A project report on

# WEB APPLICATION & NETWORK SECURITY AUDIT

Submitted in partial fulfilment of the requirements for the award of the Degree of

**B.Sc. in Cyber Security**

By

**Mrittika Debnath (Roll No. 28984422010)**

Under the supervision of:

**Mr. Sirshendu Hore**

**Assistant Professor (Department of Computer Science and Technology)**



**at**

**Institute of Leadership, Entrepreneurship and Development**

**(**113 J Matheswartala Road, Near Axis Bank, Off Park Regency Hotel, Topsia, Kolkata, West Bengal 700046)

Affiliated to

Maulana Abul Kalam Azad University of Technology

(Erstwhile WBUT)

**June – 2025**

## BONAFIDE CERTIFICATE

This is to certify that the project titled **WEB APPLICATION & NETWORK SECURITY AUDIT** is a Bonafide record of the work done by

**Mrittika Debnath (Roll No. 28984422010)**

in partial fulfillment of the requirements for the award of the degree of **Bachelor in**

**Science in Cyber Security**, Department of **Computer Science and Technology**, iLEAD, Kolkata, under **MAKAUT** during the academic year **2024-25**.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Mr. Sirshendu Hore Mr. Sirshendu Hore**

**Project Guide Head of the Department**

## ABSTRACT

This project presents a comprehensive approach to securing digital infrastructure through Web Application Penetration Testing (WAPT) and Network Auditing. In an era where cyber threats are increasingly sophisticated, organizations must proactively identify and remediate vulnerabilities within their systems. The primary objective of this project was to assess the security posture of selected web applications and internal network assets using industry-standard tools and methodologies.

The WAPT phase involved simulating real-world attack scenarios to uncover vulnerabilities such as SQL injection, cross-site scripting (XSS), broken authentication, and security misconfigurations. Manual testing was complemented with automated scanning tools like Burp Suite and OWASP ZAP, following the OWASP Top 10 framework to ensure comprehensive coverage.

Simultaneously, the network auditing phase utilized tools like Nessus, Nmap, and Wireshark to identify misconfigured services, open ports, outdated software, and other potential security weaknesses within the infrastructure. Each finding was documented with a risk rating, potential impact, and actionable remediation steps.

The results of this project highlight the importance of regular vulnerability assessments and reinforce the value of adopting a layered security approach. By addressing the identified issues, organizations can significantly reduce their attack surface and enhance their overall cyber resilience.

## ACKNOWLEDGEMENTS

I, **Mrittika Debnath** would like to express my deepest gratitude to the following people for guiding me through this course and without whom this project and the results achieved from it would not have reached completion.

**Mr. Sirshendu Hore**, Assistant Professor, Department of Computer Science and Technology (Data Science & Cyber Security), who helped and guided me in the course of this project. Without his guidance, I would not have been able to successfully complete this project. His patience and genial attitude are and always will be a source of inspiration to me.

I am also thankful to the other faculty and staff members of the Department of Computer Science and Technology, my parents and my friends for their constant support and help.

## Table of Contents

1. **Abstract** ............................................................................................................................ 3
2. **Acknowledgement** ............................................................................................................ 4
3. **List of Tables** **and Figures** ................................................................................................ 6
4. **Introduction** ..................................................................................................................7 - 8
   * *Network Security* o *Web Application Security*
5. **Methodology & Approach** ................................................................................................ 9
   * *Information Gathering* o *Vulnerability Detection* o *Information Analysis and Planning* o *Attack and Penetration* o *Privilege Escalation* o *Result Analysis* o *Reporting*
6. **Network VAPT Approach** ............................................................................................... 10
7. **WAPT Approach** ....................................................................................................... 11 - 12
   * *Web App Audit Test Standards (OWASP Top 10)* …………………………... 12 - 15
8. **Risk Level & Description** ……………………………………………………………... 16
9. **Tools Used During Assessment** ............................................................................... 16 - 18
10. **Detailed Observations for WAPT** ........................................................................... 19 - 31
11. **Detailed Observations for Network VAPT** .................................................................... 32
    * *Host Scanning using Advance Ip Scanner* …………………………………. 32 - 33 o *Port Scanning Using Nmap* ………………………………………………... 34 - 41 o *Vulnerability Assessment using Nessus* ……………………………………. 41 - 91
12. **Conclusion** ...................................................................................................................... 92
13. **Limitations** ……………………………………………………………………………... 93
14. **Future Works** …………………………………………………………………………. 93
15. **References and Bibliography** ........................................................................................ 94

### LIST TABLES

**Title** **Page No.**

**Sl. No.**

|  |  |  |
| --- | --- | --- |
| **1** | **OWASP Top 10 (2021) Vulnerabilities** | 7 |
| **2** | **Risk Level & Description for WAPT & Network VAPT** | 10 |
| **3** | **Observation 1 for WAPT Result** | 13 |
| **4** | **Observation 2 for WAPT Result** | 15 |
| **5** | **Observation 3 for WAPT Result** | 17 |
| **6** | **Observation 4 for WAPT Result** | 17 |
| **7** | **Observation 5 for WAPT Result** | 19 |
| **8** | **Observation 6 for WAPT Result** | 20 |
| **9** | **Observation 7 for WAPT Result** | 21 |
| **10** | **Observation 8 for WAPT Result** | 22 |
| **11** | **Observation 9 for WAPT Result** | 23 |
| **12** | **Observation 10 for WAPT Result** | 24 |
| **13** | **Hosts Scanning by Advance IP Scanner** | 26 - 27 |
| **14** | **Nmap Port Scanning Results for Various IP Addresses** | 28 - 35 |

**LIST OFFIGURES**

**Sl. No. Name Page No.**

|  |  |  |
| --- | --- | --- |
| **1** | **Network VAPT Testing Work Flow (Fig: 1,2,3)** | 4 |
| **2** | **WAPT Approach at a glance (Fig: 4,5)** | 5 |
| **3** | **WAPT Approach at a glance (Fig: 6,7)** | 6 |
| **4** | **Tools Description (Fig: 8)** | 10 |
| **5** | **Nmap (Fig: 9)** | 11 |

## Introduction

In today's digitally interconnected world, the security of networks and web applications is paramount. As organizations increasingly rely on digital infrastructures to conduct business, the integrity, confidentiality, and availability of their systems and data have become critical concerns. Cyber threats have evolved in sophistication and frequency, targeting vulnerabilities across both network infrastructures and web-based platforms. Consequently, comprehensive security audits have emerged as essential tools to assess, identify, and mitigate potential risks, ensuring robust defense mechanisms are in place.

**Network Security:**

Network security encompasses the policies, practices, and technologies employed to protect the integrity and usability of network and data. A network security audit systematically evaluates an organization's network infrastructure to identify vulnerabilities, ensure compliance with security standards, and recommend enhancements. Key components of such an audit include:

* Asset Inventory: Cataloging all devices, applications, and data assets within the network.
* Risk Assessment: Identifying potential threats and evaluating their impact on the organization.
* Vulnerability Scanning: Utilizing automated tools to detect known vulnerabilities in systems and applications.
* Penetration Testing: Simulating cyber-attacks to assess the effectiveness of existing security measures.
* Policy and Compliance Review: Ensuring that security policies align with industry standards and regulatory requirements.

Regular network security audits are vital for proactive threat detection, maintaining compliance, and safeguarding organizational assets.

**Web Application Security:**

Web applications serve as gateways to critical business functions and data. Their ubiquitous nature makes them prime targets for cyber-attacks. A web application security audit focuses on identifying and rectifying vulnerabilities within web applications to prevent exploitation. The audit process typically involves:

* Static Application Security Testing (SAST): Analyzing source code to detect security flaws without executing the application.
* Dynamic Application Security Testing (DAST): Evaluating the application in its running state to identify vulnerabilities that manifest during execution.
* Manual Code Review: Conducting thorough inspections of code to uncover complex vulnerabilities that automated tools might miss.
* Risk Analysis: Prioritizing identified vulnerabilities based on their potential impact and likelihood of exploitation.
* Remediation and Verification: Implementing fixes and re-evaluating to ensure vulnerabilities have been effectively addressed.

Implementing security measures such as Web Application Firewalls (WAFs) further enhances protection by filtering and monitoring HTTP traffic to and from web applications, thereby mitigating threats like SQL injection and cross-site scripting.

This project aims to conduct comprehensive security audits on both network infrastructures and web applications, employing industry-standard methodologies and tools. The objective is to identify vulnerabilities, assess risks, and recommend actionable solutions to fortify security postures, thereby contributing to the broader goal of safeguarding digital assets in an increasingly threat-laden cyber landscape.

### 1. Methodology & Approach

In accordance with the audit scope, same has been carried out in three phases: the planning phase, the conduct phase, and the reporting phase.

The purpose of the planning phase was to develop a Risk-Based Audit Program that provides a basis for the orderly, efficient, and cost-effective conduct of the audit as well as a criteria base for assessment. The methodology under which the entire work was performed are as follows:

* **Information Gathering**

Security Audit team obtains in-depth information about the IT environment of the auditee’s organization. The areas covered include the networks, IP Addresses, Operating System Versions, and more. The information is to be gathered when either of the three types of scopes such as Black Box Testing, White Box Testing, or Grey Box Testing, is being used.

* **Vulnerability Detection**

Vulnerability detection is perhaps the most critical phase of the entire process. It encompasses the use of reliable vulnerability scanners to scan the entire IT environment of the organization with the aim to identify the vulnerabilities that exist within the system, applications, infrastructure, or elsewhere within the business.

* **Information Analysis and Planning**

As a part of the VAPT methodology, Audit team analyze the vulnerabilities so identified and come up with a result-oriented plan a plan for penetrating into the systems and network. The objective is to ensure that we have a well-established plan of action for resolving the problems that are actually there within the client’s organization.

* **Attack and Penetration**

In the next phase, Team performs the exploitation of the vulnerabilities that are detected in the VA scanning. This is done by performing an attack on the system to confirm the existence of the vulnerability. The plan of action formulated in the previous step is executed to attack and penetrate the target systems.

* **Privilege Escalation**

After we penetrate successfully into the system, we apply the privilege escalation technique for the identification of vulnerabilities and escalation of access. This is done for gaining higher privileges, such as administrative privileges or registry/root access to that particular system or network in the IT environment.

* **Result Analysis**

Finally, after the penetration testing is completed, our team performs the root cause analysis. The result so obtained is analyzed to deliver relevant recommendations for making the organization’s IT environment secure by plugging the vulnerabilities and holes that are detected in one or more of the systems therein.

* **Reporting**

Once the penetration test is done and the root cause of the vulnerabilities is analyzed, we give a comprehensive report to the client. It comprises detailed information regarding the vulnerabilities detected in the security assessment, risk rating for each of them, supporting detailed exhibits, and detailed technical remediation recommendations.

#### 1.1 Network VAPT Approach

During the conduct phase, the audit team performed Network VAPT using tools like Nmap, Nessus, Advanced IP Scanner on preliminary findings.

Fig1: **Network VAPT Testing Work Flow**

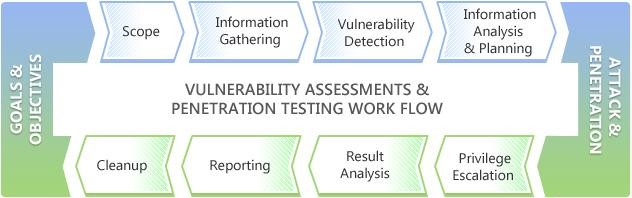
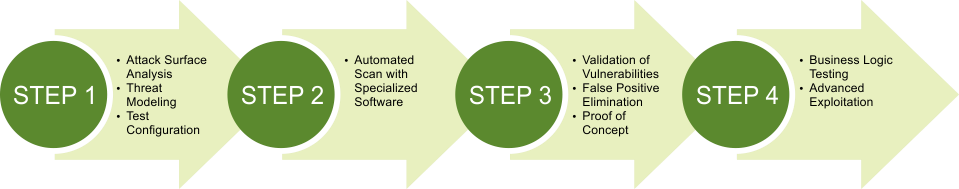
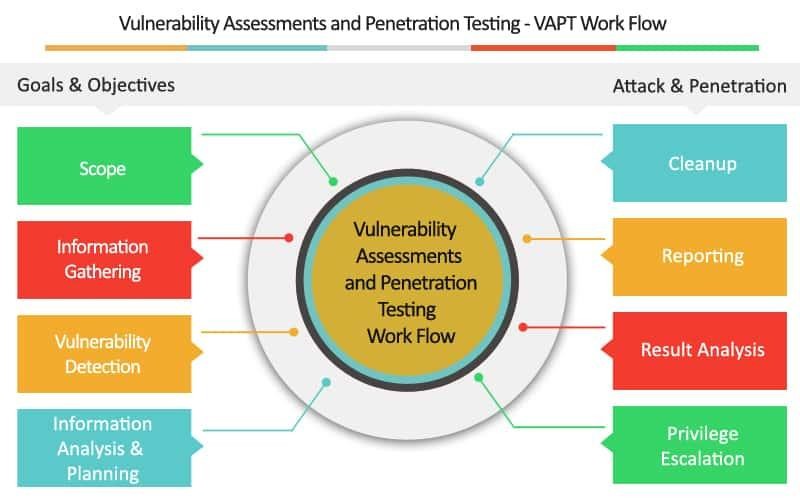


Fig 2: **Network VAPT Testing Work Flow**



Once the penetration test is done and the root cause of the vulnerabilities is analyzed, we give a comprehensive report to the client. It comprises detailed information regarding the vulnerabilities detected in the security assessment, risk rating for each of them, supporting detailed exhibits, and detailed technical remediation recommendations.

Fig 3: **Network VAPT Testing Work Flow**



#### 1.2 WAPT Approach

**Our present Web Application Audit Plan involves**

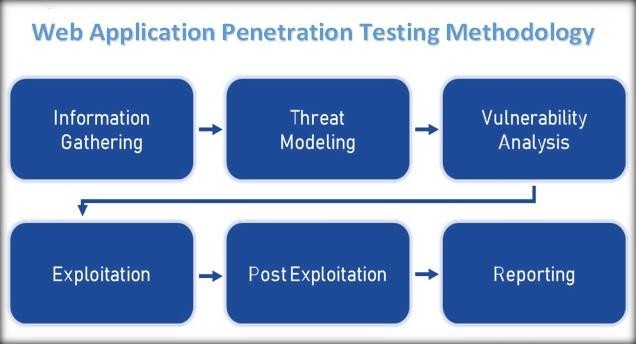


Fig 4: **WAPT Approach at a glance**

**Focusing on key nine areas:**



Fig 5: **WAPT Approach at a glance**



Fig 6: **WAPT Approach at a glance**

**Approach at a glance:**

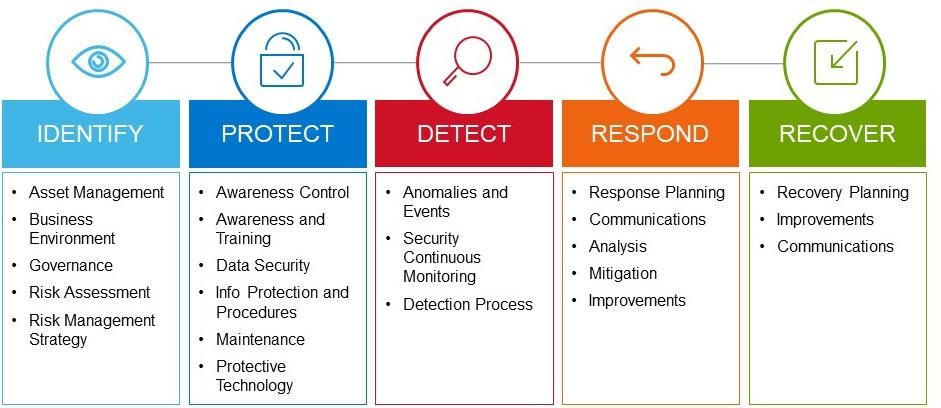


Fig 7: **WAPT Approach at a glance**

##### 1.2.1 Web App audit Test Standard Followed

* **OWASP Top 10 (2021)**

The Open Web Application Security Project (OWASP) is a non-profit community of software developers, engineers, and freelancers that provides resources and tools for web application security. Every few years, OWASP releases a report on the 10 most critical web application security risks.

We followed OWASP Top 10 vulnerabilities for testing for vulnerabilities. For more details, refer to the following link:

<https://owasp.org/www-project-top-ten/>

* **OWASP Top 10 (2021) Vulnerability Categories and Impact**

**Tab 1:** **OWASP Top 10 (2021) Vulnerabilities**

|  |  |  |
| --- | --- | --- |
| SL. No. | Vulnerability Categories | Impact |
| **A01** | **Broken Access Control**  Broken Access Control happens when access permissions are misconfigured thereby allowing attackers to access, modify or delete data, files and accounts that they should not have access to in the first place. | Attackers can exploit authorization flaws to accomplish the following:   * Access unauthorized functionality and/or data View sensitive files * Change access rights * Edit files and records. |
| **A02** | **Cryptographic Failures**  Cryptographic failures occur when sensitive data is insufficiently protected and therefore leaked or exposed to unauthorized audiences. Such failures are most common if data is transmitted or stored in clear text or using known-to-be-weak cryptographic algorithms such as MD5 or SHA-1. | An attacker monitors network traffic (e.g., at an insecure wireless network), downgrades connections from HTTPS to HTTP, intercepts requests, and steals the user’s session cookie. The attacker then replays this cookie and hijacks the user’s (authenticated) session, accessing or modifying the user’s private data. Instead of the above, they could alter all transported data. |
| **A03** | **Injection**  An attacker can execute unintended commands or gain access to sensitive data by injecting malicious data as part of a command or query. This usually happens when a website fails to filter, validate or sanitize users’ inputs or implement parameterization. | Injection vulnerabilities can occur when a query or command is used to insert untrusted data into the interpreter via SQL, OS, NoSQL, or LDAP injection. The hostile data injected through this attack vector tricks the interpreter to make the application do something it was not designed for, such as generating unintended commands or accessing data without proper authentication. |
| **A04** | **Insecure Design**  Insecure design is a new entry on the OWASP Top 10 in 2021. It is different from insecure implementation in that it has more to do with risks related to design and architectural flaws. A secure implementation might have an insecure design which still renders a web application vulnerable to attacks and exploits. | Applications that were not developed with security in mind from the very beginning are more likely to put user data and security at risk, and require updates, patches, and fixes to prevent these risks. Applications without secure design are low hanging fruit for attackers and can cost incalculable sums of damage in terms of leaked data, tarnished reputations, and paid workinghours of cleanup and future prevention. |
| **A05** | **Security Misconfiguration**  This category covers a brand range of potential vulnerabilities including insecure default | There are many types of misconfigurations that expose the |

|  |  |  |
| --- | --- | --- |
|  | configurations, incomplete configurations, and misconfigured HTTP headers, using insecure default usernames and passwords, etc. | company to cybersecurity risk, including:   * Accepting default settings that are insecure * Overly accessible cloud storage resources * Incomplete configurations * Misconfigured HTTP headers * Verbose error messages that contain sensitive information |
| **A06** | **Vulnerable and Outdated Components**  This refers to known issues where vulnerabilities exist because developers either do not know the versions of components used including those of nested dependencies, or are not aware that the software used is already unsupported or out of date. | Any component with a known vulnerability becomes a weak link that can impact the security of the entire application. |
| **A07** | **Identification and Authentication Failures**  This category covers weaknesses in authentication and session management in web applications. The resulting vulnerabilities allow attackers to gain unauthorized access to accounts and/or data. | Websites with broken authentication vulnerabilities are very common on the web. Broken authentication usually refers to logic issues that occur on the application authentication’s mechanism, like bad session management prone to username enumeration – when a malicious actor uses brute-force techniques to either guess or confirm valid users in a system. |
| **A08** | **Software and Data Integrity Failures**  It is concerned with the failure to verify the integrity of software updates and patches prior to implementation on live applications and servers. | These failures can be summarized as follows:   * Usage of code that does not verify integrity of source * Usage of third-party plugins where you do not control the source * Plugins and extensions from untrusted sources * The introduction of or potential for compromise or unauthorized access * Auto-updates assume trust of the source. |
| **A09** | **Security Logging and Monitoring Failures** Logging and monitoring are essential components in ensuring that any suspicious activity can be detected close to real-time, or diagnosed after the fact. Failure to keep sufficient records in these areas could subsequently lead to slower incident responses, thereby accentuating the potential damages of breaches. | This window gives cyber thieves plenty of time to tamper with servers, corrupt databases, steal confidential information, and plant malicious code. |
| **A10** | **Server-Side Request Forgery** | When a web application fetches a remote resource without |
|  | Server-Side Request Forgery (SSRF) occurs when a web application proceeds to fetch data without first validating user-supplied URL. In a bid to provide end-users with convenience, fetching data using a URL has become more common. The vulnerability allows an attacker to compel the web application to send a crafted request to unexpected destinations even when adequately protected by firewalls, VPNs and Network Access Control List (ACL). | validating the user-supplied URL, an SSRF fault occurs. Even if the program is secured by a firewall, VPN, or another sort of network access control list, an attacker can force it to send a forged request to an unexpected location. |

#### 1.3 Risk Level & Description

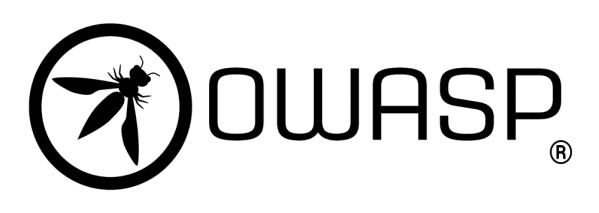
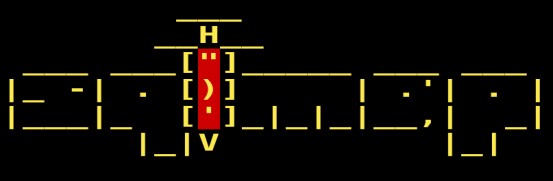
**Tab 2: Risk level & description for WAPT & Network VAPT**

|  |  |
| --- | --- |
| **Vulnerability Level** | **Description** |
| **Critical** | Exploitation of the vulnerability may result in complete compromise of the Database server or Application server. It can have a major impact on business. **(CVSS Score- 9.0-10)** |
| **High** | Exploitation of the vulnerability may result in complete compromise of the Application / disclosure of sensitive information. Vulnerability is easily exploitable. **(CVSS Score- 7.0-8.9)** |
| **Medium** | Exploitation of the vulnerability may result in some control on the Application / disclosure of semi-sensitive information. Exploitation of this vulnerability is possible but difficult. **(CVSS Score- 4.0-6.9)** |
| **Low** | Exploitation of the vulnerability may result in some control on the Application / disclosure of semi-sensitive information. Exploitation of this vulnerability is possible but difficult. **(CVSS Score- 4.0-6.9)** |
| **Informational** | **The informational risk level** indicates that some functionality or component is missing best practices implementation in the application. Such vulnerability may not have a risk associated with it currently, but it may become vulnerability in future due to change in application or due to exploiting techniques evolution or policy/legal requirements. The vulnerability mitigation depends upon owner decision; however, it is recommended to be mitigated if it not in line with the policy or law. |

#### 1.4 Tools Used During assessment

Fig 8**:Tools Description**





Tab 3**: Tools & their description**

|  |  |  |  |
| --- | --- | --- | --- |
| S.No. | Name of Tool/Software used | Version of the tool/Software used | Type |
| 1 | Nmap | 7.95 | Open Source |
| 2 | Burpsuite | 1.6 | Open Source |
| 3 | Kali Linux | 2024.4 | Open Source |
| 4 | Nessus | 10.6.3 | Open Source |
| 5 | OWASP ZAP | 2.14.0 | Open Source |
| 6 | Advance Ip Scanner | 2.5.4594.1 | Open Source |

**Tools Description:**

**Nmap**

Description: A powerful network scanner for discovery and security auditing Key Features:

* Identifies live hosts, open ports, and services.
* Performs detailed scans.



Fig 9: **Nmap**

**Advance Ip Scanner**

Description: Advance Ip Scanner helps to find alive host in a network range.

* Scans hundreds of IP addresses in seconds.
* Detects all devices on your network (wired or wireless).

**Burp Suite :**

Description: Burp Suite is a web vulnerability scanner used in ethical hacking and penetration testing.

* It intercepts and modifies HTTP/S traffic between the browser and the web server.
* Includes tools like Scanner, Intruder, and Repeater to find and exploit web app flaws like XSS and SQLi.

**Kali Linux:**

Description: Kali Linux is a Debian-based Linux distribution designed primarily for digital forensics and penetration testing.

* A Linux OS designed for penetration testing and ethical hacking.
* Comes pre-installed with hundreds of security tools.
* Widely used by security professionals and researchers.

**Nessus:**

Description: Nessus is a powerful and widely used vulnerability assessment tool developed by Tenable, Inc.

* A vulnerability assessment tool for detecting system flaws.
* Scans for misconfigurations, missing patches, and known exploits.
* Generates detailed security reports for remediation.

**OWASP ZAP:**

Description:

* A web application scanner developed by OWASP.
* Identifies issues like XSS, SQLi, and broken authentication.
* Offers automated and manual testing tools, suitable for beginners.

**RESULT AND DISCUSSION**

**Detailed Observation For WAPT Result**

### Observation: 1

Tab 4**: Observation 1 for WAPT result**

|  |  |  |
| --- | --- | --- |
| **Location of**  **Vulnerability/Error** | <http://127.0.0.1:8080/accounts/update-password/> | |
| **Vulnerable location, parameter, path, link** | /accounts/update-password/ | |
| **Name of**  **Vulnerability** | Full account Takeover via Forgot Password | |
| **Proof of Concept** | 1. Visited the URL: <http://127.0.0.1:8080/accounts/update-password/> 2. Used a generic “admin@gmail.com” email and clicked “Reset Password”      1. The web Application Responds with a new password without any user interaction. | |
|  |  |  |
| **Workaround /**  **Solutions** | 1.  2. | **Require user interaction**: Implement multi-factor authentication (MFA) for password resets, such as sending a one-time code via SMS or email that the user must enter to confirm the request.  **Secure email verification**: Ensure that reset links sent via email expire quickly (e.g., within an hour) and are one-time-use only. Verify the email sender’s identity and ensure links are HTTPS-secured. |
|  | 3. | **Enhanced identity verification**: Implement additional identity checks, such as sending a security question or requiring the user to verify their identity through known information (e.g., previous login details). |
|  | 4. | **Monitor password reset activity**: Implement monitoring for unusual patterns in password reset requests and flag suspicious activities. |

### Observation: 2

Tab 5**: Observation 2 for WAPT result**

|  |  |
| --- | --- |
| **Location of**  **Vulnerability/Error** | <http://127.0.0.1:8080/accounts/login/> |
| **Vulnerable location, parameter, path, link** | /accounts/login/ |
| **Name of**  **Vulnerability** | Improper CAPTCHA implementation in Login Page |
| **Proof of Concept** | 1. Visited the URL: <http://127.0.0.1:8080/accounts/update-password/> 2. Used a generic “admin@gmail.com” email and clicked “Reset Password”      1. The web Application Responds with a new password without any user interaction. |
|  |  |
| **Workaround /**  **Solutions** | 1. **Use a more secure CAPTCHA mechanism**: Implement a CAPTCHA   that requires the user to solve an image-based challenge, such as a distorted text or an image selection task, which is harder for automated scripts to interpret.   1. **Incorporate a CAPTCHA token**: Send the CAPTCHA challenge as a base64-encoded image or a challenge string, and verify the response on the server side by checking the integrity and randomness of the challenge token. 2. **Limit retries**: Introduce rate limiting to prevent brute-force attacks on the CAPTCHA. 3. **Implement additional security checks**: Combine CAPTCHA with other security measures like multi-factor authentication to further secure the login process. |

### Observation: 3

Tab 6**: Observation 3 for WAPT result**

|  |  |
| --- | --- |
| **Location of**  **Vulnerability/Error** | <http://127.0.0.1:8080/accounts/login/> |
| **Vulnerable location, parameter, path, link** | /accounts/login |
| **Name of**  **Vulnerability** | Clear text submission of password |
| **Proof of Concept** | 1. Visited the URL: [http://127.0.0.1:8080/accounts/login/.](http://127.0.0.1:8080/accounts/login/) 2. Entered the Credentials and then clicked the “Sign In” button. 3. Then Intercepted the request in Burp Suite, to find the Password field containing the password value in plain text. |
| **Workaround /**  **Solutions** | 1. **Implement HTTPS (SSL/TLS**): Encrypt communication between the client and server by implementing HTTPS (SSL/TLS) for all web pages where passwords are submitted. This encryption ensures that passwords and other sensitive data are transmitted securely over the network, protecting them from interception by attackers. 2. **Use Secure Channels**: Ensure that passwords are only transmitted over secure channels and not included in URLs, query strings, or other insecure transmission methods. Avoid sending passwords via unencrypted email or instant messaging platforms. 3. **Implement Secure Authentication Protocols**: Utilize secure authentication protocols such as OAuth or OpenID Connect, which delegate the authentication process to trusted third-party identity providers. This reduces the risk of exposing passwords during the authentication process. 4. **Hash and Salt Passwords**: Store passwords securely on the server by hashing and salting them before storage. Hashing ensures that even if the password database is compromised, passwords cannot be easily decrypted by attackers. |

### Observation: 4

Tab 7**: Observation 4 for WAPT result**

|  |  |
| --- | --- |
| **Location of**  **Vulnerability/Error** | <http://127.0.0.1:8080/blog-posts/3/edit/> |
| **Vulnerable location, parameter, path, link** | /blog-posts/3/edit/ |
| **Name of**  **Vulnerability** | Stack Trace Disclosure (Python) |
| **Proof of Concept** | 1. Visited the URL: <http://127.0.0.1:8080/blog-posts/3/edit/> 2. Uploaded a normal Image and clicked “Update”      1. The website responds with an error disclosing the Stack Trace. |
| **Workaround /**  **Solutions** | 1. **Configure Python Application**: To mitigate this risk, we have to configure Python applications to handle exceptions securely. Set up custom error handling mechanisms to display user-friendly error messages without exposing detailed stack traces. In production environments, set the DEBUG mode to False to prevent traceback information from being sent to clients. Regularly review exception handling practices and error reporting configurations to ensure they align with security best practices. |

### Observation: 5

Tab 8**: Observation 5 for WAPT result**

|  |  |
| --- | --- |
| **Location of**  **Vulnerability/Error** | <http://127.0.0.1:8080/> |
| **Vulnerable location, parameter, path, link** | / |
| **Name of**  **Vulnerability** | Cross Domain Script Include |
| **Proof of Concept** | 1. Visited URL: [http://127.0.0.1:8080/.](http://127.0.0.1:8080/) 2. Navigated console on accessing developer tools in the web browser. 3. Added the below payload from an attacker-owned domain and see that it is executed on the website. |
| **Workaround /**  **Solutions** | 1. **Avoid using wildcards**: By avoiding the uses of wildcards in the crossdomain policy file any domain matching the wildcard expression will be implicitly trusted, and can perform two-way interaction with the target server. |

**6**

Tab 9**: Observation 6 for WAPT result**

|  |  |
| --- | --- |
| **Location of**  **Vulnerability/Error** | <http://127.0.0.1:8080/static/img/pexels2.jpeg> |
| **Vulnerable location, parameter, path, link** | /static/img/pexels2.jpeg |
| **Name of**  **Vulnerability** | EXIF data exposure in default images |
| **Proof of Concept** | 1. Fetched the Image from the URL: <http://127.0.0.1:8080/static/img/pexels2.jpeg>using curl and piped it through exiftool to find the EXIF data. |
| **Workaround /**  **Solutions** | 1. **Remove EXIF metadata**: Strip out the EXIF data from images before uploading them to prevent this information from being publicly exposed. Use tools like exiftool or online services for this purpose. 2. **Implement automated EXIF stripping**: Ensure that content management systems (CMS) or image uploading features automatically remove EXIF metadata from images before they are stored or displayed publicly. 3. **Configure server settings**: Adjust web server configurations to prevent the disclosure of EXIF data when serving images. |

**7**

Tab 10 **Oservation 7 for WAPT result**

|  |  |
| --- | --- |
| **Location of**  **Vulnerability/Error** | <http://127.0.0.1:8080/blog-posts/3/edit/> |
| **Vulnerable location, parameter, path, link** | /blog-posts/3/edit/ |
| **Name of**  **Vulnerability** | Information Disclosure (Source Code) |
| **Proof of Concept** | 1. Visited the URL: <http://127.0.0.1:8080/blog-posts/3/edit/> 2. Uploaded a normal Image and clicked “Update”        1. The website responds with an error disclosing the source code. |
| **Workaround /**  **Solutions** | 1. **Review the cause of the code disclosure**: Server-side source code is normally disclosed to clients as a result of typographical errors in scripts or because of misconfiguration, such as failing to grant executable permissions to a script or directory. Review the cause of the code disclosure and prevent it from happening. |

**8**

Tab 11 **Observation 8 for WAPT result**

|  |  |
| --- | --- |
| **Location of**  **Vulnerability/Error** | <http://127.0.0.1:8080/post/1/like/> |
| **Vulnerable location, parameter, path, link** | /post/1/like/ |
| **Name of**  **Vulnerability** | Open Redirection via Referer Header |
| **Proof of Concept** | 1. Visited the URL: [http://127.0.0.1:8080/post/1/.](http://127.0.0.1:8080/post/1/) 2. Clicked the button “Like” and captured the request in Burpsuite. 3. Then changed the “Referer” header value to “https://evil.com” and the web application redirects to “https://evil.com”. |
| **Workaround /**  **Solutions** | 1. **Removing the redirection function**: By removing the redirection function from the application, and replace links to it with direct links to the relevant target URLs. 2. **Maintain a server-side list of all URLs**: By maintaining a server-side list of all URL that are permitted for redirection. Instead of passing the target URL as a parameter to the redirector, pass an index into this list. |

**9**

Tab 12 **Observation 9 for WAPT result**

|  |  |
| --- | --- |
| **Location of**  **Vulnerability/**  **Error** | [http://127.0.0.1:8080/'](http://127.0.0.1:8080/) |
| **Vulnerable location, parameter, path, link** | /’ |
| **Name of**  **Vulnerability** | Improper error handling |
| **Proof of Concept** | 1. Visited the URL: [http://127.0.0.1:8080/'.](http://127.0.0.1:8080/) 2. The Web Application responds with detailed error messages. |
| **Workaround /**  **Solutions** | 1. **Implement Custom Error Messages**: Replace generic error messages with custom ones that do not reveal sensitive information. Provide informative yet non-specific feedback to users to avoid disclosing details that could aid attackers. 2. **Log Errors Securely**: Ensure that detailed error messages, including stack traces and system paths, are logged securely on the server-side. Use logging mechanisms that store error logs in protected directories inaccessible to unauthorized users. 3. **Utilize Error Codes**: Instead of revealing specific system errors, utilize error codes to communicate with users. Map these codes to corresponding error messages on the server-side, maintaining consistency while obscuring underlying system details. 4. **Centralize Error Handling**: Centralize error handling logic to ensure uniformity across the application. Implement a dedicated error handling mechanism that intercepts errors and provides standardized responses, reducing the risk of inconsistencies and information leakage. |

### Observation: 10

Tab 13**: Observation 10 for WAPT result**

|  |  |
| --- | --- |
| **Location of**  **Vulnerability/**  **Error** | <http://127.0.0.1:8080/accounts/login> |
| **Vulnerable location, parameter, path, link** | /accounts/login |
| **Name of**  **Vulnerability** | Concurrent Logins |
| **Proof of Concept** | 1. Visited URL: <http://127.0.0.1:8080/accounts/login>from two different browsers using the same credentials.      1. Observed that the both the browsers has two different sessions of the same account at the same time. |
| **Workaround /**  **Solutions** | 1. **Session Management**: Utilize robust session management techniques to track user sessions securely. Implement mechanisms to associate each session with a unique identifier and maintain session state securely on the server-side. 2. **Single Sign-On (SSO**): Implement single sign-on solutions that allow users to authenticate once and access multiple related applications or services without requiring separate logins. Ensure that SSO implementations include appropriate controls for managing concurrent sessions and enforcing security policies. 3. **Session Timeout**: Configure session timeout values to automatically |
|  | expire inactive sessions after a predefined period of inactivity. This helps reduce the window of opportunity for attackers to exploit concurrent logins by hijacking inactive sessions.  Session Revocation: Implement the ability to revoke or invalidate user sessions manually or automatically in response to suspicious activity or logout events. This ensures that compromised sessions are promptly terminated, preventing unauthorized access. |

#### Detailed Observation for Network VAPT Result

1. **Alive Hosts scanning by Advance IP Scanner:**

**Introduction:**

This report outlines the findings of a network scan conducted using Advanced IP Scanner on the 192.168.30.0/24 subnet in the college lab. The purpose of the scan was to identify active hosts, open ports, and gather basic device information for network auditing and educational purposes.

Tab 14**: Hosts scanning by Advance IP Scanner**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Status Name IP Address Manufacturer MAC Address Comments** | | | | | | | | | | | |
| On | | 192.168.30.1 | | 192.168.30.1 | | TP-Link  Corporation Limited | | 60:A4:B7:D7:F9:24 | |  | |
| On | | AnshSaraswat | | 192.168.30.60 | | GIGA-BYTE  TECHNOLOGY  CO., LTD. | | D8:5E:D3:DA:43:77 | |  | |
| On | | 192.168.30.61 | | 192.168.30.61 | | PCS  Systemtechnik  GmbH | | 08:00:27:05:9B:18 | |  | |
| On | | 192.168.30.72 | | 192.168.30.72 | | GIGA-BYTE  TECHNOLOGY  CO., LTD. | | B4:2E:99:23:47:2F | |  | |
| On | | Samarjeet | | 192.168.30.74 | | GIGA-BYTE  TECHNOLOGY  CO., LTD. | | D8:5E:D3:7C:14:C7 | |  | |
| On | | susmita2003 | | 192.168.30.95 | | GIGA-BYTE  TECHNOLOGY  CO., LTD. | | D8:5E:D3:DA:44:6B | |  | |
| On | | DESKTOPNNF4PE6 | | 192.168.30.98 | | Hewlett Packard | | 10:62:E5:00:EB:C9 | |  | |
| On | | WROOM | | 192.168.30.103 | | GIGA-BYTE  TECHNOLOGY  CO., LTD. | | D8:5E:D3:7C:17:D9 | |  | |
| On | | 192.168.30.119 | | 192.168.30.119 | | VMware, Inc. | | 00:0C:29:FA:9E:4A | |  | |
| On | | DESKTOP-  MCGC7VT | | 192.168.30.123 | | GIGA-BYTE  TECHNOLOGY  CO., LTD. | | D8:5E:D3:91:DC:05 | |  | |
| On | | DESKTOP5OJR5G2 | | 192.168.30.129 | | GIGA-BYTE  TECHNOLOGY  CO., LTD. | | E0:D5:5E:30:5C:EB | |  | |
| On | | 192.168.30.131 | | 192.168.30.131 | |  | | 04:7C:16:4E:12:A2 | |  | |
| On | | DESKTOPJN87P83 | | 192.168.30.133 | | GIGA-BYTE  TECHNOLOGY  CO., LTD. | | 1C:1B:0D:FB:52:D8 | |  | |
| On | | 192.168.30.137 | | 192.168.30.137 | | GIGA-BYTE  TECHNOLOGY  CO., LTD. | | D8:5E:D3:DA:44:45 | |  | |
| On | | 192.168.30.141 | | 192.168.30.141 | | COMPAL INFORMATION | | 98:28:A6:0C:BD:B3 | |  | |
|  | |  | |  | | (KUNSHAN)  CO., LTD. | |  | |  | |
| On | | 192.168.30.148 | | 192.168.30.148 | | VMware, Inc. | | 00:0C:29:4E:50:97 | |  | |
| On | | DESKTOP-  SQU6VV5 | | 192.168.30.154 | | GIGA-BYTE  TECHNOLOGY  CO., LTD. | | D8:5E:D3:7C:17:E5 | |  | |
| On | | DESKTOPRFR9D15 | | 192.168.30.155 | | GIGA-BYTE  TECHNOLOGY  CO., LTD. | | B4:2E:99:80:08:FD | |  | |
| On | | 192.168.30.169 | | 192.168.30.169 | |  | | 04:7C:16:4E:12:95 | |  | |
| On | | 192.168.30.174 | | 192.168.30.174 | | GIGA-BYTE  TECHNOLOGY  CO., LTD. | | D8:5E:D3:DA:43:22 | |  | |
| On | | 192.168.30.175 | | 192.168.30.175 | | GIGA-BYTE  TECHNOLOGY  CO., LTD. | | D8:5E:D3:DA:43:24 | |  | |
| On | | 192.168.30.178 | | 192.168.30.178 | |  | | 04:7C:16:D1:36:CE | |  | |
| On | | 192.168.30.199 | | 192.168.30.199 | | GIGA-BYTE  TECHNOLOGY  CO., LTD. | | E0:D5:5E:30:95:5B | |  | |
| On | | DESKTOPISU6VCI | | 192.168.30.202 | | GIGA-BYTE  TECHNOLOGY  CO., LTD. | | B4:2E:99:23:4E:16 | |  | |
| On | | 192.168.30.210 | | 192.168.30.210 | |  | | 08:BF:B8:DB:41:36 | |  | |
| On | | 192.168.30.211 | | 192.168.30.211 | | PEGATRON CORPORATION | | 20:25:64:0D:2D:D0 | |  | |
| On | | 192.168.30.213 | | 192.168.30.213 | | GIGA-BYTE  TECHNOLOGY  CO., LTD. | | D8:5E:D3:7C:1A:96 | |  | |
| On | | 192.168.30.243 | | 192.168.30.243 | | GIGA-BYTE  TECHNOLOGY  CO., LTD. | | E0:D5:5E:D0:F7:68 | |  | |
| On | | DEVC3D072 | | 192.168.30.250 | |  | | 00:00:00:00:00:00 | |  | |
| On | | DEVC3D072 | | 192.168.30.255 | | GIGA-BYTE  TECHNOLOGY  CO., LTD. | | D8:5E:D3:DA:44:6B | |  | |

1. **Network Port scanning by Nmap:**

**Introduction:**

To perform a network scan on the subnet **192.168.30.0/24** using Nmap in order to identify active hosts, open ports, running services, and potential vulnerabilities.

**Tool Name**: Nmap

**Version**: 7.95

**Operating System**: Kali Linux (version: 2024.4)

**Nmap scan report for 192.168.30.1**

Tab 15**: Port scanning by Nmap**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Port** | **State** | **Protocol** | **Service** | **Version** |
| **22** | open | tcp | ssh | Dropbear sshd  2011.54 (protocol  2.0) |
| **53** | open | tcp | domain | ISC BIND 9.16.23 (Redhat /linux) |
| **80** | open | tcp | http | BusyBox http  1.94.4 |
| **443** | open | tcp | ssl/http | BusyBox http  1.94.4 |
| **1900** | open | tcp | upnp | MiniUPnp 1.8 (UPnP 1.1) |

**Nmap scan for 192.168.30.60**

Tab 16**: Port scanning by Nmap**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Port** | **State** | **Protocol** | **Service** | **Version** |
| **135** | open | tcp | msrpc | Microsoft  Windows RPC |
| **139** | open | tcp | netbios-ssn | Microsoft  Windows netbiosssn |
| **445** | open | tcp | microsoft-ds |  |
| **7680** | open | tcp | pando-pub |  |
| **49668** | open | tcp | msrpc | Microsoft  Windows RPC |

**Nmap scan for 192.168.30.61**

Tab 17**: Port scanning by Nmap**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Port** | **State** | **Protocol** | **Service** | **Version** |
| **42195** | open | tcp | ssl/unknown | Microsoft  Windows RPC |

**Nmap scan port 192.168.30.72**

Tab 18**: Port scanning by Nmap**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Port** | **State** | **Protocol** | **Service** | **Version** |
| **7680** | open | tcp | pando-pub |  |

**Nmap scan for 192.168.30.74**

Tab 19**: Port scanning by Nmap**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Port** | **State** | **Protocol** | **Service** | **Version** |
| **135** | open | tcp | msrpc | Microsoft windows RPC |
| **139** | open | tcp | netbios-ssn | Microsoft windows netbios-  ssn |
| **445** | open | tcp | microsoft-ds |  |
| **3389** | open | tcp | ms-wbt-server | Microsoft Terminal Services |
| **7680** | open | tcp | pando-pub |  |
| **49669** | open | tcp | msrpc | Microsoft windows RPC |

**Nmap scan report for 192.168.30.91**

Tab 20**: Port scanning by Nmap**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Port** | **State** | **Protocol** | **Service** | **Version** |
| **3306** | open | tcp | mysql | Mysql (unauthorized) |
| **33060** | Open | tcp | mysqlx | Mysql X protocol  listener |

**Nmap scan report for 192.168.30.95**

Tab 21**: Port scanning by Nmap**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Port** | **State** | **Protocol** | **Service** | **Version** |
| **3389** | Open | tcp | ms-wbt-server | Microsoft Terminal Services |
| **7070** | Open | tcp | ssl/realserver |  |

**Nmap scan report for 192.168.30.98**

Tab 22**: Port scanning by Nmap**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Port State Protocol** | | | **Service** | **Version** |
| **135** | Open | tcp | msrpc | Microsoft windows RPC |
| **139** | Open | tcp | netbios-ssn | Microsoft windows netbios-ssn |
| **445** | Open | tcp | microsoft-ds |  |
| **5040** | Open | tcp | unknown |  |
| **5357** | Open | tcp | http |  |
| **49664** | Open | tcp | msrpc | Microsoft windows RPC |
| **49665** | Open | tcp | msrpc | Microsoft windows RPC |
| **49666** | Open | tcp | msrpc | Microsoft windows RPC |
| **49667** | Open | tcp | msrpc | Microsoft windows RPC |
| **49674** | Open | tcp | msrpc | Microsoft windows RPC |
| **49700** | Open | tcp | msrpc | Microsoft windows RPC |
| **49720** | Open | tcp | msrpc | Microsoft windows RPC |

**Nmap scan report for 192.168.30.103**

Tab 23**: Port scanning by Nmap**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Port State** | | | | **Protocol** | | **Service** | | **Version** | |
| **135** | | Open | | tcp | | msrpc | | Microsoft windows RPC | |
| **139** | | Open | | tcp | | msrpc | | Microsoft windows RPC | |
| **445** | | Open | | tcp | | msrpc | | Microsoft windows RPC | |
| **902** | | Open | | tcp | | msrpc | | Microsoft windows RPC | |
| **912** | | Open | | tcp | | msrpc | | Microsoft windows RPC | |
| **1521** | | Open | | tcp | | msrpc | | Microsoft windows RPC | |
| **3306** | | Open | | tcp | | msrpc | | Microsoft windows RPC | |
| **5040** | | Open | | tcp | | msrpc | | Microsoft windows RPC | |
| **5357** | | Open | | tcp | | msrpc | | Microsoft windows RPC | |
| **8080** | | Open | | tcp | | msrpc | | Microsoft windows RPC | |
| **33060** | | Open | | tcp | | msrpc | | Microsoft windows RPC | |
| **49664** | | Open | | tcp | | msrpc | | Microsoft windows RPC | |
| **49665** | | Open | | tcp | | msrpc | | Microsoft windows RPC | |
| **49666** | | Open | | tcp | | msrpc | | Microsoft windows RPC | |
| **49667** | | Open | | tcp | | msrpc | | Microsoft windows RPC | |
| **49668** | | Open | | tcp | | msrpc | | Microsoft windows RPC | |
| **49669** | | Open | | tcp | | msrpc | | Microsoft windows RPC | |
| **49670** | | Open | | tcp | | msrpc | | Microsoft windows RPC | |
| **49683** | | Open | | tcp | | msrpc | | Microsoft windows RPC | |

**Nmap scan report for 192.168.30.119**

Tab 24**: Port scanning by Nmap**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Port** | **State** | **Protocol** | **Service** | **Version** |
| **135** | Open | tcp | msrpc | Microsoft windows RPC |
| **139** | Open | tcp | netbios-ssn | Microsoft windows netbios-  ssn |
| **445** | Open | tcp | microsoft-ds |  |
| **3306** | Open | tcp | mysql | Mysql (unauthorized) |
| **5357** | Open | tcp | http | Microsoft  HTTPAPI httpd  2.0 |
| **7680** | Open | tcp | Pando-pub |  |
| **33060** | Open | tcp | mysqlx | Mysql X protocol  listener |
| **49668** | Open | tcp | msrpc | Microsoft windows RPC |

**Nmap scan report for 192.168.30.129**

Tab 25**: Port scanning by Nmap**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Port** | **State** | **Protocol** | **Service** | **Version** |
| **135** | Open | tcp | msrpc | Microsoft windows RPC |
| **139** | Open | tcp | msrpc | Microsoft windows RPC |
| **49664** | Open | tcp | msrpc | Microsoft windows RPC |
| **49666** | Open | tcp | msrpc | Microsoft windows RPC |

**Nmap scan report for 192.168.30.131**

Tab 26**: Port scanning by Nmap**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Port** | **State** | **Protocol** | **Service** | **Version** |
| **3360** | Open | tcp | mysql | MySQL  (unauthorized) |
| **7680** | Open | tcp | pando-pub |  |
| **33060** | Open | tcp | mysqlx | Mysql X protocol  listener |

**Nmap scan report for 192.168.30.133**

Tab 27**: Port scanning by Nmap**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Port** | **State** | **Protocol** | **Service** | **Version** |
| **5040** | Open | tcp | unknown |  |
| **5357** | Open | tcp | http | Microsoft  HTTPAPI httpd  2.0 |

**Nmap scan report for 192.168.30.137**

Tab 28**: Port scanning by Nmap**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Port** | **State** | **Protocol** | **Service** | **Version** |
| **3306** | Open | tcp | mysql | MySQL  (unauthorized |
| **3389** | Open | tcp | ms-wbt-server | Microsoft Terminal Services |
| **7070** | Open | tcp | ssl/realserver |  |
| **33060** | Open | tcp | mysqlx | Mysql X protocol  listener |

**Nmap scan report for 192.168.30.141**

Tab 29**: Port scanning by Nmap**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Port** | **State** | **Protocol** | **Service** | **Version** |
| **7680** | open | tcp | pando-pub |  |

**Nmap scan report for 192.168.30.151**

Tab 30**: Port scanning by Nmap**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Port** | **State** | **Protocol** | **Service** | **Version** |
| **5355** | open | tcp | llmr |  |
| **27500** | open | tcp | unknown |  |

**Nmap scan report for 192.168.30.154**

Tab 31**: Port scanning by Nmap**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Port** | **State** | **Protocol** | **Service** | **Version** |
| **49664** | open | tcp | msrpc | Microsoft windows RPC |
| **49665** | open | tcp | msrpc | Microsoft windows RPC |
| **49666** | open | tcp | msrpc | Microsoft windows RPC |
| **49667** | open | tcp | msrpc | Microsoft windows RPC |

**Nmap scan report for 192.168.30.155**

Tab 32**: Port scanning by Nmap**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Port** | **State** | **Protocol** | **Service** | **Version** |
| **80** | open | tcp | http | Apache httpd  2.4.58 |
| **443** | open | tcp | ssl/http | Apache httpd  2.4.58 |
| **3306** | open | tcp | mysql | Mariadb 10.3.23 |
| **7680** | open | tcp | pando-pub |  |

**Nmap scan report for 192.168.30.169**

Tab 33**: Port scanning by Nmap**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Port** | **State** | **Protocol** | **Service** | **Version** |
| **3389** | open | tcp | ms-wbt-server |  |

**Nmap scan report for 192.168.30.174**

Tab 34**: Port scanning by Nmap**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Port** | **State** | **Protocol** | **Service** | **Version** |
| **3389** | open | tcp | ms-wbt-server | Microsoft Terminal Services |
| **7680** | open | tcp | Pando-pub |  |

**Nmap scan report for 192.168.30.175** Tab 35**: Port scanning by Nmap**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Port** | **State** | **Protocol** | **Service** | **Version** |
| **3306** | open | tcp | mysql | MySQL  (unauthorized |
| **7070** | open | tcp | ssl/realserver |  |
| **7680** | open | tcp | Pando-pub |  |
| **33060** | open | tcp | mysqlx | Mysql X protocol  listener |

**Nmap scan report for 192.168.30.178**

Tab 36**: Port scanning by Nmap**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Port** | **State** | **Protocol** | **Service** | **Version** |
| **7680** | open | open | pando-pub |  |

**Nmap scan report for 192.168.30.181**

Tab 37**: Port scanning by Nmap**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Port** | **State** | **Protocol** | **Service** | **Version** |
| **135** | open | open | msrpc | Microsoft windows RPC |
| **139** | open | open | netbios-ssn | Microsoft windows netbios-  ssn |
| **445** | open | open | microsoft-ds |  |
| **3306** | open | open | mysql | MySQL  (unauthorized |
| **33060** | open | open | mysqlx | Mysql X protocol  listener |
| **49668** | open | open | msrpc | Microsoft windows RPC |

**Nmap scan report for 192.168.30.199**

Tab 38**: Port scanning by Nmap**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Port** | **State** | **Protocol** | **Service** | **Version** |
| **7680** | open | tcp | pando-pub |  |

**Nmap scan report for 192.168.30.202**

Tab 39**: Port scanning by Nmap**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Port** | **State** | **Protocol** | **Service** | **Version** |
| **135** | open | tcp | msrpc | Microsoft windows RPC |
| **139** | open | tcp | netbios-ssn | Microsoft windows netbios-  ssn |
| **445** | open | tcp | microsoft-ds |  |
| **7070** | open | tcp | pando-pub |  |
| **7680** | open | tcp | msrpc | Microsoft windows RPC |

**Nmap scan report for 192.168.30.211**

Tab 40**: Port scanning by Nmap**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Port** | **State** | **Protocol** | **Service** | **Version** |
| **7680** | open | tcp | pando-pub |  |

**Nmap scan report for 192.168.30.213**

Tab 41**: Port scanning by Nmap**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Port** | **State** | **Protocol** | **Service** | **Version** |
| **7680** | open | tcp | Pando-pub |  |
| **9009** | open | tcp | http | Microsoft  HTTPAPI httpd  2.0 |

**Nmap scan report for 192.168.30.243**

All 65535 scanned ports on 192.168.30.243 are in ignored states.

**Nmap scan report for 192.168.30.246**

Tab 42**: Port scanning by Nmap**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Port** | **State** | **Protocol** | **Service** | **Version** |
| **7680** | open | tcp | pando-pub |  |

**Nmap scan report for 192.168.30.148**

All 65535 scanned ports on 192.168.30.148 are in ignored states.

**3. Network Vulnerability Assessment by Nessus:**

**Vulnerabilities by Hosts**

**i. Ip Address**: 192.168.30.1



**Scan Information**

Start time: Tue Apr 29 13:24:42 2025

End Time: Tue Apr 29 13:33:58 2025

**Host Information**

IP: 192.168.30.1

MAC Address: 60:A4:B7:D7:F9:24

OS: TP-LINK Archer C6 1.0

**Vulnerability level: Critical**

**93650 - Dropbear SSH Server < 2016.72 Multiple Vulnerabilities**

**Synopsis**

The SSH service running on the remote host is affected by multiple vulnerabilities.

**Description**

According to its self-reported version in its banner, Dropbear SSH running on the remote host is prior to 2016.74. It is, therefore, affected by the following vulnerabilities:

* + - A format string flaw exists due to improper handling of string format specifiers (e.g., %s and %x) in usernames and host arguments. An unauthenticated, remote attacker can exploit this to execute arbitrary code with root privileges. (CVE-2016-7406)
    - A flaw exists in dropbearconvert due to improper handling of specially crafted OpenSSH key files. An unauthenticated, remote attacker can exploit this to execute arbitrary code. (CVE-2016-7407)
    - A flaw exists in dbclient when handling the -m or -c arguments in scripts. An unauthenticated, remote attacker can exploit this, via a specially crafted script, to execute arbitrary code. (CVE-2016-7408)- A flaw exists in dbclient or dropbear server if they are compiled with the DEBUG\_TRACE option and then run using the -v switch. A local attacker can exploit this to disclose process memory. (CVE-20167409)

**See Also** [https://matt.ucc.asn.au/dropbear/CHANGE](https://matt.ucc.asn.au/dropbear/CHANGES)  [S](https://matt.ucc.asn.au/dropbear/CHANGES)

**Solution**

Upgrade to Dropbear SSH version 2016.74 or later.

**Risk Factor**

Critical

**CVSS v3.0 Base Score**

9.8 (CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H)

**CVSS v3.0 Temporal Score**

8.5 (CVSS:3.0/E:U/RL:O/RC:C)

**VPR Score**

5.9

**EPSS Score**

0.0815

**CVSS v2.0 Base Score**

10.0 (CVSS2#AV:N/AC:L/Au:N/C:C/I:C/A:C)

**CVSS v2.0 Temporal Score**

7.4 (CVSS2#E:U/RL:OF/RC:C)

**Plugin Information**

Published: 2016/09/22, Modified: 2019/11/14



**Vulnerability level: Critical**



**Synopsis**

The remote service encrypts traffic using a protocol with known weaknesses.

**Description**

The remote service accepts connections encrypted using SSL 2.0 and/or SSL 3.0. These versions of SSL are affected by several cryptographic flaws, including:

* An insecure padding scheme with CBC ciphers.
* Insecure session renegotiation and resumption schemes.

An attacker can exploit these flaws to conduct man-in-the-middle attacks or to decrypt communications between the affected service and clients.

Although SSL/TLS has a secure means for choosing the highest supported version of the protocol (so that these versions will be used only if the client or server support nothing better), many web browsers implement this in an unsafe way that allows an attacker to downgrade a connection (such as in POODLE). Therefore, it is recommended that these protocols be disabled entirely.

NIST has determined that SSL 3.0 is no longer acceptable for secure communications. As of the date of enforcement found in PCI DSS v3.1, any version of SSL will not meet the PCI SSC's definition of 'strong cryptography'.

**See Also**

<https://www.schneier.com/academic/paperfiles/paper-ssl.pdf><http://www.nessus.org/u?b06c7e95><http://www.nessus.org/u?247c4540><https://www.openssl.org/~bodo/ssl-poodle.pdf><http://www.nessus.org/u?5d15ba70><https://www.imperialviolet.org/2014/10/14/poodle.html><https://tools.ietf.org/html/rfc7507>

[https://tools.ietf.org/html/rfc756](https://tools.ietf.org/html/rfc7568)  [8](https://tools.ietf.org/html/rfc7568)

**Solution**

Consult the application's documentation to disable SSL 2.0 and 3.0. Use TLS 1.2 (with approved cipher suites) or higher instead.

**Risk Factor**

Critical

**CVSS v3.0 Base Score**

9.8 (CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H)

**CVSS v2.0 Base Score**

10.0 (CVSS2#AV:N/AC:L/Au:N/C:C/I:C/A:C)

**Plugin Information**

Published: 2005/10/12, Modified: 2022/04/04



**Vulnerability level: Medium**



**Synopsis**

The remote DNS server is vulnerable to cache snooping attacks.

**Description**

The remote DNS server responds to queries for third-party domains that do not have the recursion bit set.

This may allow a remote attacker to determine which domains have recently been resolved via this name server, and therefore which hosts have been recently visited.

For instance, if an attacker was interested in whether your company utilizes the online services of a particular financial institution, they would be able to use this attack to build a statistical model regarding company usage of that financial institution. Of course, the attack can also be used to find B2B partners, web-surfing patterns, external mail servers, and more.

Note: If this is an internal DNS server not accessible to outside networks, attacks would be limited to the internal network. This may include employees, consultants and potentially users on a guest network or WiFi connection if supported.

**See Also**

<http://cs.unc.edu/~fabian/course_papers/cache_snooping.pdf>

**Solution**

Contact the vendor of the DNS software for a fix.

**Risk Factor**

Medium

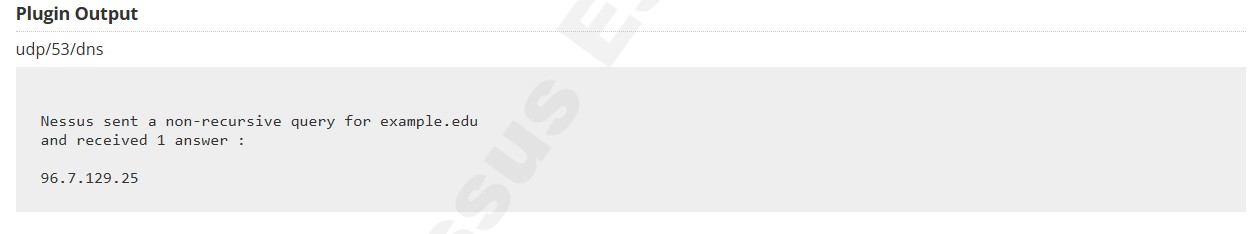
**CVSS v3.0 Base Score**

5.3 (CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N)

**CVSS v2.0 Base Score**

5.0 (CVSS2#AV:N/AC:L/Au:N/C:P/I:N/A:N)  **Plugin Information**

Published: 2004/04/27, Modified: 2020/04/07



**Vulnerability level: Medium**



**Synopsis**

The remote SSH service is affected by multiple vulnerabilities.

**Description**

According to its self-reported banner, the version of Dropbear SSH running on this port is earlier than 2013.59. As such, it is potentially affected by multiple vulnerabilities :

* A denial of service vulnerability caused by the way the 'buf\_decompress()' function handles compressed files.

(CVE-2013-4421)

* User-enumeration is possible due to a timing error when authenticating users. (CVE-2013-4434)

**See Also** <https://matt.ucc.asn.au/dropbear/CHANGES><https://secure.ucc.asn.au/hg/dropbear/rev/0bf76f54de6f><https://secure.ucc.asn.au/hg/dropbear/rev/a625f9e135a4>**Solution**

Upgrade to the Dropbear SSH 2013.59 or later.

**Risk Factor**

Medium

**VPR Score**

4.2

**EPSS Score**

0.2591

**CVSS v2.0 Base Score**

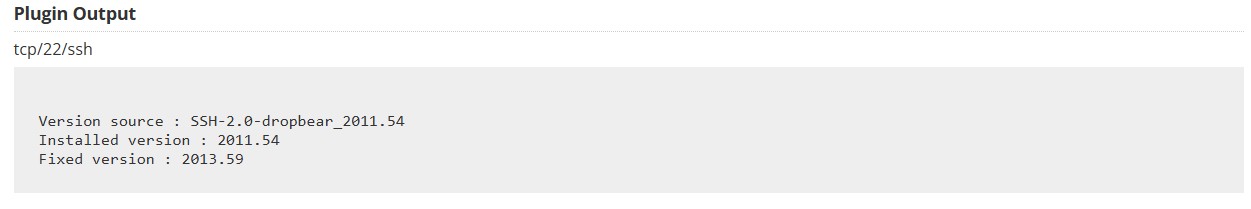
5.0 (CVSS2#AV:N/AC:L/Au:N/C:P/I:N/A:N)

**CVSS v2.0 Temporal Score**

3.7 (CVSS2#E:U/RL:OF/RC:C)

**Plugin Information**

Published: 2013/10/22, Modified: 2019/11/27



**Vulnerability Level: Medium**



**Synopsis**

The SSL certificate for this service cannot be trusted.

**Description**

The server's X.509 certificate cannot be trusted. This situation can occur in three different ways, in which the chain of trust can be broken, as stated below :

* First, the top of the certificate chain sent by the server might not be descended from a known public certificate authority. This can occur either when the top of the chain is an unrecognized, selfsigned certificate, or when intermediate certificates are missing that would connect the top of the certificate chain to a known public certificate authority.

* Second, the certificate chain may contain a certificate that is not valid at the time of the scan. This can occur either when the scan occurs before one of the certificate's 'notBefore' dates, or after one of the certificate's 'notAfter' dates.

* Third, the certificate chain may contain a signature that either didn't match the certificate's information or could not be verified. Bad signatures can be fixed by getting the certificate with the bad signature to be re-signed by its issuer. Signatures that could not be verified are the result of the certificate's issuer using a signing algorithm that Nessus either does not support or does not recognize.

If the remote host is a public host in production, any break in the chain makes it more difficult for users to verify the authenticity and identity of the web server. This could make it easier to carry out man-in-the-middle attacks against the remote host. **See Also** <https://www.itu.int/rec/T-REC-X.509/en><https://en.wikipedia.org/wiki/X.509>

**Solution**

Purchase or generate a proper SSL certificate for this service.

**Risk Factor**

Medium

**CVSS v3.0 Base Score**

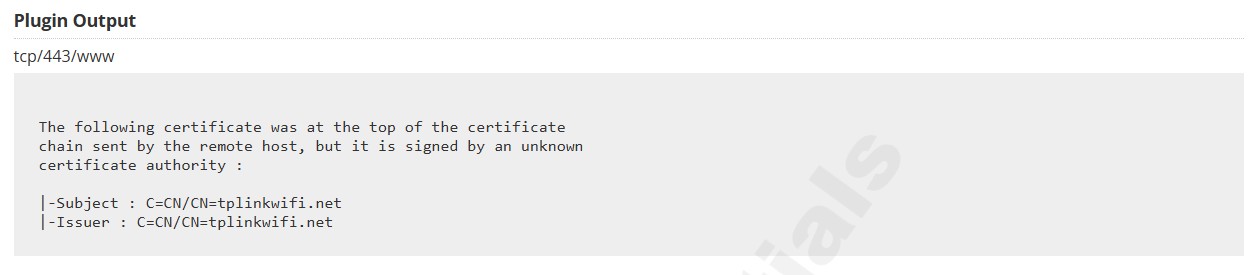
6.5 (CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:L/A:N)

**CVSS v2.0 Base Score**

6.4 (CVSS2#AV:N/AC:L/Au:N/C:P/I:P/A:N)

**Plugin Information**

Published: 2010/12/15, Modified: 2020/04/27



**Vulnerability Leve: Medium**



**Synopsis**

The remote service supports the use of the RC4 cipher.

**Description**

The remote host supports the use of RC4 in one or more cipher suites.

The RC4 cipher is flawed in its generation of a pseudo-random stream of bytes so that a wide variety of small biases are introduced into the stream, decreasing its randomness.

If plaintext is repeatedly encrypted (e.g., HTTP cookies), and an attacker is able to obtain many (i.e., tens of millions) ciphertexts, the attacker may be able to derive the plaintext. **See Also** <https://www.rc4nomore.com/><http://www.nessus.org/u?ac7327a0><http://cr.yp.to/talks/2013.03.12/slides.pdf><http://www.isg.rhul.ac.uk/tls/><https://www.imperva.com/docs/HII_Attacking_SSL_when_using_RC4.pdf>

**Solution**

Reconfigure the affected application, if possible, to avoid use of RC4 ciphers. Consider using TLS

1.2 with AES-GCM suites subject to browser and web server support.

**Risk Factor**

Medium

**CVSS v3.0 Base Score**

5.9 (CVSS:3.0/AV:N/AC:H/PR:N/UI:N/S:U/C:H/I:N/A:N)

**CVSS v3.0 Temporal Score**

5.4 (CVSS:3.0/E:U/RL:X/RC:C)

**VPR Score**

7.3

**EPSS Score**

0.9303

**CVSS v2.0 Base Score**

4.3 (CVSS2#AV:N/AC:M/Au:N/C:P/I:N/A:N)

**CVSS v2.0 Temporal Score**

3.7 (CVSS2#E:U/RL:ND/RC:C)

**Plugin Information**

Published: 2013/04/05, Modified: 2025/04/04



**Vulnerability Level: Medium**



**Synopsis**

The remote service encrypts traffic using an older version of TLS.

**Description**

The remote service accepts connections encrypted using TLS 1.0. TLS 1.0 has a number of cryptographic design flaws. Modern implementations of TLS 1.0 mitigate these problems, but newer versions of TLS like 1.2 and 1.3 are designed against these flaws and should be used whenever possible.

As of March 31, 2020, Endpoints that aren’t enabled for TLS 1.2 and higher will no longer function properly with major web browsers and major vendors.

PCI DSS v3.2 requires that TLS 1.0 be disabled entirely by June 30, 2018, except for POS POI terminals (and the SSL/TLS termination points to which they connect) that can be verified as not being susceptible to any known exploits. **See Also** <https://tools.ietf.org/html/draft-ietf-tls-oldversions-deprecate-00>

**Solution**

Enable support for TLS 1.2 and 1.3, and disable support for TLS 1.0.

**Risk Factor**

Medium

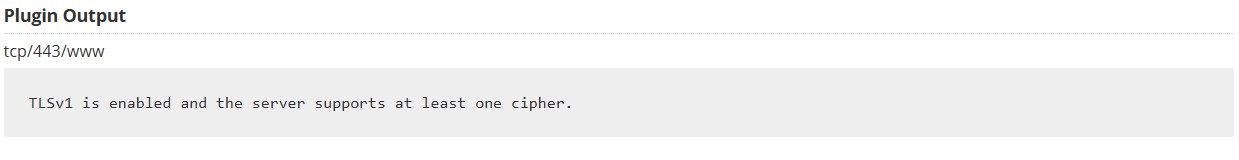
**CVSS v3.0 Base Score**

6.5 (CVSS:3.0/AV:N/AC:H/PR:N/UI:N/S:U/C:H/I:L/A:N)

**CVSS v2.0 Base Score**

6.1 (CVSS2#AV:N/AC:H/Au:N/C:C/I:P/A:N) **Plugin Information**

Published: 2017/11/22, Modified: 2023/04/19



**Vulnerability Level: Medium**



**Synopsis**

The remote service encrypts traffic using an older version of TLS.

**Description**

The remote service accepts connections encrypted using TLS 1.1. TLS 1.1 lacks support for current and recommended cipher suites. Ciphers that support encryption before MAC computation, and authenticated encryption modes such as GCM cannot be used with TLS 1.1

As of March 31, 2020, Endpoints that are not enabled for TLS 1.2 and higher will no longer function properly with major web browsers and major vendors. **See Also** <https://datatracker.ietf.org/doc/html/rfc8996><http://www.nessus.org/u?c8ae820d>

**Solution**

Enable support for TLS 1.2 and/or 1.3, and disable support for TLS 1.1.

**Risk Factor**

Medium

**CVSS v3.0 Base Score**

6.5 (CVSS:3.0/AV:N/AC:H/PR:N/UI:N/S:U/C:H/I:L/A:N)

**CVSS v2.0 Base Score**

6.1 (CVSS2#AV:N/AC:H/Au:N/C:C/I:P/A:N) **Plugin Information**

Published: 2022/04/04, Modified: 2024/05/14





**Synopsis**

The remote DHCP server may expose information about the associated network.

**Description**

This script contacts the remote DHCP server (if any) and attempts to retrieve information about the network layout.

Some DHCP servers provide sensitive information such as the NIS domain name, or network layout information such as the list of the network web servers, and so on.

It does not demonstrate any vulnerability, but a local attacker may use DHCP to become intimately familiar with the associated network.

**Solution**

Apply filtering to keep this information off the network and remove any options that are not in use.

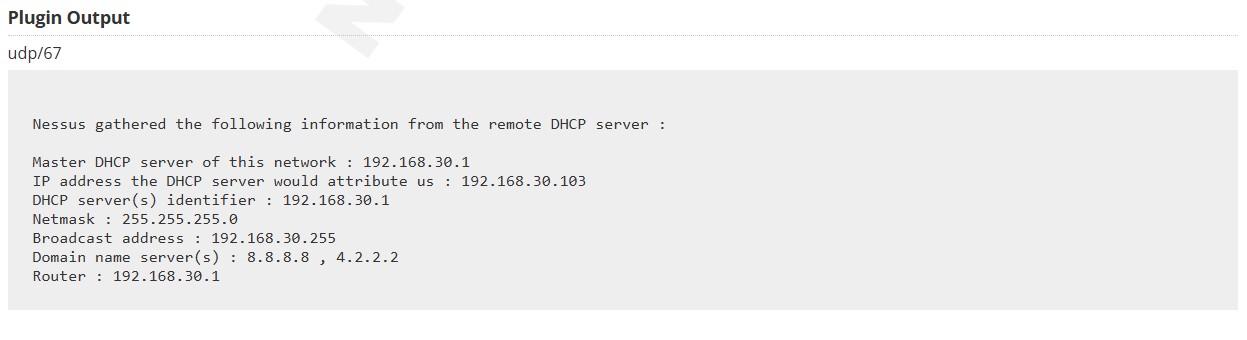
**Risk Factor**

Low

**CVSS v2.0 Base Score**

3.3 (CVSS2#AV:A/AC:L/Au:N/C:P/I:N/A:N) **Plugin Information**

Published: 2001/05/05, Modified: 2019/03/06





**Synopsis**

It is possible to determine the exact time set on the remote host.

**Description**

The remote host answers to an ICMP timestamp request. This allows an attacker to know the date that is set on the targeted machine, which may assist an unauthenticated, remote attacker in defeating time-based authentication protocols.

Timestamps returned from machines running Windows Vista / 7 / 2008 / 2008 R2 are deliberately incorrect, but usually within 1000 seconds of the actual system time.

**Solution**

Filter out the ICMP timestamp requests (13), and the outgoing ICMP timestamp replies (14).

**Risk Factor**

Low

**VPR Score**

2.2

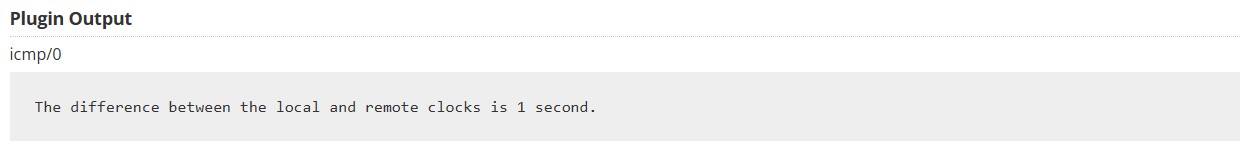
**EPSS Score**

0.0037

**CVSS v2.0 Base Score**

2.1 (CVSS2#AV:L/AC:L/Au:N/C:P/I:N/A:N) **Plugin Information**

Published: 1999/08/01, Modified: 2024/10/07





**Synopsis**

The remote host appears to leak memory in network packets.

**Description**

The remote host uses a network device driver that pads ethernet frames with data which vary from one packet to another, likely taken from kernel memory, system memory allocated to the device driver, or a hardware buffer on its network interface card.

Known as 'Etherleak', this information disclosure vulnerability may allow an attacker to collect sensitive information from the affected host provided he is on the same physical subnet as that host. **See Also** <http://www.nessus.org/u?719c90b4>

**Solution**

Contact the network device driver's vendor for a fix.

**Risk Factor**

Low

**VPR Score**

4.2

**EPSS Score**

0.0185

**CVSS v2.0 Base Score**

3.3 (CVSS2#AV:A/AC:L/Au:N/C:P/I:N/A:N)

**CVSS v2.0 Temporal Score**

2.6 (CVSS2#E:POC/RL:OF/RC:C)

**Plugin Information**

Published: 2003/01/14, Modified: 2019/03/06



**Vulnerability Level: Low**



**Synopsis**

The SSH server is configured to use Cipher Block Chaining.

**Description**

The SSH server is configured to support Cipher Block Chaining (CBC) encryption. This may allow an attacker to recover the plaintext message from the ciphertext.

Note that this plugin only checks for the options of the SSH server and does not check for vulnerable software versions.

**Solution**

Contact the vendor or consult product documentation to disable CBC mode cipher encryption, and enable CTR or GCM cipher mode encryption.

**Risk Factor**

Low

**CVSS v3.0 Base Score**

3.7 (CVSS:3.0/AV:N/AC:H/PR:N/UI:N/S:U/C:L/I:N/A:N)

**VPR Score**

6.5

**EPSS Score**

0.0307

**CVSS v2.0 Base Score**

2.6 (CVSS2#AV:N/AC:H/Au:N/C:P/I:N/A:N)

**CVSS v2.0 Temporal Score**

1.9 (CVSS2#E:U/RL:OF/RC:C)

**Plugin Information**

Published: 2013/10/28, Modified: 2023/10/27



**Vulnerability Level: Low**



**Synopsis**

The remote SSH server is configured to allow weak key exchange algorithms.

**Description**

The remote SSH server is configured to allow key exchange algorithms which are considered weak.

This is based on the IETF draft document Key Exchange (KEX) Method Updates and Recommendations for Secure Shell (SSH) RFC9142. Section 4 lists guidance on key exchange algorithms that SHOULD NOT and MUST NOT be enabled. This includes:

diffie-hellman-group-exchange-sha1

diffie-hellman-group1-sha1

gss-gex-sha1-\*

gss-group1-sha1-\*

gss-group14-sha1-\*

rsa1024-sha1

Note that this plugin only checks for the options of the SSH server, and it does not check for vulnerable software versions. **See Also** <https://datatracker.ietf.org/doc/html/rfc9142>

**Solution**

Contact the vendor or consult product documentation to disable the weak algorithms.

**Risk Factor**

Low

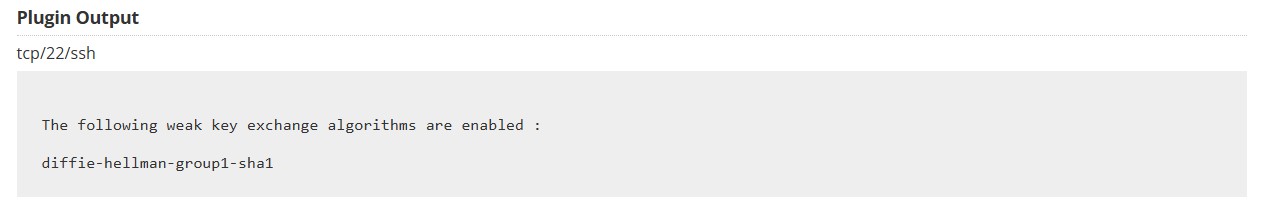
**CVSS v3.0 Base Score**

3.7 (CVSS:3.0/AV:N/AC:H/PR:N/UI:N/S:U/C:L/I:N/A:N)

**CVSS v2.0 Base Score**

2.6 (CVSS2#AV:N/AC:H/Au:N/C:P/I:N/A:N) **Plugin Information**

Published: 2021/10/13, Modified: 2024/03/22



**Vulnerability Level: Low**



**Synopsis**

The remote SSH server is configured to allow MD5 and 96-bit MAC algorithms.

**Description**

The remote SSH server is configured to allow either MD5 or 96-bit MAC algorithms, both of which are considered weak.

Note that this plugin only checks for the options of the SSH server, and it does not check for vulnerable software versions.

**Solution**

Contact the vendor or consult product documentation to disable MD5 and 96-bit MAC algorithms.

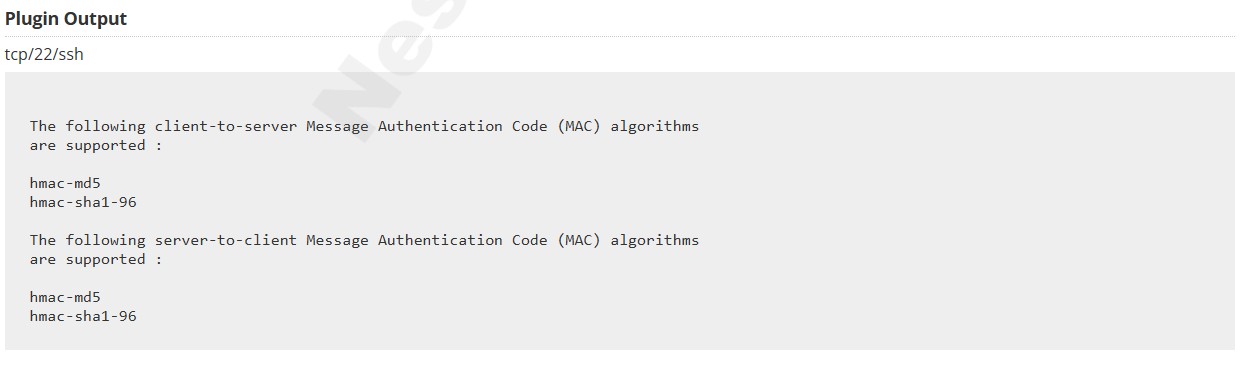
**Risk Factor**

Low

**CVSS v2.0 Base Score**

2.6 (CVSS2#AV:N/AC:H/Au:N/C:P/I:N/A:N) **Plugin Information**

Published: 2013/11/22, Modified: 2016/12/14



**Vulnerability Level: Low**



**Synopsis**

The X.509 certificate chain used by this service contains certificates with RSA keys shorter than 2048 bits.

**Description**

At least one of the X.509 certificates sent by the remote host has a key that is shorter than 2048 bits. According to industry standards set by the Certification Authority/Browser (CA/B) Forum, certificates issued after January 1, 2014 must be at least 2048 bits.

Some browser SSL implementations may reject keys less than 2048 bits after January 1, 2014. Additionally, some SSL certificate vendors may revoke certificates less than 2048 bits before January 1, 2014.

Note that Nessus will not flag root certificates with RSA keys less than 2048 bits if they were issued prior to December 31, 2010, as the standard considers them exempt. **See Also** <https://www.cabforum.org/wp-content/uploads/Baseline_Requirements_V1.pdf>

**Solution**

Replace the certificate in the chain with the RSA key less than 2048 bits in length with a longer key, and reissue any certificates signed by the old certificate.

**Risk Factor**

Low

**Plugin Information**

Published: 2013/09/03, Modified: 2018/11/15

 **i. IP Address: 192.168.30.103**



**Scan Information:**

Start Date: Tue Apr 29 13:24:42 2025

End Date: Tue Apr 29 13:33:48 2025

**Host Information:**

Netbios Name: WROOM

IP: 192.168.30.103

OS: Microsoft Windows

Vulnerability Level: Critical



**Synopsis**

The remote host is running an unsupported version of a database server.

**Description**

According to its version, the installation of Oracle Database running on the remote host is no longer supported.

Lack of support implies that no new security patches for the product will be released by the vendor. As a result, it is likely to contain security vulnerabilities.

**See Also** <http://www.nessus.org/u?ccd068d1>

**Solution**

Upgrade to a version of Oracle Database that is currently supported.

**Risk Factor**

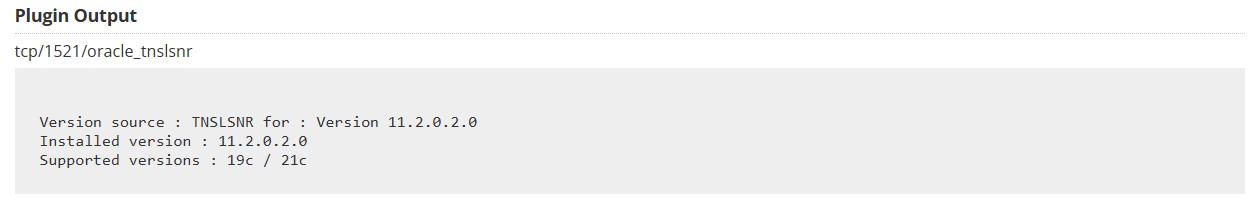
Critical

**CVSS v2.0 Base Score**

10.0 (CVSS2#AV:N/AC:L/Au:N/C:C/I:C/A:C)

**Plugin Information**

Published: 2011/08/09, Modified: 2022/09/28



Vulnerability Level: High



**Synopsis**

It was possible to register with a remote Oracle TNS listener.

**Description**

The remote Oracle TNS listener allows service registration from a remote host. An attacker can exploit this issue to divert data from a legitimate database server or client to an attacker-specified system.

Successful exploits will allow the attacker to manipulate database instances, potentially facilitating man-in-the-middle, session- hijacking, or denial of service attacks on a legitimate database server. **See Also**

<http://www.nessus.org/u?8c8334e6><http://www.nessus.org/u?06d298e5><https://seclists.org/fulldisclosure/2012/Apr/204>

**Solution**

Apply the workaround in Oracle's advisory.

**Risk Factor**

High

**CVSS v3.0 Base Score**

7.3 (CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:L/A:L)

**CVSS v3.0 Temporal Score**

6.8 (CVSS:3.0/E:F/RL:O/RC:C)

**VPR Score**

4.9

**EPSS Score**

0.9141

**CVSS v2.0 Base Score**

7.5 (CVSS2#AV:N/AC:L/Au:N/C:P/I:P/A:P)

**CVSS v2.0 Temporal Score**

6.8 (CVSS2#E:F/RL:W/RC:C)

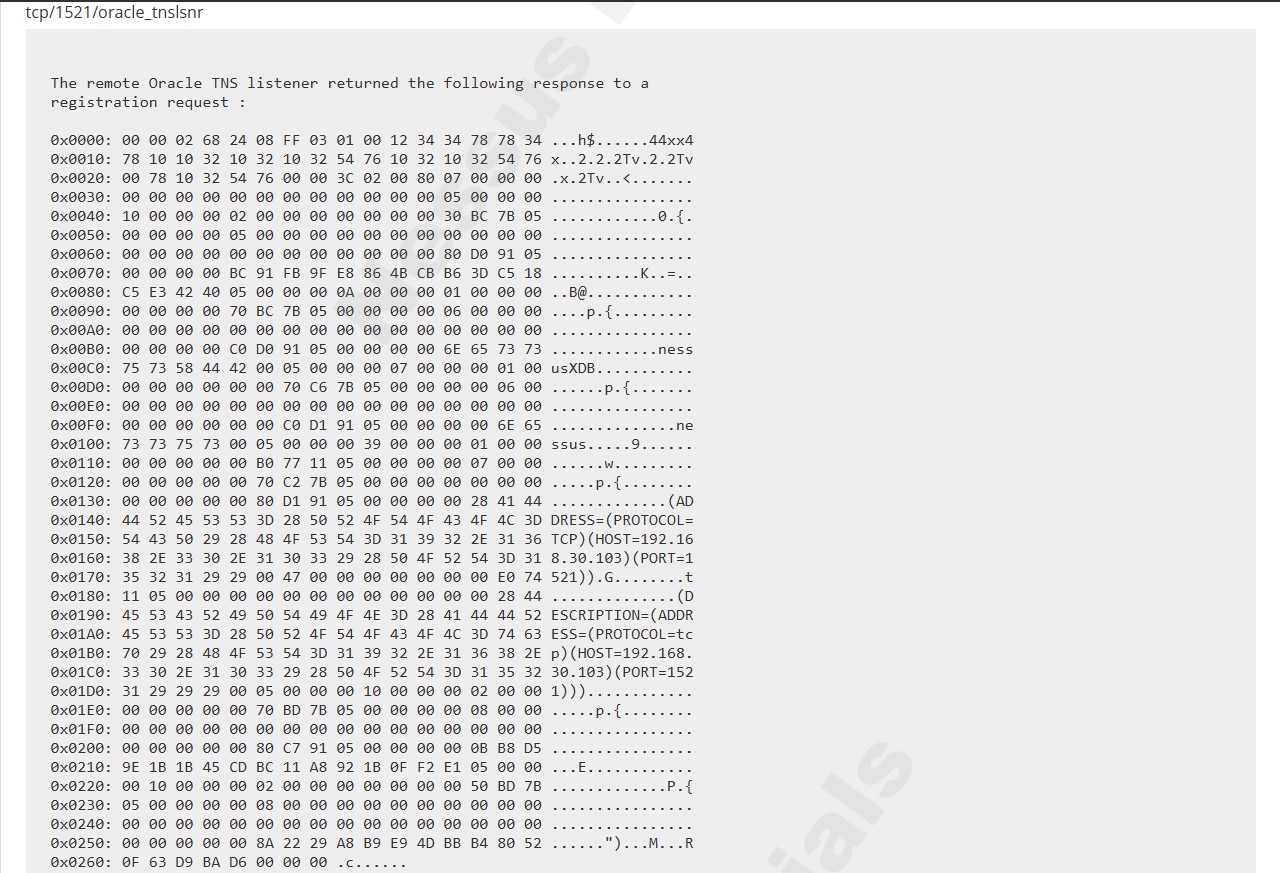
**Exploitable With**

Core Impact (true)

**Plugin Information**

Published: 2013/08/26, Modified: 2025/03/31

**Plugin Output**



Vulnerability Level: Medium



**Synopsis**

The remote host is running a vulnerable version of Oracle Apex.

**Description**

An unspecified vulnerability in versions 3.2 and 4.0 of the Application Express (Apex) component of the Oracle Database Server allows remote, authenticated users to affect confidentiality, integrity, and availability, relating to the Apex developer user.

**See Also**

<http://www.oracle.com/technetwork/developer-tools/apex/index.html><https://www.oracle.com/technetwork/topics/security/cpuoct2011-330135.html><https://www.recx.co.uk/downloads/Recx-Apex-CVE-2011-3525.pdf>

**Solution**

Upgrade Application Express to at least version 4.1.

**Risk Factor**

Medium

**VPR Score**

5.5

**EPSS Score**

0.0057

**CVSS v2.0 Base Score**

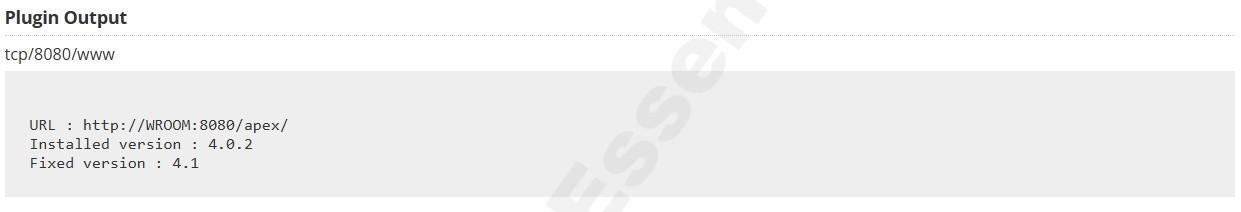
6.5 (CVSS2#AV:N/AC:L/Au:S/C:P/I:P/A:P)

**CVSS v2.0 Temporal Score**

4.8 (CVSS2#E:U/RL:OF/RC:C)

**Plugin Information**

Published: 2013/02/20, Modified: 2020/06/12



Vulnerability Level: Medium



**Synopsis**

The remote host is running a vulnerable version of Oracle Apex.

**Description**

An unspecified vulnerability in versions 4.0 and 4.1 of the Application Express (Apex) component of the Oracle Database Server allows remote attackers to affect integrity via unpublished vectors.

**See Also**

<http://www.oracle.com/technetwork/developer-tools/apex/index.html><https://www.oracle.com/technetwork/topics/security/cpuapr2012-366314.html><https://www.recx.co.uk/downloads/Recx-Apex-CVE-2012-1708.pdf>

**Solution**

Upgrade Application Express to at least version 4.1.1.

**Risk Factor**

Medium

**VPR Score**

2.7

**EPSS Score**

0.0039

**CVSS v2.0 Base Score**

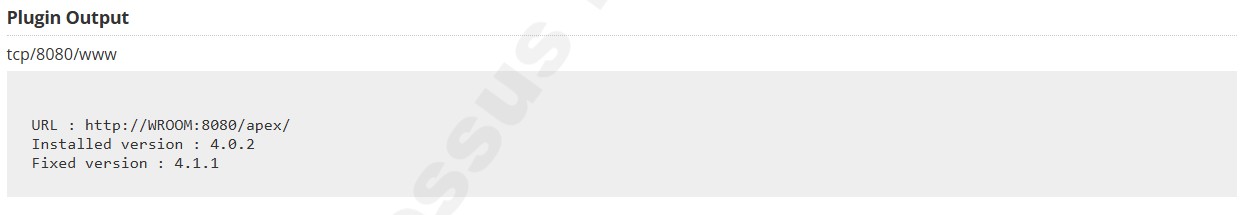
4.3 (CVSS2#AV:N/AC:M/Au:N/C:N/I:P/A:N)

**CVSS v2.0 Temporal Score**

3.2 (CVSS2#E:U/RL:OF/RC:C)

**Plugin Information**

Published: 2013/02/20, Modified: 2020/06/12



Vulnerability Level: Medium



**Synopsis**

Signing is not required on the remote SMB server.

**Description**

Signing is not required on the remote SMB server. An unauthenticated, remote attacker can exploit this to conduct man-in-the-middle attacks against the SMB server.

**See Also**

<http://www.nessus.org/u?df39b8b3><http://technet.microsoft.com/en-us/library/cc731957.aspx><http://www.nessus.org/u?74b80723><https://www.samba.org/samba/docs/current/man-html/smb.conf.5.html><http://www.nessus.org/u?a3cac4ea>

**Solution**

Enforce message signing in the host's configuration. On Windows, this is found in the policy setting 'Microsoft network server: Digitally sign communications (always)'. On Samba, the setting is called 'server signing'. See the 'see also' links for further details.

**Risk Factor**

Medium

**CVSS v3.0 Base Score**

5.3 (CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:L/A:N)

**CVSS v3.0 Temporal Score**

4.6 (CVSS:3.0/E:U/RL:O/RC:C)

**CVSS v2.0 Base Score**

5.0 (CVSS2#AV:N/AC:L/Au:N/C:N/I:P/A:N)

**CVSS v2.0 Temporal Score**

3.7 (CVSS2#E:U/RL:OF/RC:C)

**Plugin Information**

Published: 2012/01/19, Modified: 2022/10/05

**Plugin Output** tcp/445/cifs

**Vulnerability Level: Medium**



**Synopsis**

The SSL certificate for this service cannot be trusted.

**Description**

The server's X.509 certificate cannot be trusted. This situation can occur in three different ways, in which the chain of trust can be broken, as stated below :

* First, the top of the certificate chain sent by the server might not be descended from a known public certificate authority. This can occur either when the top of the chain is an unrecognized, self-signed certificate, or when intermediate certificates are missing that would connect the top of the certificate chain to a known public certificate authority.

* Second, the certificate chain may contain a certificate that is not valid at the time of the scan. This can occur either when the scan occurs before one of the certificate's 'notBefore' dates, or after one of the certificate's 'notAfter' dates.

* Third, the certificate chain may contain a signature that either didn't match the certificate's information or could not be verified. Bad signatures can be fixed by getting the certificate with the bad signature to be re-signed by its issuer. Signatures that could not be verified are the result of the certificate's issuer using a signing algorithm that Nessus either does not support or does not recognize.

If the remote host is a public host in production, any break in the chain makes it more difficult for users to verify the authenticity and identity of the web server. This could make it easier to carry out man-in-the-middle attacks against the remote host.

**See Also**

<https://www.itu.int/rec/T-REC-X.509/en><https://en.wikipedia.org/wiki/X.509>

**Solution**

Purchase or generate a proper SSL certificate for this service.

**Risk Factor**

Medium

**CVSS v3.0 Base Score**

6.5 (CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:L/A:N)

**CVSS v2.0 Base Score**

6.4 (CVSS2#AV:N/AC:L/Au:N/C:P/I:P/A:N)

**Plugin Information**

Published: 2010/12/15, Modified: 2020/04/27



**ii. IP Address: 192.168.30.155**



**Scan Information:**

Start time: Tue Apr 29 13:31:03 2025

End time: Tue Apr 29 13:43:04 2025

**Host Information:**

**IP: 192.168.30.155**

**MAC Address: B4:2E:99:80:08:FD**

**OS: Microsoft Windows 10 Enterprise**

**Vulnerability Level: Critical**



**Synopsis**

The remote web server is affected by multiple vulnerabilities.

**Description**

The version of Apache httpd installed on the remote host is prior to 2.4.60. It is, therefore, affected by multiple vulnerabilities as referenced in the 2.4.60 advisory.

* Serving WebSocket protocol upgrades over a HTTP/2 connection could result in a Null Pointer dereference, leading to a crash of the server process, degrading performance. (CVE-2024-36387)

* SSRF in Apache HTTP Server on Windows allows to potentially leak NTML hashes to a malicious server via SSRF and malicious requests or content Users are recommended to upgrade to version 2.4.60 which fixes this issue. Note: Existing configurations that access UNC paths will have to configure new directive UNCList to allow access during request processing. (CVE-2024-38472)

* Encoding problem in mod\_proxy in Apache HTTP Server 2.4.59 and earlier allows request URLs with incorrect encoding to be sent to backend services, potentially bypassing authentication via crafted requests. Users are recommended to upgrade to version 2.4.60, which fixes this issue. (CVE2024-38473)

* Substitution encoding issue in mod\_rewrite in Apache HTTP Server 2.4.59 and earlier allows attacker to execute scripts in directories permitted by the configuration but not directly reachable by any URL or source disclosure of scripts meant to only to be executed as CGI. Users are recommended to upgrade to version 2.4.60, which fixes this issue. Some RewriteRules that capture and substitute unsafely will now fail unless rewrite flag UnsafeAllow3F is specified. (CVE-202438474)

* Improper escaping of output in mod\_rewrite in Apache HTTP Server 2.4.59 and earlier allows an attacker to map URLs to filesystem locations that are permitted to be served by the server but are not intentionally/directly reachable by any URL, resulting in code execution or source code disclosure.

Substitutions in server context that use a backreferences or variables as the first segment of the substitution are affected. Some unsafe RewiteRules will be broken by this change and the rewrite flag UnsafePrefixStat can be used to opt back in once ensuring the substitution is appropriately constrained.

(CVE-2024-38475)

* Vulnerability in core of Apache HTTP Server 2.4.59 and earlier are vulnerably to information disclosure, SSRF or local script execution via backend applications whose response headers are malicious or exploitable. Users are recommended to upgrade to version 2.4.60, which fixes this issue. (CVE-2024-38476)

* null pointer dereference in mod\_proxy in Apache HTTP Server 2.4.59 and earlier allows an attacker to crash the server via a malicious request. Users are recommended to upgrade to version

2.4.60, which fixes this issue. (CVE-2024-38477)

* Potential SSRF in mod\_rewrite in Apache HTTP Server 2.4.59 and earlier allows an attacker to cause unsafe RewriteRules to unexpectedly setup URL's to be handled by mod\_proxy. Users are recommended to upgrade to version 2.4.60, which fixes this issue. (CVE-2024-39573)

Note that Nessus has not tested for these issues but has instead relied only on the application's selfreported version number.

**Solution**

Upgrade to Apache version 2.4.60 or later.

**Risk Factor**

Critical

**CVSS v3.0 Base Score**

9.8 (CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H)

**CVSS v3.0 Temporal Score**

8.8 (CVSS:3.0/E:P/RL:O/RC:C)

**VPR Score**

6.7

**EPSS Score**

0.8354

**CVSS v2.0 Base Score**

10.0 (CVSS2#AV:N/AC:L/Au:N/C:C/I:C/A:C)

**CVSS v2.0 Temporal Score**

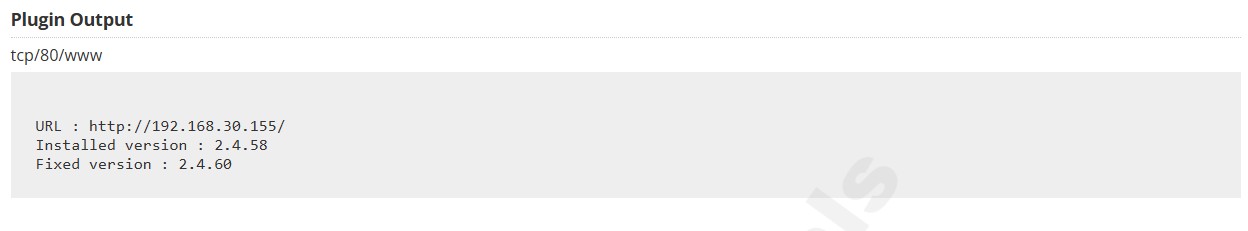
7.8 (CVSS2#E:POC/RL:OF/RC:C)

**STIG Severity**

I

**Plugin Information**

Published: 2024/07/01, Modified: 2024/08/22



Vulnerability Level: Critical

**Synopsis**

The remote service is affected by a vulnerability.

**Description**

The version of OpenSSL installed on the remote host is prior to 3.1.7. It is, therefore, affected by a vulnerability as referenced in the 3.1.7 advisory.

* Issue summary: Calling the OpenSSL API function SSL\_select\_next\_proto with an empty supported client protocols buffer may cause a crash or memory contents to be sent to the peer. Impact summary: A buffer overread can have a range of potential consequences such as unexpected application behavior or a crash.

In particular this issue could result in up to 255 bytes of arbitrary private data from memory being sent to the peer leading to a loss of confidentiality. However, only applications that directly call the SSL\_select\_next\_proto function with a 0 length list of supported client protocols are affected by this issue. This would normally never be a valid scenario and is typically not under attacker control but may occur by accident in the case of a configuration or programming error in the calling application. The OpenSSL API function SSL\_select\_next\_proto is typically used by

TLS applications that support ALPN (Application Layer Protocol Negotiation) or NPN (Next

Protocol Negotiation). NPN is older, was never standardized and is deprecated in favor of ALPN.

We believe that ALPN is significantly more widely deployed than NPN. The

SSL\_select\_next\_proto function accepts a list of protocols from the server and a list of protocols from the client and returns the first protocol that appears in the server list that also appears in the client list. In the case of no overlap between the two lists it returns the first item in the client list. In either case it will signal whether an overlap between the two lists was found. In the case where SSL\_select\_next\_proto is called with a zero-length client list it fails to notice this condition and returns the memory immediately following the client list pointer (and reports that there was no overlap in the lists). This function is typically called from a server-side application callback for ALPN or a client-side application callback for NPN. In the case of ALPN the list of protocols supplied by the client is guaranteed by libssl to never be zero in length. The list of server protocols comes from the application and should never normally be expected to be of zero length. In this case if the SSL\_select\_next\_proto function has been called as expected (with the list supplied by the client passed in the client/client\_len parameters), then the application will not be vulnerable to this issue. If the application has accidentally been configured with a zero-length server list, and has accidentally passed that zero length server list in the client/client\_len parameters, and has additionally failed to correctly handle a no overlap response (which would normally result in a handshake failure in ALPN) then it will be vulnerable to this problem. In the case of NPN, the protocol permits the client to opportunistically select a protocol when there is no overlap. OpenSSL returns the first client protocol in the no overlap case in support of this. The list of client protocols comes from the application and should never normally be expected to be of zero length. However, if the SSL\_select\_next\_proto function is accidentally called with a client\_len of 0 then an invalid memory pointer will be returned instead. If the application uses this output as the opportunistic protocol, then the loss of confidentiality will occur. This issue has been assessed as Low severity because applications are most likely to be vulnerable if they are using NPN instead of ALPN - but NPN is not widely used. It also requires an application configuration or programming error. Finally, this issue would not typically be under attacker control making active exploitation unlikely. The FIPS modules in 3.3, 3.2, 3.1 and 3.0 are not affected by this issue. Due to the low severity of this issue, we are not issuing new releases of

OpenSSL at this time. The fix will be included in the next releases when they become available.

Found by Joseph Birr-Pixton. Thanks to David Benjamin (Google). Fix developed by Matt Caswell. Fixed in OpenSSL 3.3.2 (Affected since 3.3.0). (CVE-2024-5535)

Note that Nessus has not tested for this issue but has instead relied only on the application's selfreported version number.

**See Also**

<http://www.nessus.org/u?f87142a6><https://www.cve.org/CVERecord?id=CVE-2024-5535>

**Solution**

Upgrade to OpenSSL version 3.1.7 or later.

**Risk Factor**

Medium

**CVSS v3.0 Base Score**

9.1 (CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:N/A:H)

**CVSS v3.0 Temporal Score**

8.2 (CVSS:3.0/E:P/RL:O/RC:C)

**VPR Score**

6.0

**EPSS Score**

0.1077

**CVSS v2.0 Base Score**

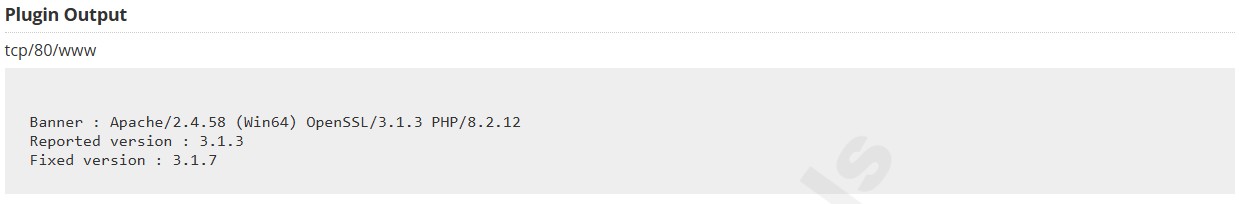
4.3 (CVSS2#AV:N/AC:M/Au:N/C:N/I:N/A:P)

**CVSS v2.0 Temporal Score**

3.4 (CVSS2#E:POC/RL:OF/RC:C)

**Plugin Information**

Published: 2024/06/27, Modified: 2025/04/14



**Vulnerability Level: Critical**



**Synopsis**

The version PHP running on the remote web server is affected by multiplevulnerabilities.

**Description**

The version of PHP installed on the remote host is prior to 8.2.20. It is, therefore, affected by multiple vulnerabilities as referenced in the Version 8.2.20 advisory.

* In PHP versions 8.1.\* before 8.1.29, 8.2.\* before 8.2.20, 8.3.\* before 8.3.8, when using Apache and PHP- CGI on Windows, if the system is set up to use certain code pages, Windows may use Best-Fit behavior to replace characters in command line given to Win32 API functions. PHP CGI module may misinterpret those characters as PHP options, which may allow a malicious user to pass options to PHP binary being run, and thus reveal the source code of scripts, run arbitrary PHP code on the server, etc. (CVE-2024-4577)

* In PHP versions 8.1.\* before 8.1.29, 8.2.\* before 8.2.20, 8.3.\* before 8.3.8, due to a code logic error, filtering functions such as filter\_var when validating URLs (FILTER\_VALIDATE\_URL) for certain types of URLs the function will result in invalid user information (username + password part of URLs) being treated as valid user information. This may lead to the downstream code accepting invalid URLs as valid and parsing them incorrectly. (CVE-20245458)

* In PHP versions 8.1.\* before 8.1.29, 8.2.\* before 8.2.20, 8.3.\* before 8.3.8, the fix for CVE2024-1874 does not work if the command name includes trailing spaces. Original issue: when using proc\_open() command with array syntax, due to insufficient escaping, if the arguments of the executed command are controlled by a malicious user, the user can supply arguments that would execute arbitrary commands in Windows shell.

(CVE-2024-5585)

Note that Nessus has not tested for these issues but has instead relied only on the application's self-reported version number.

**See Also** <http://php.net/ChangeLog-8.php#8.2.20>

**Solution**

Upgrade to PHP version 8.2.20 or later.

**Risk Factor**

Critical

**CVSS v3.0 Base Score**

9.8 (CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H)

**CVSS v3.0 Temporal Score**

9.4 (CVSS:3.0/E:H/RL:O/RC:C)

**VPR Score**

9.6

**EPSS Score**

0.9438

**CVSS v2.0 Base Score**

10.0 (CVSS2#AV:N/AC:L/Au:N/C:C/I:C/A:C)

**CVSS v2.0 Temporal Score**

8.7 (CVSS2#E:H/RL:OF/RC:C)

**STIG Severity**

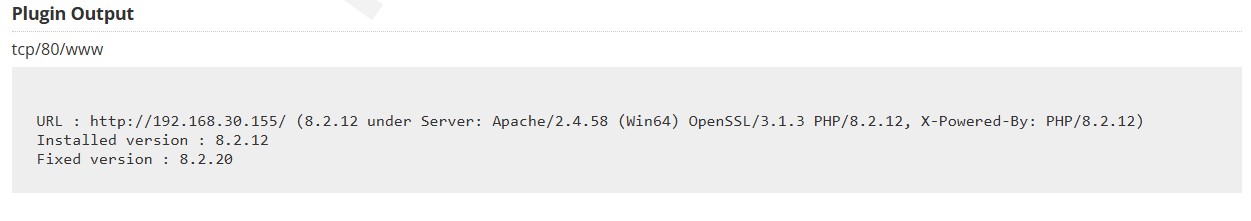
I

**Exploitable With**

CANVAS (true) Core Impact (true) Metasploit (true)

**Plugin Information**

Published: 2024/06/06, Modified: 2024/10/23



Vulnerability Level: Critical



**Synopsis**

The version PHP running on the remote web server is affected by multiple vulnerabilities.

**Description**

The version of PHP installed on the remote host is prior to 8.2.26. It is, therefore, affected by multiple vulnerabilities as referenced in the Version 8.2.26 advisory.

Note that Nessus has not tested for these issues but has instead relied only on the application's self-reported version number.

**See Also**

<http://php.net/ChangeLog-8.php#8.2.26><https://github.com/php/php-src/security/advisories/GHSA-5hqh-c84r-qjcv><https://github.com/php/php-src/security/advisories/GHSA-c5f2-jwm7-mmq2><https://github.com/php/php-src/security/advisories/GHSA-g665-fm4p-vhff><https://github.com/php/php-src/security/advisories/GHSA-h35g-vwh6-m678><https://github.com/php/php-src/security/advisories/GHSA-r977-prxv-hc43>

**Solution**

Upgrade to PHP version 8.2.26 or later.

**Risk Factor**

Critical

**CVSS v3.0 Base Score**

9.8 (CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H)

**CVSS v3.0 Temporal Score**

8.8 (CVSS:3.0/E:P/RL:O/RC:C)

**VPR Score**

6.7

**EPSS Score**

0.0014

**CVSS v2.0 Base Score**

10.0 (CVSS2#AV:N/AC:L/Au:N/C:C/I:C/A:C)

**CVSS v2.0 Temporal Score**

7.8 (CVSS2#E:POC/RL:OF/RC:C)

**STIG Severity**

I

**Plugin Information**

Published: 2024/11/21, Modified: 2025/03/21



Vulnerability Level: High



**Synopsis**

The remote web server is affected by multiple vulnerabilities.

**Description**

The version of Apache httpd installed on the remote host is prior to 2.4.59. It is, therefore, affected by multiple vulnerabilities as referenced in the 2.4.59 advisory.

* Apache HTTP Server: HTTP Response Splitting in multiple modules: HTTP Response splitting in multiple modules in Apache HTTP Server allows an attacker that can inject malicious response headers into backend applications to cause an HTTP desynchronization attack. Users are recommended to upgrade to version 2.4.59, which fixes this issue.

Acknowledgements: (CVE-2024-24795)

* Apache HTTP Server: HTTP/2 DoS by memory exhaustion on endless continuation frames: HTTP/2 incoming headers exceeding the limit are temporarily buffered in nghttp2 in order to generate an informative HTTP 413 response. If a client does not stop sending headers, this leads to memory exhaustion.

Acknowledgements: finder: Bartek Nowotarski (https://nowotarski.info/) (CVE-2024-27316)

Note that Nessus has not tested for these issues but has instead relied only on the application's self-reported version number.

**Solution**

Upgrade to Apache version 2.4.59 or later.

**Risk Factor**

High

**CVSS v3.0 Base Score**

7.5 (CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:H)

**CVSS v3.0 Temporal Score**

6.5 (CVSS:3.0/E:U/RL:O/RC:C)

**VPR Score**

4.4

**EPSS Score**

0.9045

**CVSS v2.0 Base Score**

7.8 (CVSS2#AV:N/AC:L/Au:N/C:N/I:N/A:C)

**CVSS v2.0 Temporal Score**

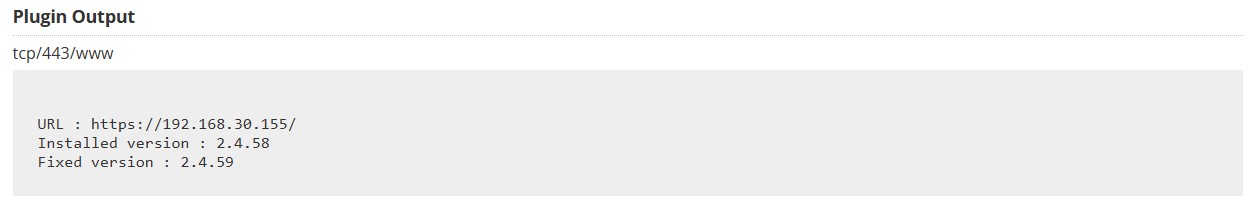
5.8 (CVSS2#E:U/RL:OF/RC:C)

**STIG Severity**

I

**Plugin Information**

Published: 2024/04/04, Modified: 2024/07/12



Vulnerability Level: High



**Synopsis**

The remote web server is affected by multiple vulnerabilities.

**Description**

The version of Apache httpd installed on the remote host is prior to 2.4.62. It is, therefore, affected by multiple vulnerabilities as referenced in the 2.4.62 advisory.

* SSRF in Apache HTTP Server on Windows with mod\_rewrite in server/vhost context, allows to potentially leak NTML hashes to a malicious server via SSRF and malicious requests. Users are recommended to upgrade to version 2.4.62 which fixes this issue. (CVE-2024-40898)

Note that Nessus has not tested for these issues but has instead relied only on the application's self-reported version number.

**See Also**

<https://httpd.apache.org/security/vulnerabilities_24.html>

**Solution**

Upgrade to Apache version 2.4.62 or later.

**Risk Factor**

High

**CVSS v3.0 Base Score**

7.5 (CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:N/A:N)

**CVSS v3.0 Temporal Score**

6.5 (CVSS:3.0/E:U/RL:O/RC:C)

**VPR Score**

3.6

**EPSS Score**

0.0044

**CVSS v2.0 Base Score**

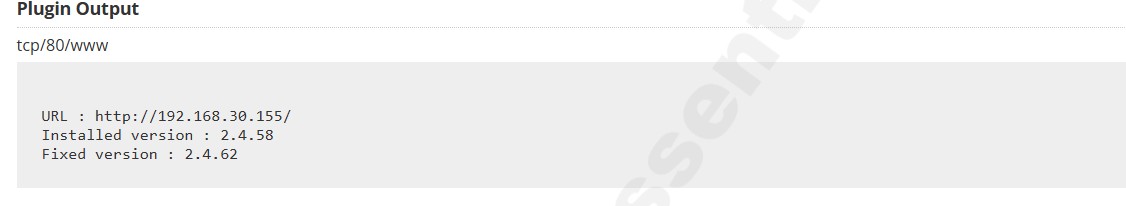
7.8 (CVSS2#AV:N/AC:L/Au:N/C:C/I:N/A:N)

**CVSS v2.0 Temporal Score**

5.8 (CVSS2#E:U/RL:OF/RC:C)

**Plugin Information**

Published: 2024/11/06, Modified: 2024/11/06



Vulnerability Level: High



**Synopsis**

The remote service is affected by a vulnerability.

**Description**

The version of OpenSSL installed on the remote host is prior to 3.1.4. It is, therefore, affected by a vulnerability as referenced in the 3.1.4 advisory.

* Issue summary: A bug has been identified in the processing of key and initialisation vector (IV) lengths.

This can lead to potential truncation or overruns during the initialisation of some symmetric ciphers.

Impact summary: A truncation in the IV can result in non-uniqueness, which could result in loss of confidentiality for some cipher modes. When calling EVP\_EncryptInit\_ex2(), EVP\_DecryptInit\_ex2() or EVP\_CipherInit\_ex2() the provided OSSL\_PARAM array is processed after the key and IV have been established. Any alterations to the key length, via the keylen parameter or the IV length, via the ivlen parameter, within the OSSL\_PARAM array will not take effect as intended, potentially causing truncation or overreading of these values. The following ciphers and cipher modes are impacted: RC2, RC4, RC5, CCM, GCM and OCB. For the CCM, GCM and OCB cipher modes, truncation of the IV can result in loss of confidentiality. For example, when following NIST's SP 800-38D section 8.2.1 guidance for constructing a deterministic IV for AES in GCM mode, truncation of the counter portion could lead to IV reuse. Both truncations and overruns of the key and overruns of the IV will produce incorrect results and could, in some cases, trigger a memory exception. However, these issues are not currently assessed as security critical. Changing the key and/or IV lengths is not considered to be a common operation and the vulnerable API was recently introduced. Furthermore it is likely that application developers will have spotted this problem during testing since decryption would fail unless both peers in the communication were similarly vulnerable. For these reasons we expect the probability of an application being vulnerable to this to be quite low. However if an application is vulnerable then this issue is considered very serious. For these reasons we have assessed this issue as Moderate severity overall. The OpenSSL SSL/TLS implementation is not affected by this issue. The OpenSSL 3.0 and 3.1 FIPS providers are not affected by this because the issue lies outside of the FIPS provider boundary. OpenSSL 3.1 and 3.0 are vulnerable to this issue. (CVE-2023-5363)

Note that Nessus has not tested for this issue but has instead relied only on the application's selfreported version number.

**See Also**

<http://www.nessus.org/u?442518e0><https://www.cve.org/CVERecord?id=CVE-2023-5363><https://www.openssl.org/news/secadv/20231024.txt><https://www.openssl.org/policies/secpolicy.html>

**Solution**

Upgrade to OpenSSL version 3.1.4 or later.

**Risk Factor**

High

**CVSS v3.0 Base Score**

7.5 (CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:N/A:N)

**CVSS v3.0 Temporal Score**

6.5 (CVSS:3.0/E:U/RL:O/RC:C)

**VPR Score**

4.4

**EPSS Score**

0.0573

**CVSS v2.0 Base Score**

7.8 (CVSS2#AV:N/AC:L/Au:N/C:C/I:N/A:N)

**CVSS v2.0 Temporal Score**

5.8 (CVSS2#E:U/RL:OF/RC:C)

**STIG Severity**

I

**Plugin Information**

Published: 2023/10/25, Modified: 2024/10/07



Vulnerability Level: High



**Synopsis**

The remote service is affected by multiple vulnerabilities.

**Description**

The version of OpenSSL installed on the remote host is prior to 3.1.6. It is, therefore, affected by multiple vulnerabilities as referenced in the 3.1.6 advisory.

* Issue summary: Checking excessively long DSA keys or parameters may be very slow. Impact summary:

Applications that use the functions EVP\_PKEY\_param\_check() or EVP\_PKEY\_public\_check() to check a DSA public key or DSA parameters may experience long delays. Where the key or parameters that are being checked have been obtained from an untrusted source this may lead to a Denial of Service. The functions EVP\_PKEY\_param\_check() or EVP\_PKEY\_public\_check() perform various checks on DSA parameters. Some of those computations take a long time if the modulus (`p` parameter) is too large. Trying to use a very large modulus is slow and OpenSSL will not allow using public keys with a modulus which is over 10,000 bits in length for signature verification. However the key and parameter check functions do not limit the modulus size when performing the checks. An application that calls EVP\_PKEY\_param\_check() or EVP\_PKEY\_public\_check() and supplies a key or parameters obtained from an untrusted source could be vulnerable to a Denial of Service attack. These functions are not called by OpenSSL itself on untrusted DSA keys so only applications that directly call these functions may be vulnerable. Also vulnerable are the OpenSSL pkey and pkeyparam command line applications when using the `-check` option. The OpenSSL SSL/TLS implementation is not affected by this issue. The OpenSSL 3.0 and 3.1 FIPS providers are affected by this issue. (CVE-2024-4603)

* Issue summary: Some non-default TLS server configurations can cause unbounded memory growth when processing TLSv1.3 sessions Impact summary: An attacker may exploit certain server configurations to trigger unbounded memory growth that would lead to a Denial of Service This problem can occur in TLSv1.3 if the non-default SSL\_OP\_NO\_TICKET option is being used (but not if early data support is also configured and the default anti-replay protection is in use). In this case, under certain conditions, the session cache can get into an incorrect state and it will fail to flush properly as it fills. The session cache will continue to grow in an unbounded manner. A malicious client could deliberately create the scenario for this failure to force a Denial of Service. It may also happen by accident in normal operation. This issue only affects TLS servers supporting TLSv1.3. It does not affect TLS clients. The FIPS modules in 3.2, 3.1 and 3.0 are not affected by this issue. OpenSSL 1.0.2 is also not affected by this issue. (CVE-2024-2511)

* Issue summary: Calling the OpenSSL API function SSL\_free\_buffers may cause memory to be accessed that was previously freed in some situations Impact summary: A use after free can have a range of potential consequences such as the corruption of valid data, crashes or execution of arbitrary code. However, only applications that directly call the SSL\_free\_buffers function are affected by this issue. Applications that do not call this function are not vulnerable. Our investigations indicate that this function is rarely used by applications. The SSL\_free\_buffers function is used to free the internal OpenSSL buffer used when processing an incoming record from the network. The call is only expected to succeed if the buffer is not currently in use. However, two scenarios have been identified where the buffer is freed even when still in use. The first scenario occurs where a record header has been received from the network and processed by OpenSSL, but the full record body has not yet arrived. In this case calling SSL\_free\_buffers will succeed even though a record has only been partially processed and the buffer is still in use. The second scenario occurs where a full record containing application data has been received and processed by OpenSSL but the application has only read part of this data. Again a call to SSL\_free\_buffers will succeed even though the buffer is still in use. While these scenarios could occur accidentally during normal operation a malicious attacker could attempt to engineer a stituation where this occurs. We are not aware of this issue being actively exploited. The FIPS modules in 3.3, 3.2, 3.1 and 3.0 are not affected by this issue. Found by William Ahern (Akamai).

Fix developed by Matt Caswell. Fix developed by Watson Ladd (Akamai). Fixed in OpenSSL

3.3.1 (Affected since 3.3.0). (CVE-2024-4741) Note that Nessus has not tested for these issues but has instead relied only on the application's self-reported version number.

**See Also**

<http://www.nessus.org/u?5ee92eab><http://www.nessus.org/u?6f15218c><http://www.nessus.org/u?f40bd907><https://www.cve.org/CVERecord?id=CVE-2024-2511><https://www.cve.org/CVERecord?id=CVE-2024-4603><https://www.cve.org/CVERecord?id=CVE-2024-4741>

**Solution**

Upgrade to OpenSSL version 3.1.6 or later.

**Risk Factor**

Medium

**CVSS v3.0 Base Score**

7.5 (CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:H)

**CVSS v3.0 Temporal Score**

6.5 (CVSS:3.0/E:U/RL:O/RC:C)

**VPR Score**

4.4

**EPSS Score**

0.0165

**CVSS v2.0 Base Score**

5.4 (CVSS2#AV:N/AC:H/Au:N/C:N/I:N/A:C)

**CVSS v2.0 Temporal Score**

4.0 (CVSS2#E:U/RL:OF/RC:C)

**STIG Severity**

I

**Plugin Information**

Published: 2024/04/08, Modified: 2024/11/14



Vulnerability Level: High



**Synopsis**

The version PHP running on the remote web server is affected by multiple vulnerabilities.

**Description**

The version of PHP installed on the remote host is prior to 8.2.24. It is, therefore, affected by multiple vulnerabilities as referenced in the Version 8.2.24 advisory.

* In PHP versions 8.1.\* before 8.1.30, 8.2.\* before 8.2.24, 8.3.\* before 8.3.12, when using a certain non- standard configurations of Windows codepages, the fixes for CVE-2024-4577 https://github.com/advisories/GHSA-vxpp-6299-mxw3 may still be bypassed and the same command injection related to Windows Best Fit codepage behavior can be achieved. This may allow a malicious user to pass options to PHP binary being run, and thus reveal the source code of scripts, run arbitrary PHP code on the server, etc. (CVE-2024-8926)

* In PHP versions 8.1.\* before 8.1.30, 8.2.\* before 8.2.24, 8.3.\* before 8.3.12, erroneous parsing of multipart form data contained in an HTTP POST request could lead to legitimate data not being processed.

This could lead to malicious attacker able to control part of the submitted data being able to exclude portion of other data, potentially leading to erroneous application behavior. (CVE-20248925)

* In PHP versions 8.1.\* before 8.1.30, 8.2.\* before 8.2.24, 8.3.\* before 8.3.12, HTTP\_REDIRECT\_STATUS variable is used to check whether or not CGI binary is being run by the HTTP server. However, in certain scenarios, the content of this variable can be controlled by the request submitter via HTTP headers, which can lead to cgi.force\_redirect option not being correctly applied. In certain configurations this may lead to arbitrary file inclusion in PHP. (CVE2024-8927)

* In PHP versions 8.1.\* before 8.1.30, 8.2.\* before 8.2.24, 8.3.\* before 8.3.12, when using PHPFPM SAPI and it is configured to catch workers output through catch\_workers\_output = yes, it may be possible to pollute the final log or remove up to 4 characters from the log messages by manipulating log message content.

Additionally, if PHP-FPM is configured to use syslog output, it may be possible to further remove log data using the same vulnerability. (CVE-2024-9026)

Note that Nessus has not tested for these issues but has instead relied only on the application's self-reported version number.

**See Also** <http://php.net/ChangeLog-8.php#8.2.24>

**Solution**

Upgrade to PHP version 8.2.24 or later.

**Risk Factor**

High

**CVSS v3.0 Base Score**

8.8 (CVSS:3.0/AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:H)

**CVSS v3.0 Temporal Score**

7.9 (CVSS:3.0/E:P/RL:O/RC:C)

**VPR Score**

6.7

**EPSS Score**

0.0217

**CVSS v2.0 Base Score**

9.0 (CVSS2#AV:N/AC:L/Au:S/C:C/I:C/A:C)

**CVSS v2.0 Temporal Score**

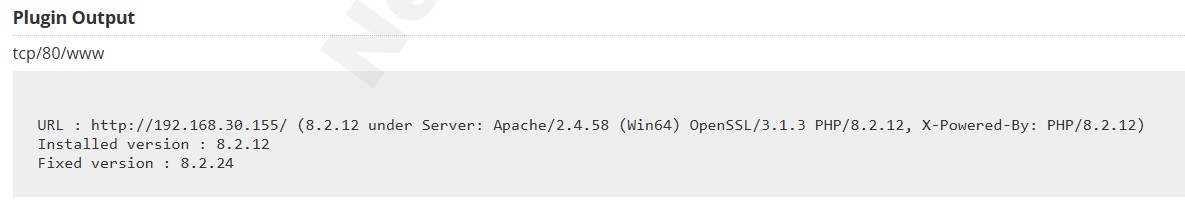
7.0 (CVSS2#E:POC/RL:OF/RC:C)

**STIG Severity**

I

**Plugin Information**

Published: 2024/09/26, Modified: 2025/04/17



Vulnerability Level: High



**Synopsis**

The version PHP running on the remote web server is affected by multiple vulnerabilities.

**Description**

The version of PHP installed on the remote host is prior to 8.2.28. It is, therefore, affected by multiple vulnerabilities as referenced in the Version 8.2.28 advisory.

Note that Nessus has not tested for these issues but has instead relied only on the application's self-reported version number.

**See Also**

<http://php.net/ChangeLog-8.php#8.2.28><https://github.com/php/php-src/security/advisories/GHSA-52jp-hrpf-2jff><https://github.com/php/php-src/security/advisories/GHSA-hgf5-96fm-v528><https://github.com/php/php-src/security/advisories/GHSA-p3x9-6h7p-cgfc><https://github.com/php/php-src/security/advisories/GHSA-pcmh-g36c-qc44><https://github.com/php/php-src/security/advisories/GHSA-v8xr-gpvj-cx9g>

**Solution**

Upgrade to PHP version 8.2.28 or later.

**Risk Factor**

Medium

**CVSS v3.0 Base Score**

8.6 (CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:C/C:N/I:N/A:H)

**CVSS v3.0 Temporal Score**

7.5 (CVSS:3.0/E:U/RL:O/RC:C)

**VPR Score**

4.2

**EPSS Score**

0.0013

**CVSS v2.0 Base Score**

4.3 (CVSS2#AV:N/AC:M/Au:N/C:N/I:N/A:P)

**CVSS v2.0 Temporal Score**

3.2(CVSS2#E:U/RL:OF/RC:C)

**STIG Severity**

I

**Plugin Information**

Published: 2025/03/13, Modified: 2025/03/21



Vulnerability Level: Medium



**Synopsis**

Debugging functions are enabled on the remote web server.

**Description**

The remote web server supports the TRACE and/or TRACK methods. TRACE and TRACK are HTTP methods that are used to debug web server connections.

**See Also**

<http://www.nessus.org/u?e979b5cb><http://www.apacheweek.com/issues/03-01-24><https://download.oracle.com/sunalerts/1000718.1.html>

**Solution**

Disable these HTTP methods. Refer to the plugin output for more information.

**Risk Factor**

Medium

**CVSS v3.0 Base Score**

5.3 (CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N)

**CVSS v3.0 Temporal Score**

4.6 (CVSS:3.0/E:U/RL:O/RC:C)

**VPR Score**

4.0

**EPSS Score**

0.8269

**CVSS v2.0 Base Score**

5.0 (CVSS2#AV:N/AC:L/Au:N/C:P/I:N/A:N)

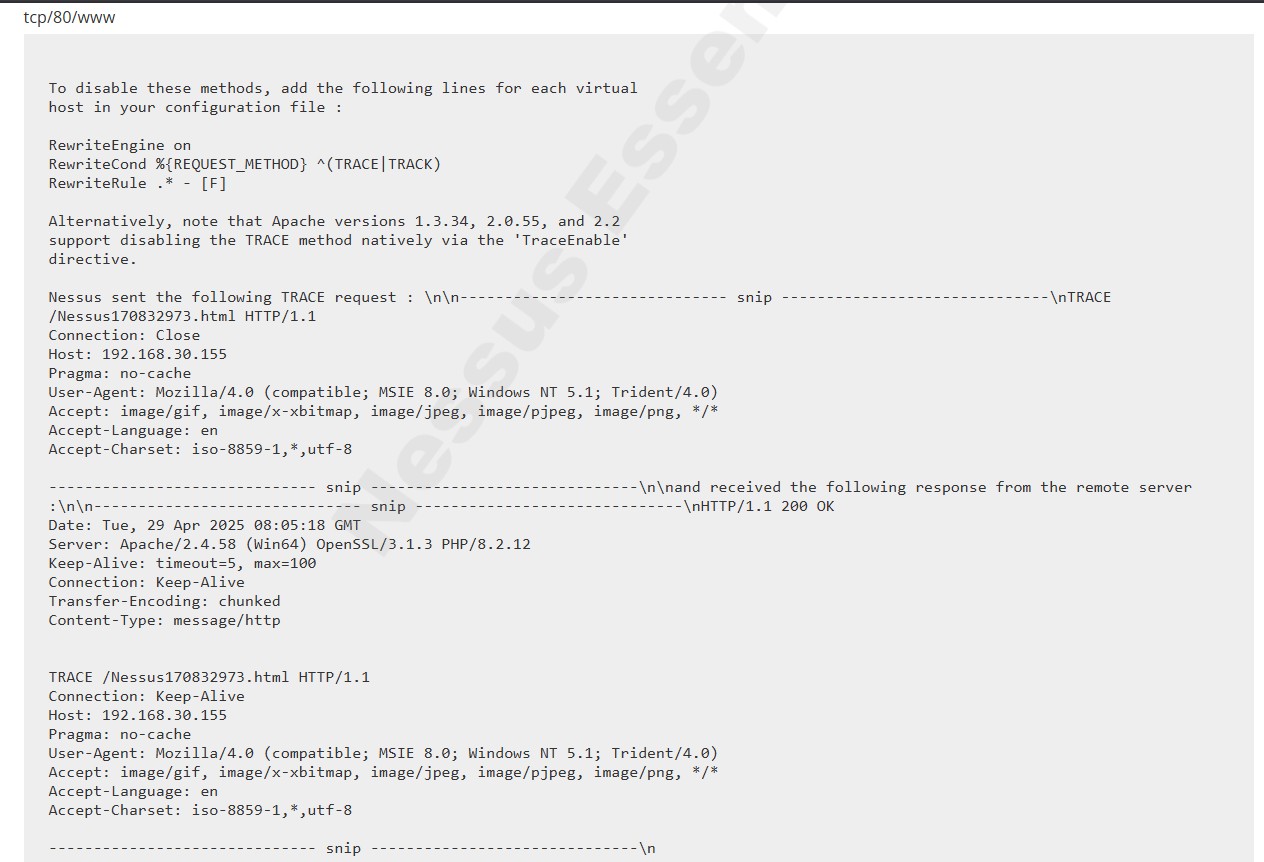
**CVSS v2.0 Temporal Score**

3.7 (CVSS2#E:U/RL:OF/RC:C)

**Plugin Information**

Published: 2003/01/23, Modified: 2024/04/09

**Plugin Output**



Vulnerability Level: Medium



**Synopsis**

The remote service is affected by multiple vulnerabilities.

**Description**

The version of OpenSSL installed on the remote host is prior to 3.1.5. It is, therefore, affected by multiple vulnerabilities as referenced in the 3.1.5 advisory.

- Issue summary: Processing a maliciously formatted PKCS12 file may lead OpenSSL to crash leading to a potential Denial of Service attack Impact summary: Applications loading files in the PKCS12 format from untrusted sources might terminate abruptly. A file in PKCS12 format can contain certificates and keys and may come from an untrusted source. The PKCS12 specification allows certain fields to be NULL, but OpenSSL does not correctly check for this case. This can lead to a NULL pointer dereference those results in OpenSSL crashing. If an application processes PKCS12 files from an untrusted source using the OpenSSL APIs then that application will be vulnerable to this issue. OpenSSL APIs that are vulnerable to this are:

PKCS12\_parse(), PKCS12\_unpack\_p7data(), PKCS12\_unpack\_p7encdata(),

PKCS12\_unpack\_authsafes() and PKCS12\_newpass(). We have also fixed a similar issue in SMIME\_write\_PKCS7(). However, since this function is related to writing data we do not consider it security significant. The FIPS modules in 3.2, 3.1 and 3.0 are not affected by this issue. (CVE-2024-0727) **See Also**

<http://www.nessus.org/u?0a42ec4e><http://www.nessus.org/u?950a9188><http://www.nessus.org/u?aca829a1><http://www.nessus.org/u?d086a7ea><https://www.cve.org/CVERecord?id=CVE-2023-5678><https://www.cve.org/CVERecord?id=CVE-2023-6129><https://www.cve.org/CVERecord?id=CVE-2023-6237><https://www.cve.org/CVERecord?id=CVE-2024-0727>

**Solution**

Upgrade to OpenSSL version 3.1.5 or later.

**Risk Factor**

Medium

**CVSS v3.0 Base Score**

6.5 (CVSS:3.0/AV:N/AC:H/PR:N/UI:N/S:U/C:N/I:L/A:H)

**CVSS v3.0 Temporal Score**

5.7 (CVSS:3.0/E:U/RL:O/RC:C)

**VPR Score**

5.0

**EPSS Score**

0.0274

**CVSS v2.0 Base Score**

6.1 (CVSS2#AV:N/AC:H/Au:N/C:N/I:P/A:C)

**CVSS v2.0 Temporal Score**

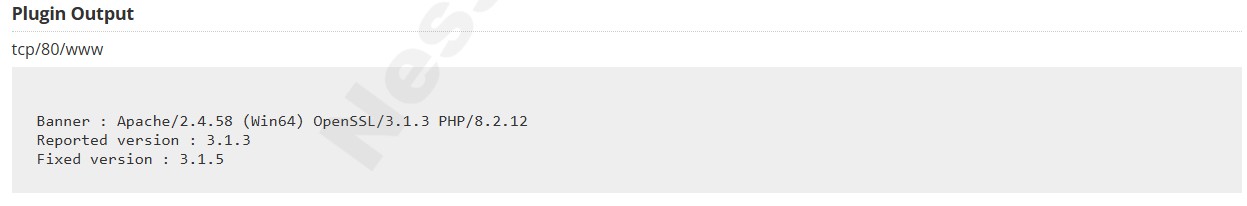
4.4 (CVSS2#E:U/RL:OF/RC:C)

**STIG Severity**

I

**Plugin Information**

Published: 2023/11/07, Modified: 2024/10/07



Vulnerability Level: Medium



**Synopsis**

The remote service is affected by a vulnerability.

**Description**

The version of OpenSSL installed on the remote host is prior to 3.1.8. It is, therefore, affected by a vulnerability as referenced in the 3.1.8 advisory.

- Issue summary: Use of the low-level GF(2^m) elliptic curve APIs with untrusted explicit values for the field polynomial can lead to out-of-bounds memory reads or writes. Impact summary: Out of bound memory writes can lead to an application crash or even a possibility of a remote code execution, however, in all the protocols involving Elliptic Curve Cryptography that we're aware of, either only named curves are supported, or, if explicit curve parameters are supported, they specify an X9.62 encoding of binary (GF(2^m)) curves that can't represent problematic input values. Thus the likelihood of existence of a vulnerable application is low. In particular, the X9.62 encoding is used for ECC keys in X.509 certificates, so problematic inputs cannot occur in the context of processing X.509 certificates. Any problematic use-cases would have to be using an exotic curve encoding. The affected APIs include:

EC\_GROUP\_new\_curve\_GF2m(), EC\_GROUP\_new\_from\_params(), and various supporting

BN\_GF2m\_\*() functions.

Applications working with exotic explicit binary (GF(2^m)) curve parameters, that make it possible to represent invalid field polynomials with a zero constant term, via the above or similar APIs, may terminate abruptly as a result of reading or writing outside of array bounds. Remote code execution cannot easily be ruled out. The FIPS modules in 3.3, 3.2, 3.1 and 3.0 are not affected by this issue.

(CVE-2024-9143)

Note that Nessus has not tested for this issue but has instead relied only on the application's selfreported version number.

**See Also**

<http://www.nessus.org/u?5f636435><https://openssl-library.org/news/secadv/20241016.txt><https://openssl-library.org/policies/general/security-policy/#low>

[https://www.cve.org/CVERecord?id=CV](https://www.cve.org/CVERecord?id=CVE-2024-9143)  [E-2024-9143](https://www.cve.org/CVERecord?id=CVE-2024-9143)

**Solution**

Upgrade to OpenSSL version 3.1.8 or later.

**Risk Factor**

Medium

**CVSS v3.0 Base Score**

4.3 (CVSS:3.0/AV:N/AC:L/PR:L/UI:N/S:U/C:N/I:L/A:N)

**CVSS v3.0 Temporal Score**

3.8(CVSS:3.0/E:U/RL:O/RC:C)

**VPR Score**

2.2

**EPSS Score**

0.0036

**CVSS v2.0 Base Score**

6.8 (CVSS2#AV:N/AC:M/Au:N/C:P/I:P/A:P)

**CVSS v2.0 Temporal Score**

5.0 (CVSS2#E:U/RL:OF/RC:C)

**STIG Severity**

I

**Plugin Information**

Published: 2024/10/16, Modified: 2025/02/21



Vulnerability Level: Medium



**Synopsis**

The version PHP running on the remote web server is affected by multiple vulnerabilities.

**Description**

The version of PHP installed on the remote host is prior to 8.2.18. It is, therefore, affected by multiple vulnerabilities as referenced in the Version 8.2.18 advisory.

* In PHP versions before 7.4.31, 8.0.24 and 8.1.11, the vulnerability enables network and samesite attackers to set a standard insecure cookie in the victim's browser which is treated as a `\_\_Host-` or `\_\_Secure-` cookie by PHP applications. (CVE-2022-31629)

Note that Nessus has not tested for these issues but has instead relied only on the application's self-reported version number.

**See Also** <http://php.net/ChangeLog-8.php#8.2.18>

**Solution**

Upgrade to PHP version 8.2.18 or later.

**Risk Factor**

High

**CVSS v3.0 Base Score**

6.5 (CVSS:3.0/AV:N/AC:L/PR:N/UI:R/S:U/C:N/I:H/A:N)

**CVSS v3.0 Temporal Score**

5.9 (CVSS:3.0/E:P/RL:O/RC:C)

**VPR Score**

6.3

**EPSS Score**

0.4225

**CVSS v2.0 Base Score**

7.8 (CVSS2#AV:N/AC:L/Au:N/C:N/I:C/A:N)

**CVSS v2.0 Temporal Score**

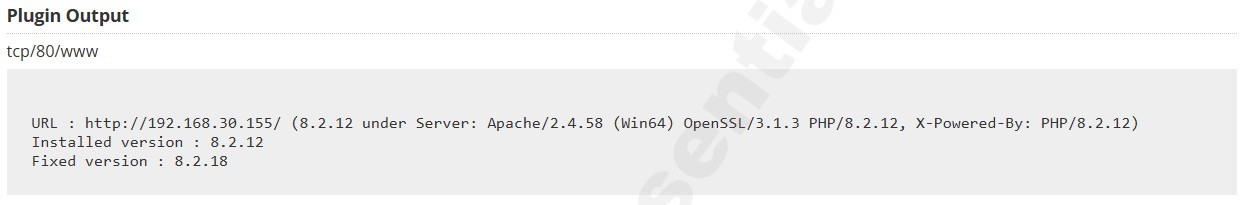
6.1 (CVSS2#E:POC/RL:OF/RC:C)

**STIG Severity**

I

**Plugin Information**

Published: 2024/04/11, Modified: 2024/11/22



Vulnerability Level: Medium



**Synopsis**

The SSL certificate for this service cannot be trusted.

**Description**

The server's X.509 certificate cannot be trusted. This situation can occur in three different ways, in which the chain of trust can be broken, as stated below :

* First, the top of the certificate chain sent by the server might not be descended from a known public certificate authority. This can occur either when the top of the chain is an unrecognized, self-signed certificate, or when intermediate certificates are missing that would connect the top of the certificate chain to a known public certificate authority.

* Second, the certificate chain may contain a certificate that is not valid at the time of the scan. This can occur either when the scan occurs before one of the certificate's 'notBefore' dates, or after one of the certificate's 'notAfter' dates.

* Third, the certificate chain may contain a signature that either didn't match the certificate's information or could not be verified. Bad signatures can be fixed by getting the certificate with the bad signature to be re-signed by its issuer. Signatures that could not be verified are the result of the certificate's issuer using a signing algorithm that Nessus either does not support or does not recognize.

If the remote host is a public host in production, any break in the chain makes it more difficult for users to verify the authenticity and identity of the web server. This could make it easier to carry out man-in-the-middle attacks against the remote host.

**See Also**

<https://www.itu.int/rec/T-REC-X.509/en>

[https://en.wikipedia.org/wiki/X.50](https://en.wikipedia.org/wiki/X.509)  [9](https://en.wikipedia.org/wiki/X.509)

**Solution**

Purchase or generate a proper SSL certificate for this service.

**Risk Factor** Medium **CVSS v3.0 Base Score**

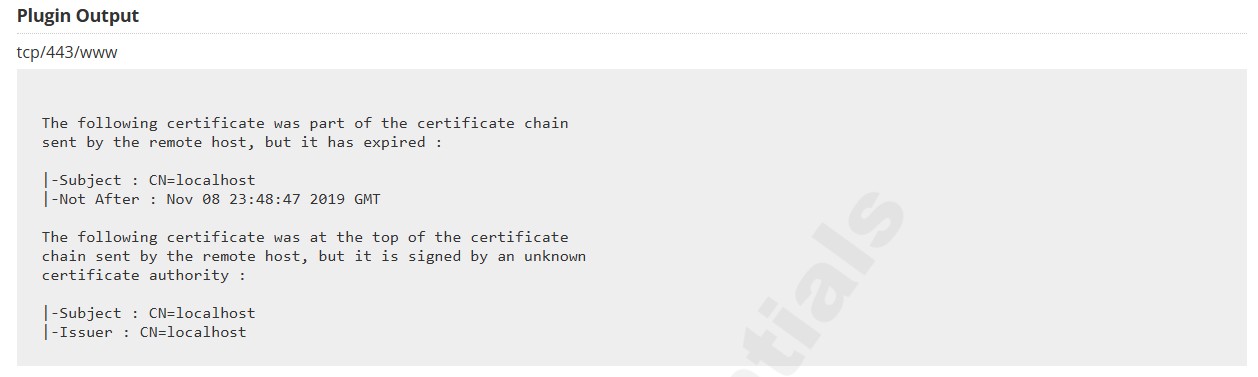
6.5 (CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:L/A:N)

**CVSS v2.0 Base Score**

6.4 (CVSS2#AV:N/AC:L/Au:N/C:P/I:P/A:N)

**Plugin Information**

Published: 2010/12/15, Modified: 2020/04/27



**CONCLUSION**

The project successfully demonstrated the practical implementation of Web Application Penetration Testing (WAPT) and Network Auditing using industry-standard tools and methodologies. Through systematic assessment, multiple security weaknesses were identified, ranging from misconfigurations and outdated services to web-based vulnerabilities that could potentially be exploited by attackers.

By leveraging tools such as Nessus for network vulnerability assessment and Burp Suite and OWASP ZAP for web application testing, the project highlighted the importance of proactive security measures in identifying and mitigating risks before they can be exploited. The findings emphasize the need for regular vulnerability assessments, timely patch management, and secure coding practices as part of a robust cybersecurity posture.

Overall, this project provided valuable hands-on experience in ethical hacking, vulnerability management, and secure infrastructure design, aligning with real-world cybersecurity practices. It also reinforced the critical role of continuous monitoring and assessment in maintaining the confidentiality, integrity, and availability of digital assets.

#### Limitation

1. **Limited Audit Scope**:

The assessment was restricted to selected web applications and network assets due to time and resource constraints.

1. **Time-Bound Execution**:

A limited timeframe prevented exhaustive testing, especially for low-severity and informational vulnerabilities.

1. **Tool Dependency**:

Reliance on open-source tools may have missed deeper or more complex vulnerabilities not detectable through automated scanning.

1. **Lack of Real-Time Traffic**:

The audit environment lacked production-level traffic, which limits the realism of attack simulations.

1. **No Zero-Day Analysis**:

Ethical and legal boundaries excluded the testing of unknown or zero-day vulnerabilities.

#### Future Works

* **Broader Scope Coverage**:

Future assessments can include APIs, mobile applications, and cloud-based infrastructures for a more comprehensive evaluation.

* **DevSecOps Integration**:

Embedding security testing into the software development lifecycle can help identify vulnerabilities early.

* **Continuous Monitoring**:

Implementing real-time monitoring and alerting mechanisms would enhance threat detection capabilities.

* **Advanced Threat Simulation**:

Including red teaming and post-exploitation activities can provide deeper insights into real-world attack resilience.

* **Security Awareness Programs**:

Conducting regular training for developers and system administrators can reduce human error and improve security hygiene.

#### References and Bibliography

i. OWASP Top Ten: <https://owasp.org/www-project-top-ten/> ii. SANS Top 25 Software Errors: [https://www.sans.org/top25-software-](https://www.sans.org/top25-software-errors/)

[errors/](https://www.sans.org/top25-software-errors/)

iii. National Vulnerability Database: <https://nvd.nist.gov/vuln-metrics/cvss>iv. Synk Vulnerability Database: <https://security.snyk.io/>

v. Common Weakness Enumeration: <https://cwe.mitre.org/> vi. Common Vulnerabilities and Exposures <https://cve.mitre.org/> vii. Advanced IP Scanner – User Guide and Documentation. Retrieved from <https://www.advanced-ip-scanner.com/help/>

1. Nmap Network Scanning: The Official Nmap Project Guide to Network Discovery and Security Scanning. Insecure.Com LLC.

<https://nmap.org/book/>

1. Zeltser, L. (2018). Introduction to Network Auditing and Vulnerability Assessment. SANS Institute Reading Room. Retrieved from <https://www.sans.org/reading-room>
2. Tenable, Inc. (2024). Nessus User Guide. Retrieved from <https://docs.tenable.com/nessus>