

***Vulnerability Assessment Report***

***Introduction***

*This report aims to outline and evaluate the security weaknesses identified by scanning tools for networks and web applications such as Nmap, Nikto, and OpenVAS. These instruments were utilized to detect accessible ports, unprotected services, and crucial vulnerabilities, which might be taken advantage of by malicious perpetrators. The objective of this evaluation is to offer an in-depth analysis of the security status of the examined system and suggest ways to address the identified risks.*

***Scope***

*The assessment of vulnerabilities was performed on the target system, which included analyzing both the network infrastructure and web application components. The evaluation involved the scans listed below:  
  
Nmap is used for network mapping and port scanning in order to detect open ports and the services running on them.  
  
Nikto is a tool utilized for detecting server misconfigurations, outdated software, and exposed directories on websites.  
  
OpenVAS provides thorough vulnerability scanning for identifying high and medium severity vulnerabilities throughout the system.*

***Overview***

*The evaluation identified various issues within both the network and web application elements. Multiple open ports were found running services that could potentially present a security threat. Furthermore, Nikto discovered exposed folders and obsolete software versions on the web server. The OpenVAS scan revealed several high and medium severity vulnerabilities that need to be addressed promptly.*

***Results***

1. ***Nmap Scan***
2. ***Screenshot***

A screenshot of a computer screen

Description automatically generated

A computer screen shot of a computer screen

Description automatically generated

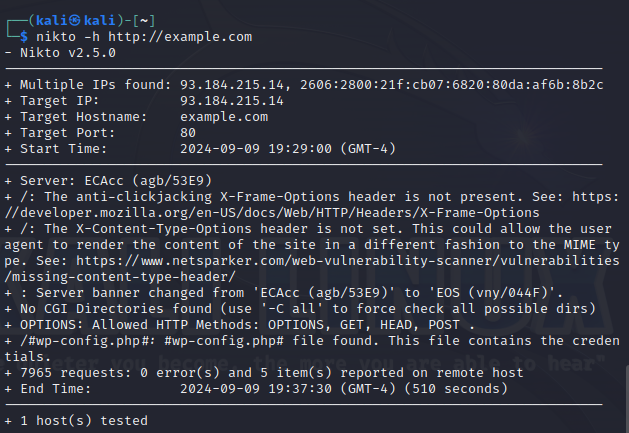
A computer screen shot of a computer code

Description automatically generated

A computer screen shot of a computer code

Description automatically generated

1. ***Key Vulnerabilities:***
2. *Outdated Apache:*
3. *CVE-2019-0211: Apache is vulnerable to privilege escalation attacks, where a local user could gain elevated privileges.*
4. *CVE-2018-17199: Potential for a denial-of-service (DoS) attack due to improper handling of HTTP/2 requests.*
5. *Weak SSL/TLS Configuration: Outdated versions may lack modern TLS settings, allowing the use of deprecated protocols like TLS 1.0 or 1.1.*
6. ***Potential Impact:***
7. *Port 80 (HTTP): Running HTTP without encryption exposes user data to interception, increasing the risk of man-in-the-middle (MITM) attacks. Sensitive information, such as login credentials or session tokens, can be captured if transmitted over HTTP.*
8. *Port 443 (HTTPS): The use of an outdated version of Apache (2.4.29) increases the risk of known exploits being used to compromise the server. Attackers could use privilege escalation vulnerabilities to take over the system, execute arbitrary code, or cause a denial of service, disrupting services.*
9. ***Recommended Mitigations:***
10. *Update Apache: Upgrade Apache to the latest stable version (currently 2.4.54 or higher), which addresses the vulnerabilities in the older versions.*
11. *Enforce HTTPS (Port 443):*
12. *Redirect all HTTP traffic (Port 80) to HTTPS (Port 443) using strict HTTPS policies, such as HTTP Strict Transport Security (HSTS).*
13. *Ensure SSL/TLS configurations are up-to-date, enforcing modern encryption protocols like TLS 1.2 or higher, and disabling deprecated versions (TLS 1.0, 1.1).*
14. *Harden Apache Configuration:*
15. *Disable or remove unnecessary modules that could increase the attack surface.*
16. *Implement rate-limiting to mitigate potential DoS attacks on the web server.*
17. *Secure SSL/TLS Configuration:*
18. *Review and configure SSL/TLS certificates to use secure ciphers and protocols.*
19. *Use tools such as SSL Labs to test the strength of SSL/TLS configurations and ensure compliance with security best practices.*
20. ***Nikto Results***



1. ***Key Vulnerabilities:***
2. *Outdated Apache:*
   * *CVE-2019-0211: This version is susceptible to privilege escalation and remote code execution, which could allow attackers to gain control over the server.*
3. *Exposed Directories (/admin, /backup):*
   * *Public access to administrative directories and backup files increases the risk of unauthorized access. Attackers could potentially download sensitive files or gain administrative privileges.*
4. *Insecure HTTP Headers:*
   * *Missing headers such as HTTP Strict Transport Security (HSTS), X-Frame-Options, and X-Content-Type-Options expose the server to attacks like clickjacking, MIME-type sniffing, and downgrade attacks.*
5. *Directory Indexing:*
   * *When directory indexing is enabled, it lists the contents of directories, potentially exposing configuration files or other sensitive data to attackers.*
6. *Sensitive Files Accessible:*
   * *Files like config.php and .htaccess contain sensitive configuration data (e.g., database credentials) and were accessible, which could lead to data leakage or further attacks.*
7. ***Potential Impact:***
8. *Outdated Apache: The use of an outdated Apache version could lead to unauthorized access, data leakage, or even full server compromise due to known vulnerabilities.*
9. *Exposed Directories: Attackers could gain access to sensitive files or back-end administrative panels, leading to the compromise of confidential information.*
10. *Insecure Headers: Without security headers, the server is at risk of various attacks, such as clickjacking, session hijacking, and protocol downgrade attacks.*
11. *Directory Indexing: Revealing directory contents may expose information useful for attackers, such as file structures or sensitive files.*
12. *Sensitive File Access: Unauthorized access to configuration files could lead to exposure of sensitive information like database credentials, enabling attackers to exploit the system further.*
13. ***Recommended Mitigations:***
14. *Update Apache: Upgrade Apache to a more recent version to mitigate vulnerabilities like privilege escalation and remote code execution.*
15. *Restrict Access to Sensitive Directories:*
    * *Implement proper access controls to ensure that sensitive directories (e.g., /admin, /backup) are not publicly accessible.*
    * *Use .htaccess files to block direct access to directories or apply authentication mechanisms (e.g., password protection).*
16. *Implement Security Headers:*
    * *HSTS: Enforce HSTS to ensure all communications are sent over HTTPS.*
    * *X-Frame-Options: Prevent clickjacking by adding the X-Frame-Options header.*
    * *X-Content-Type-Options: Use this header to prevent MIME-type sniffing by browsers.*
17. *Disable Directory Indexing:*
    * *Modify the Apache configuration to disable directory listing by adding Options -Indexes in the .htaccess file or server configuration.*
18. *Secure Sensitive Files:*
    * *Restrict access to sensitive files like config.php and .htaccess. Ensure these files are not accessible from the public internet by properly configuring access controls in the web server.*
    * *Move sensitive files to directories that are outside the web root, where they cannot be accessed directly by HTTP requests.*

***3.* OpenVAS Results**

*A screenshot of a computer

Description automatically generated*

***Key Vulnerabilities:***

1. *Apache Outdated Version (CVE-2019-0211):*
   * *Running an outdated version of Apache can allow attackers to escalate privileges and execute arbitrary code on the server.*
2. *SSL/TLS Weak Ciphers:*
   * *The server supports outdated and weak encryption ciphers, which are susceptible to attacks such as SSL stripping or cipher downgrade attacks.*
3. *PHP Outdated Version (CVE-2019-11043):*
   * *An outdated PHP version introduces the risk of remote code execution, which could allow an attacker to take control of the server by exploiting this vulnerability.*
4. *Missing Security Patches:*
   * *The operating system has multiple missing security patches, leaving the system exposed to various exploits, especially for known vulnerabilities in unpatched software.*
5. *Insecure File Upload:*
   * *File upload functionality lacks validation, allowing attackers to upload malicious scripts or executables that could lead to server compromise.*
6. *Cross-Site Scripting (XSS):*
   * *Several web pages are vulnerable to reflected XSS, enabling attackers to inject malicious scripts, which could steal sensitive information or alter the functionality of the site.*

***Potential Impact:***

* *Apache Outdated Version: This vulnerability can lead to full server compromise by enabling remote code execution or privilege escalation.*
* *Weak SSL/TLS Ciphers: Supporting weak ciphers exposes users to man-in-the-middle attacks and could allow attackers to decrypt communications between clients and the server.*
* *Outdated PHP Version: This opens the door for remote attackers to execute malicious code, potentially leading to a complete takeover of the system.*
* *Missing Security Patches: Failure to apply critical patches increases the attack surface, making the server more vulnerable to a wide array of known exploits.*
* *Insecure File Upload: Unrestricted or poorly validated file uploads allow attackers to upload and execute malicious scripts, compromising the entire server.*
* *Cross-Site Scripting (XSS): XSS can be used by attackers to hijack user sessions, redirect users to malicious sites, or execute harmful code within the browser.*

***Recommended Mitigations:***

1. *Update Apache: Immediately update Apache to the latest version to mitigate the risk of remote code execution (CVE-2019-0211) and other vulnerabilities. Regularly apply security patches to the web server.*
2. *Harden SSL/TLS Configuration:*
   * *Disable weak ciphers (e.g., RC4, MD5) and ensure that the server supports only strong encryption algorithms like AES256.*
   * *Enforce the use of TLS 1.2 or higher while disabling deprecated versions such as TLS 1.0 and 1.1.*
3. *Update PHP: Upgrade PHP to the latest stable release to address critical vulnerabilities like CVE-2019-11043, preventing potential remote code execution exploits.*
4. *Apply Missing Security Patches:*
   * *Regularly update the operating system and all installed software with the latest security patches. Set up an automated patch management system to ensure timely updates.*
5. *Secure File Uploads:*
   * *Implement proper file upload validation, including checks for file type, size, and content. Use secure upload directories and disable direct execution of uploaded files.*
6. *Mitigate Cross-Site Scripting (XSS):*
   * *Implement proper input validation and output encoding on all web forms and URLs to prevent malicious scripts from being executed. Use Content Security Policy (CSP) headers to block XSS attacks.*

***Conclusion***

*The security assessment identified critical vulnerabilities, including outdated software versions, weak SSL/TLS configurations, and insecure file uploads. These issues pose risks such as remote code execution, unauthorized access, and data leakage. Immediate actions are needed to update software, harden encryption settings, restrict access to sensitive directories, and address missing patches. Implementing these recommendations will significantly enhance the security posture and reduce the risk of compromise. Regular monitoring and patch management should be part of an ongoing security strategy.*