-based access control (RBAC) should be used to guarantee that users have appropriate rights based on their job responsibilities. Review and remove unneeded rights on a regular basis.
* Patching and updating on a regular basis: Keep all software, operating systems, and apps up to date with the most recent security patches and upgrades. If feasible, use an automated patch management system.
* Configure and manage firewalls to regulate incoming and outgoing traffic, preventing unauthorized access and harmful network activities.
* Antivirus and Malware Protection: To identify and prevent malware infestations, install and frequently update antivirus and anti-malware software on all platforms.
* Data Backup and Recovery: Create frequent backups of key data and a recovery strategy to reduce data loss in the case of a security incident or system failure.
* Conduct security awareness training programs to educate staff on common security dangers, social engineering, and best practices for securing sensitive information.
* Encryption: Encrypt sensitive data in transit and at rest using encryption methods. This involves encrypting stored data and deploying SSL/TLS for secure communication.
* Intrusion Detection and Monitoring: Put intrusion detection and monitoring systems in place to detect and respond to possible security breaches or suspicious activities. Examine logs and network data on a regular basis.

These recommendations provide a solid foundation for developing a good security baseline. However, it is critical to consider particular sector standards, risk assessments, and your organization's unique demands when customizing and improving security measures.

## Executive Summary

The article focuses at manners for protecting an Apache server and strengthening weaknesses, with an emphasis on the concepts of confidentiality, integrity, and availability (C.I.A). With the threat landscape becoming more complex, it is vital for organizations to implement comprehensive security measures to protect their systems and key assets. The article investigates how the C.I.A principles might lead security efforts by preserving the confidentiality of sensitive data, information integrity, and service availability. It also goes through particular methods for protecting an Apache server, such as performing regular updates and patching, setting access limits, installing secure SSL/TLS protocols, and monitoring server activities. organizations may improve their security posture, eliminate vulnerabilities, and guard against possible attacks on their Apache server by using these practices.

**REFERENCES**

Lu, H. J., & Yu, Y. (2021). Research on WiFi penetration testing with Kali Linux. *Complexity*, *2021*, 1-8.

Asaad, R. R. (2021). Penetration testing: Wireless network attacks method on Kali Linux OS. Academic Journal of Nawroz University, 10(1), 7-12.

Khawaja, G. (2021). *Kali Linux penetration testing bible*. John Wiley & Sons.

ul Hassan, S. Z., Muzaffar, Z., & Ahmad, S. Z. (2021). Operating Systems for Ethical Hackers-A Platform Comparison of Kali Linux and Parrot OS. *International Journal*, *10*(3).

Qureshi, S., He, J., Tunio, S., Zhu, N., Akhtar, F., Ullah, F., ... & Wajahat, A. (2022). Browser Forensics: Extracting Evidence from Browser Using Kali Linux and Parrot OS Forensics Tools. *International Journal of Network Security*, *24*(3), 557-572.

Velu, V. K., & Beggs, R. (2019). Mastering Kali Linux for Advanced Penetration Testing: Secure your network with Kali Linux 2019.1–the ultimate white hat hackers' toolkit. Packt Publishing Ltd.

Wang, J., Gao, Y., Liu, W., Wu, W., & Lim, S. J. (2019). An asynchronous clustering and mobile data gathering schema based on timer mechanism in wireless sensor networks. *Comput. Mater. Contin*, *58*(3), 711-725.

Harper, A., Linn, R., Sims, S., Baucom, M., Fernandez, D., Tejeda, H., & Frost, M. (2022). *Gray hat hacking: the ethical hacker's handbook*. McGraw-Hill Education.

Lever, C., & Noveck, D. (2019). *NFS Version 4.0 Trunking Update* (No. rfc8587).

Ylonen, T. (2019). SSH Key management challenges and requirements. In 2019 10th IFIP International Conference on New Technologies, Mobility and Security (NTMS) (pp. 1-5). IEEE.

Kim, J. (2020). *Burp suite: Automating web vulnerability scanning* (Doctoral dissertation, Utica College).

Baklizi, M., Atoum, I., Abdullah, N., Al-Wesabi, O. A., Otoom, A. A., & Hasan, M. A. S. (2022). A Technical Review of SQL Injection Tools and Methods: A Case Study of SQLMap. *International Journal of Intelligent Systems and Applications in Engineering*, *10*(3), 75-85.

Hara, K., Sato, T., Imamura, M., & Omote, K. (2020). Profiling of Malicious Users Targeting Ethereum’s RPC Port Using Simple Honeypots. In *2020 IEEE International Conference on Blockchain (Blockchain)* (pp. 1-8). IEEE.

Singh, S. P. (2019). Security configuration and performance analysis of ftp server. *International Journal of Communication and Computer Technologies*, *2*(2), 106-106.

Riasetiawan, M., & Amien, N. (2022). Random and Sequence Workload for Web-Scale Architecture for NFS, GlusterFS and MooseFS Performance Enhancement. *International Journal of Advanced Computer Science and Applications*, *13*(3).

Nance, B. P., & Bengston, A. S. (2020). *An Integrated Two-Factor Authentication Solution Using Pulse Connect Secure and Apache HTTP Server* (No. ORNL/TM-2020/1324). Oak Ridge National Lab.(ORNL), Oak Ridge, TN (United States).

Piantadosi, V., Scalabrino, S., & Oliveto, R. (2019). Fixing of security vulnerabilities in open source projects: A case study of apache http server and apache tomcat. In *2019 12th IEEE Conference on software testing, validation and verification (ICST)* (pp. 68-78). IEEE.

Olenčin, M., & Perháč, J. (2019). Automated configuration of a Linux web server security. In *2019 IEEE 15th International Scientific Conference on Informatics* (pp. 000491-000496). IEEE.