Nikto Security Scan Report for Metasploitable

Target: 192.168.56.101

Port: 80

Scan Version: Nikto v2.5.0

Findings Summary:

The scan revealed several potential security issues, misconfigurations, and exposed sensitive information on the target server. Below is a detailed breakdown of the vulnerabilities and misconfigurations identified.

1. Outdated Software

- PHP Version: The server is running PHP 5.2.4, which is outdated and may have unpatched security vulnerabilities. It is strongly recommended to upgrade PHP to a supported version.

2. Missing Security Headers

- X-Frame-Options: The server does not send the X-Frame-Options header. This can expose the site to Clickjacking attacks.

Recommendation: Set this header to DENY or SAMEORIGIN to prevent embedding the site in frames.

- X-Content-Type-Options: The server lacks the X-Content-Type-Options header, which prevents MIME-type sniffing.

Recommendation: Set this header to nosniff to ensure content is rendered according to its declared MIME type.

3. Web Server Misconfigurations

- Apache Version: The server is running Apache/2.2.8, which is outdated and has reached its end of life.

Recommendation: Upgrade to a supported version, at least Apache/2.4.54.

- Apache mod_negotiation: The multi-views feature of Apache mod_negotiation is enabled. This feature can be exploited to perform file name brute-force attacks.

Recommendation: Disable the multi-views option unless absolutely necessary.

4. **Vulnerable HTTP Methods**

- TRACE Method Active: The HTTP TRACE method is enabled, which may allow Cross-Site Tracing (XST) attacks.

Recommendation: Disable the TRACE method on the web server to prevent XST attacks.

5. **Sensitive Information Exposure**

- phpinfo() Script Found: The phpinfo.php file was found, which reveals sensitive server and PHP configuration information.

Recommendation: Remove or restrict access to phpinfo() scripts to authorized users only.

- PHP Query Strings Exposing Sensitive Information: Several PHP query strings such as PHPE9568F36-D428-11d2-A769-00AA001ACF42 were found, which can expose configuration details or sensitive data.

Recommendation: Avoid exposing sensitive information in URLs. Use alternative methods to handle sensitive data.

6. **Directory Listings Enabled**

- Directory Indexing Found: Multiple directories such as /doc/, /test/, and /icons/ have directory listings enabled, which could expose sensitive files and system information.

Recommendation: Disable directory indexing on the web server to prevent unauthorized access to directory contents.

7. **Exposed phpMyAdmin**

- phpMyAdmin Found: The phpMyAdmin directory is exposed, which is used for managing MySQL databases.

Recommendation: Restrict access to phpMyAdmin by IP address or use strong authentication mechanisms. phpMyAdmin should not be exposed to the public internet.

8. **Exposed WordPress Configuration File**

- wp-config.php File Found: The wp-config.php file, which contains sensitive WordPress configuration information such as database credentials, was found.

Recommendation: Ensure that the wp-config.php file is properly protected and not accessible via the web.

9. **Miscellaneous Findings**

- Uncommon Header tcn Found: The server returned an uncommon header tcn with the value list.

While this is not necessarily a vulnerability, it may indicate non-standard server configurations or potential security risks.

- Server Returns Valid Response with Junk HTTP Methods: The server returns valid responses to non-standard HTTP methods, which may cause false positives in security scanners but should still be reviewed for security risks.

Conclusion and Recommendations:

The scan has identified several areas of concern that need to be addressed to improve the security of the target web server:

1. Software Updates: Upgrade PHP and Apache to supported versions.
2. Security Headers: Implement necessary security headers to protect against Clickjacking and MIME sniffing attacks.
3. Configuration Hardening: Disable unnecessary HTTP methods, particularly TRACE, and review Apache configurations.
4. Sensitive Data Exposure: Restrict access to phpMyAdmin, remove phpinfo() scripts, and protect configuration files like wp-config.php.
5. Directory Listings: Disable directory indexing to prevent unauthorized browsing of server directories.