Alden A. Quiñones

IT Protection and Security: MIT 264

Security Audit of PNexus Web Application and its Deployment Environment.

Table of Contents

[PHASE 2: DATA COLLECTION & ANALYSIS 2](#_Toc195355634)

[1. System/Network Reconnaissance 2](#_Toc195355635)

[2. Use of Security Tools for Data Gathering 2](#_Toc195355636)

[A. Application-Level Vulnerabilities 2](#_Toc195355637)

[B. Deployment Environment Vulnerabilities 3](#_Toc195355638)

[3. Risk and Vulnerability Analysis 3](#_Toc195355639)

[A. Infrastructure 3](#_Toc195355640)

[The Department of Social Welfare and Development (DSWD) maintains a comprehensive and secure ICT infrastructure designed to support its nationwide operations. 3](#_Toc195355641)

[1. Nationwide ICT Infrastructure: ENDDSWD 3](#_Toc195355642)

[2. Regional Data Center (FO12. ENTDSWD) 3](#_Toc195355643)

[3. FO12 Internal Infrastructure 3](#_Toc195355644)

[4. Device Configuration and Security 4](#_Toc195355645)

[5. Field-Use Devices 4](#_Toc195355646)

[6. Personal Devices (DSWD Personnel) 4](#_Toc195355647)

[7. Guest Devices (Visitors, Clients, etc.) 4](#_Toc195355648)

[8. Backup and Disaster Recovery Policy 5](#_Toc195355649)

[9. Power backup 5](#_Toc195355650)

[B. Deployment Environment 5](#_Toc195355651)

[C. Application-Level Analysis 9](#_Toc195355652)

[4. Comparison with Industry Security Standards 13](#_Toc195355653)

[A. Infrastructure Level 13](#_Toc195355654)

[B. Application Level 14](#_Toc195355655)

# PHASE 2: DATA COLLECTION & ANALYSIS

1. System/Network Reconnaissance

Given the highly secure network setup in the PNexus Web Application environment, a full network reconnaissance was not applicable due to security policy of the Cyber Security Unit of ICTMS. However, a local application-level reconnaissance scan was conducted using ZAP to assess the security posture of the system. The scan targeted the application on http://127.0.0.1, simulating real-world attack scenarios and attack surfaces that could be exposed through the web application. This approach ensured a comprehensive identification of web-based vulnerabilities, despite the limited scope of the assessment at the network level.

The local reconnaissance scan focused on evaluating potential security risks associated with the application infrastructure. While the network itself was secured via the GlobalProtect VPN, which restricts access to authenticated users and adds a layer of security, it was still important to analyze the web application for vulnerabilities that could be exploited even when protected by the VPN.

The PNexus Web Application, developed using CodeIgniter 3 on the back-end, was assessed for potential exposure to SQL injection vulnerabilities, due to the lack of built-in ORM support. Additionally, the front-end components, including jQuery, HTML5, and Bootstrap 4, were evaluated for common vulnerabilities like cross-site scripting (XSS) and cross-site request forgery (CSRF), especially since the application lacks proper configuration to mitigate these risks.

On the security mechanisms such as CAPTCHA and Two-Factor Authentication (2FA) were not implemented, which significantly increases the risk of bot attacks and account compromises. Although the system integrates with Active Directory for user authentication, the integration with the web application was also assessed for potential weaknesses. Given that the application operates over HTTP, an insecure protocol, this raised concerns about the integrity and confidentiality of the data transmitted.

The scan results indicated that while the network setup itself provided strong protection, the application’s vulnerabilities, including lack of encryption, insecure authentication mechanisms, and missing security headers, were clear areas for immediate improvement.

1. Use of Security Tools for Data Gathering
2. Application-Level Vulnerabilities

A security assessment was conducted using OWASP ZAP (Zed Attack Proxy) v2.16.1 as the primary tool for data gathering. ZAP was configured to perform both passive and active scans on the application running in an isolated server environment. The scanning process was designed to identify potential vulnerabilities across various endpoints of the application. This comprehensive scan included a detailed analysis of HTTP headers, JavaScript libraries, form handling mechanisms, hidden files, session and cookie management, and response metadata. The scan specifically targeted key application endpoints, such as login forms, APIs, and session management features. Several vulnerabilities were detected during the scan, and they were subsequently categorized based on their severity level.

1. Deployment Environment Vulnerabilities

Wireshark was used to analyze the network traffic in the deployment environment. It helped identify vulnerabilities such as sensitive data being transmitted without encryption that could be intercepted. It also detected the use of outdated protocols like old SSL/TLS versions, which are insecure.

NMAP (Network Mapper) was employed to conduct a thorough network discovery and vulnerability assessment of the deployment environment. NMAP was used to map out open ports, services, and active hosts within the network. By performing a variety of scans, including TCP Connect, SYN Scan, and OS detection, NMAP provided valuable insights into the network's exposed attack surface.

1. Risk and Vulnerability Analysis
2. Infrastructure

The Department of Social Welfare and Development (DSWD) maintains a comprehensive and secure ICT infrastructure designed to support its nationwide operations.

### 1. Nationwide ICT Infrastructure: ENDDSWD

* The DSWD utilizes a nationwide **IP-VPN infrastructure** called **ENDDSWD**.
* The network is segmented into **15 regional intranets**, each represented by local data centers (e.g., **FO1.ENTDSWD** to **FO12.ENTDSWD**).

### 2. Regional Data Center (FO12. **ENTDSWD**)

* The **Field Office 12 Data Center (FO12.ENTDSWD)** is located within the **Information and Communications Technology Management Service (ICTMS)**.
* Only one authorized personnel allowed to go inside the enclosed datacenter.

### 3. FO12 Internal Infrastructure

#### a. Intermediary Data Frames (IDF)

* Connected to the **Main Distribution Frame (MDF)** at the Central Office.
* Each IDF includes the following components:
  + Modular Gateway Router
  + Palo Alto Ingress Firewall
  + Palo Alto Intrusion Detection System (IDS)
  + PA-Series Next-Generation Firewall (NGFW)
  + Distribution Router with two connected switches
  + Patch panels for structured distribution

#### b. Office-Specific Network Cabinets

* Each office has a dedicated cabinet containing:
  + Repeater
  + Router
  + Switches
  + Patch Panels

#### c. Device Cabling

* All computer units are connected to the network via **Cat6e cables** routed through patch panels

### 4. Device Configuration and Security

#### All devices issued by DSWD are:

* Registered to the Domain Controller (DC) located at the Central Office.
* User profiles are synchronized with a centralized file server (NAS).
* Protected by Cortex XDR, enabling real-time monitoring, threat prevention, and reporting.
* Restricted from unauthorized software installation; any attempt is automatically blocked and logged by XDR.
* Software installation requests are handled by ICTMS through remote deployment.

### 5. Field-Use Devices

#### Devices used outside DSWD premises by field staff:

* Configured identically to internal devices.
* Access DSWD resources (e.g., web applications, shared drives) via GlobalProtect VPN only.

### 6. Personal Devices (DSWD Personnel)

#### Allowed to connect to the internet via a separate private network.

#### Access to DSWD internal resources is strictly restricted.

### 7. Guest Devices (Visitors, Clients, etc.)

* Can connect to public Wi-Fi provided by DSWD.
* No access to internal DSWD systems and resources.

### 8. Backup and Disaster Recovery Policy

#### a. Database Backups

* **Full** backups scheduled on the first day of every month at 6pm.
* Daily differential backups every 9am.
* All backups stored on a dedicated **Network Attached Storage (NAS)**.

#### b. File Backups

* Devices registered under **Active Directory** automatically back up files to another NAS.

#### c. Offsite Redundancy

* **All NAS** devices **are mirrored to an offsite NAS** located in **General Santos City** to ensure disaster recovery and data availability.

### 9. Power backup

*In case of power failure, the data center has the following power backup feature:*

* Heavy Duty UPS
* Automatic transfer switches
* Backed with industrial heavy duty generator

Risk Assessment

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Potential Risk** | **Risk Level** | **Impact** | **Recommendation** |
| 1 | Absence of VPNs on very division/section | Low | If there a breach on one of the division/office, the damage could spread in the entire network. | Isolate the risk by clustering offices in separate VPN. |
| 2 | Absence of reverse proxy device. | Low | If global protect account of one of the field staffs exploited, the hacker could exploit also the Ingres Firewall. | For extra layer of security, there should be an additional router that will serve as revise proxy that would prevent hacker from obtaining the real ip-vpn address of the ingres-router. |

1. Deployment Environment

The deployment environment of the PNexus Web Application was also assessed to identify potential infrastructure-level risks. This evaluation focused on configuration weaknesses, absence of recommended production-grade practices, and the overall security posture of the application hosting environment.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **#** | **Vulnerability** | **Description** | **Risk Level** | **Impact** | **Recommendation** |
| 1 | CVE-2024-38476 (Apache HTTPD) | Apache HTTPD 2.4.58 vulnerable to remote code execution and other attacks. | Critical | Potential for remote code execution and service disruption if exploited. | Update Apache HTTPD to the latest stable version. Apply available security patches. |
| 2 | CVE-2024-38474 (Apache HTTPD) | Apache HTTPD 2.4.58 vulnerable to denial-of-service (DoS) attacks. | High | Could lead to service outages and disruption in server availability. | Update Apache HTTPD to the latest stable version. Ensure that DoS protections are implemented. |
| 3 | Slowloris DoS Vulnerability | Apache HTTPD vulnerable to Slowloris Denial of Service attack. | High | Can cause a server crash or service denial by exhausting server resources. | Configure the server to mitigate Slowloris attacks (e.g., increase connection timeouts, use a WAF). |
| 4 | CVE-2024-38475 (Apache HTTPD) | Apache HTTPD 2.4.58 vulnerable to privilege escalation. | High | Attackers may escalate privileges and gain unauthorized access to the server. | Update Apache HTTPD to the latest stable version. |
| 5 | CVE-2024-39573 (Apache HTTPD) | Apache HTTPD vulnerable to multiple exploits affecting version 2.4.58. | High | May lead to data leakage, unauthorized access, and potential privilege escalation. | Apply all patches available for Apache HTTPD and review security configurations. |
| 6 | CVE-2024-40898 (Apache HTTPD) | Vulnerability allowing DoS or remote code execution in Apache HTTPD. | High | Server could be taken offline or used to perform malicious activities. | Update Apache HTTPD and verify any related security configurations. |
| 7 | CVE-2024-27316 (Apache HTTPD) | Apache HTTPD vulnerability related to improper input validation. | High | Exploitation could lead to application-level vulnerabilities such as arbitrary code execution. | Update Apache HTTPD and implement strong input validation mechanisms. |
| 8 | CVE-2007-6750 (Slowloris DOS attack) | Potential for Slowloris DoS attack, which can exhaust HTTP server resources by keeping connections open. | Medium | Affects server availability and may lead to service denial. | Implement mitigation techniques like rate-limiting, connection timeouts, or using a WAF to block slow requests. |
| 9 | CVE-2017-1001000 (Grafana) | Unauthenticated access vulnerability in Grafana (if accessible publicly). | High | Unauthorized users may gain access to sensitive data or administrative functions. | Ensure Grafana is properly secured behind authentication and only exposed to trusted networks. |
| 10 | CVE-2010-0738 (JMX Console) | Authentication vulnerability in the JMX console, allowing unauthorized access. | High | Unauthorized users may gain access to management functions of the Java server. | Secure JMX console access by enforcing strong authentication and limiting exposure. |
| 11 | PostgreSQL Default Version (9.6.0 or later) | Potential vulnerabilities in older PostgreSQL versions. | Medium | Potential for database compromise if vulnerabilities exist in the version being used. | Ensure that PostgreSQL is updated to the latest stable version. Implement strong database access control. |
| 12 | MSRPC (Port 135) | Microsoft RPC service exposed, vulnerable to remote exploits. | Critical | Can allow attackers to exploit known vulnerabilities and gain unauthorized access to systems. | Disable MSRPC if not required. If necessary, restrict access to trusted IPs and patch vulnerabilities. |
| 13 | RDP (Port 3389) | Remote Desktop Protocol exposed, vulnerable to brute force and other attacks. | Critical | Attackers can exploit RDP for unauthorized access to the host machine. | Disable RDP if not needed. Use strong passwords and multi-factor authentication if RDP must be enabled. |
| 14 | Microsoft DS (Port 445) | SMBv1 vulnerabilities on open Microsoft DS port, potentially exposing the system to WannaCry-style exploits. | Critical | Remote code execution and full system compromise can occur if SMB vulnerabilities are exploited. | Disable SMBv1, restrict access to port 445, and apply security patches to mitigate SMB vulnerabilities. |

The table helps prioritize remediation efforts, starting with the most critical vulnerabilities (such as RDP and SMB issues) that require urgent attention to prevent potential exploitation. It provides a structured way of assessing and addressing the security weaknesses found in the system during the scan.

Figure 5: Screenshot showing unencrypted packet of user credential.

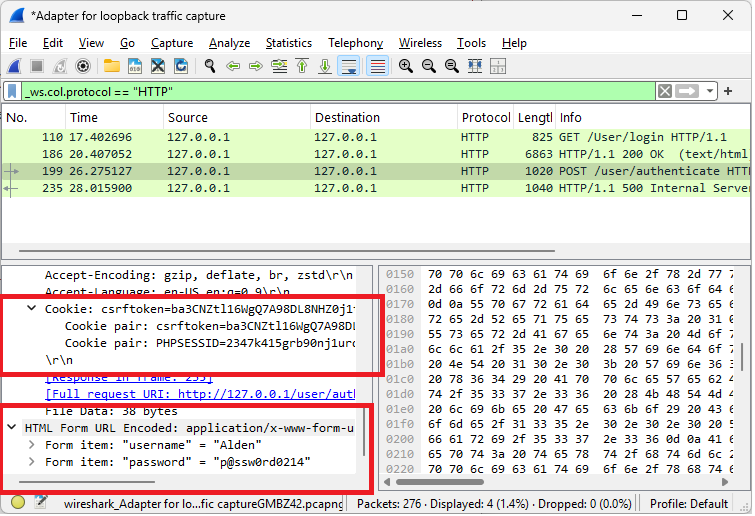
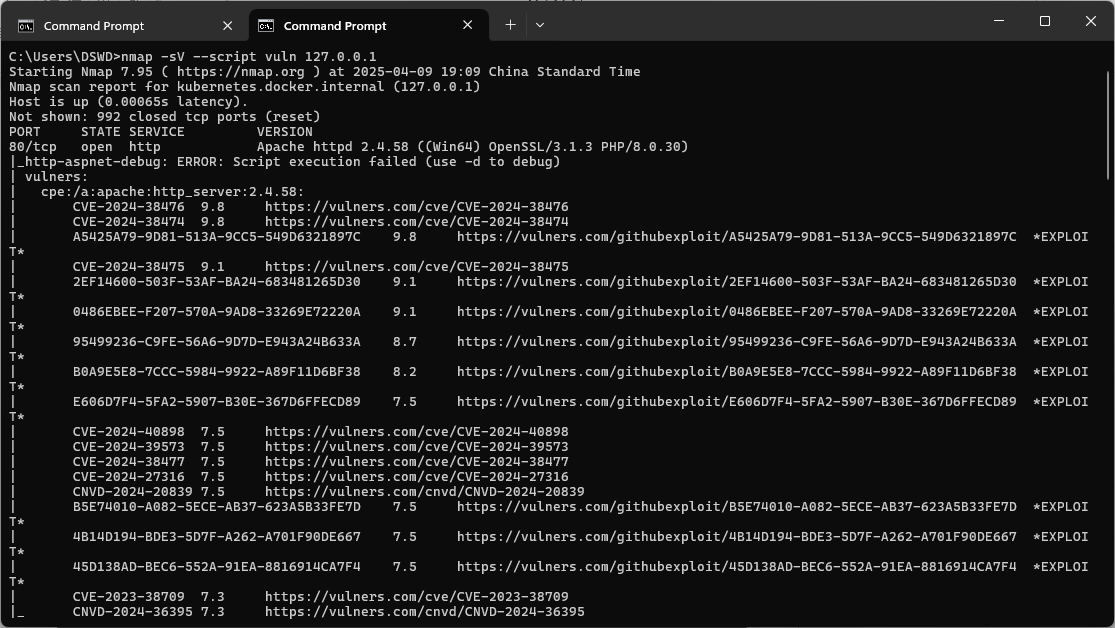


Figure 6: Screenshot of server vulnerability scan using NMAP



1. Application-Level Analysis

Figure 1: the number of alerts for each level of risk and confidence

|  | | **Confidence** | | | |
| --- | --- | --- | --- | --- | --- |
| High | Medium | Low | Total |
| **Risk** | High | - | 1 (6.2%) | - | 1 (6.2%) |
| Medium | 1 (6.2%) | 2 (12.5%) | 2 (12.5%) | 5 (31.2%) |
| Low | 1 (6.2%) | 5 (31.2%) | 1 (6.2%) | 7 (43.8%) |
| Informational | - | 3 (18.8%) | - | 3 (18.8%) |
| Total | 2 (12.5%) | 11 (68.8%) | 3 (18.8%) | 16 (100%) |

The table provides a distribution of alerts categorized by Risk and Confidence levels. It shows the number of alerts for each combination of these two categories and represents the proportion of each combination in relation to the total alerts. The data helps to analyze where the system has uncertainty in its risk assessments and how confident it is in identifying those risks.

A key observation is that Medium Confidence is prevalent across all risk categories. Of the total 16 alerts, 11 (68.8%) are classified as Medium Confidence. This means that the system is uncertain about the risk levels of most alerts, particularly for Low Risk and Medium Risk categories. This could indicate that the alerts in these categories are harder to classify with high certainty, which may point to a need for further refinement in the risk-assessment algorithms or additional data for better accuracy.

The High Risk category, with only 1 alert (6.2%) in the entire dataset. Interestingly, there are no High Confidence or User Confirmed High Risk alerts, further highlighting the uncertainty around high-risk alerts. This suggests that either high-risk scenarios are less frequent, or the system struggles to classify high-risk situations with confidence, making them harder to verify.

Both the Medium Risk and Low Risk categories have similar distributions. While the Medium Risk category has a mix of Low Confidence (12.5%) and User Confirmed (12.5%) alerts, the Low Risk category contains a larger portion of Low Confidence alerts (31.2%). Despite this, Low Risk still represents the largest number of alerts in the dataset (7 alerts or 43.8%). This indicates that the system frequently identifies low-risk situations but often with low certainty, potentially due to the nature of the data or the complexity of the risk classification process.

Finally, Informational alerts make up 18.8% of the total, with all three of them categorized under Low Confidence. Informational alerts typically don’t indicate an immediate risk but might provide valuable context or background information. The low confidence in these alerts suggests that they may be harder to categorize or verify, yet they still represent a meaningful portion of the overall alert system.

In summary, the table shows that Low Confidence is a dominant factor across the alert categories, particularly in Low Risk and Medium Risk situations. This shows a potential area for improvement in the alert system’s confidence level. It might be beneficial to further assess and refine the risk classification algorithms to increase the system's certainty in identifying and verifying risks. Additionally, investigating the cause of low-confidence alerts, especially for Low Risk items, could help improve the system’s overall performance.

*Note: The percentages in brackets represent the count as a percentage of the total number of alerts included in the report, rounded to one decimal place.*

Figure 2: ZAP reported a total of 16 security alerts, broken down as follows:

The security scan of the PNexus Web Application, conducted using ZAP, identified a total of 16 vulnerabilities, which span various levels of severity and confidence. These vulnerabilities are categorized into high, medium, low, and informational risks, as outlined below:

Figure 3: Vulnerabilities classified as high, medium, low, or informational risks.

|  |  |  |  |
| --- | --- | --- | --- |
| Risk Level | Count | Description | Recommendation |
| High | 1 | - Vulnerable JavaScript Library Detected (File: moment.min.js)  - Risk: Contains known security issues that may be exploited by attackers.  - Info: The use of an outdated version of the moment.js library exposes the application to known security vulnerabilities that could potentially be exploited by attackers. | It is recommended to replace this library with the latest, secure version to mitigate the risk. |
| Medium | 5 | * Missing Content Security Policy (CSP) Header: A missing CSP header can allow attackers to inject malicious content into the application. Implementing a CSP header will mitigate the risk of XSS attacks by restricting the sources from which content can be loaded. * Missing Anti-clickjacking Header (X-Frame-Options): The absence of the X-Frame-Options header leaves the application vulnerable to clickjacking attacks, where an attacker could trick users into clicking on hidden or invisible buttons. Setting this header to "DENY" or "SAMEORIGIN" will prevent such attacks. * Missing Anti-CSRF Tokens: The lack of anti-CSRF tokens in the application increases the risk of cross-site request forgery (CSRF) attacks, where an attacker can trick authenticated users into making unwanted requests. Implementing anti-CSRF tokens will protect against these types of attacks. * Hidden File Detected (.hg): The detection of a hidden file (.hg) suggests that version control data might be exposed to unauthorized access. It is essential to remove any unnecessary or sensitive files from the public directory. * Outdated JavaScript Library (bootstrap.bundle.min.js): An outdated version of Bootstrap exposes the application to potential vulnerabilities. Updating to the latest version of the library will ensure that known security issues are addressed | Implement the missing security headers and anti-CSRF tokens. Investigate and address the hidden file and outdated library. |
| Low | 7 | * The application is disclosing potentially sensitive information, such as server version numbers, in HTTP headers and timestamps. Removing this information will make it harder for attackers to identify the underlying server technology. * Debugging or error messages exposed in the application responses could provide attackers with information about the system that can be exploited. These messages should be suppressed in the production environment. * Cross-domain script inclusion vulnerabilities could allow an attacker to inject malicious scripts into the application. Review and secure the implementation to prevent cross-origin resource sharing (CORS) issues. | Review and remove sensitive information in headers and responses. Investigate and address the cross-domain script inclusion. |
| Informational | 3 | * The source code contains suspicious comments that could provide useful information to potential attackers. These comments should be removed or sanitized to prevent leakage of sensitive information. * The application uses modern web technologies, which may require periodic review to ensure they adhere to current security best practices. Although this finding is informational, ongoing assessment of emerging threats is recommended. * The application's session management behavior was identified, which could potentially be optimized to ensure secure handling of user sessions. No immediate action is required unless further vulnerabilities in session management are discovered. | Review and remove any sensitive or unnecessary comments. (No specific action needed for the other informational findings). |

Figure 6: Screenshot showing automated vulnerability scan using Owasp Zap

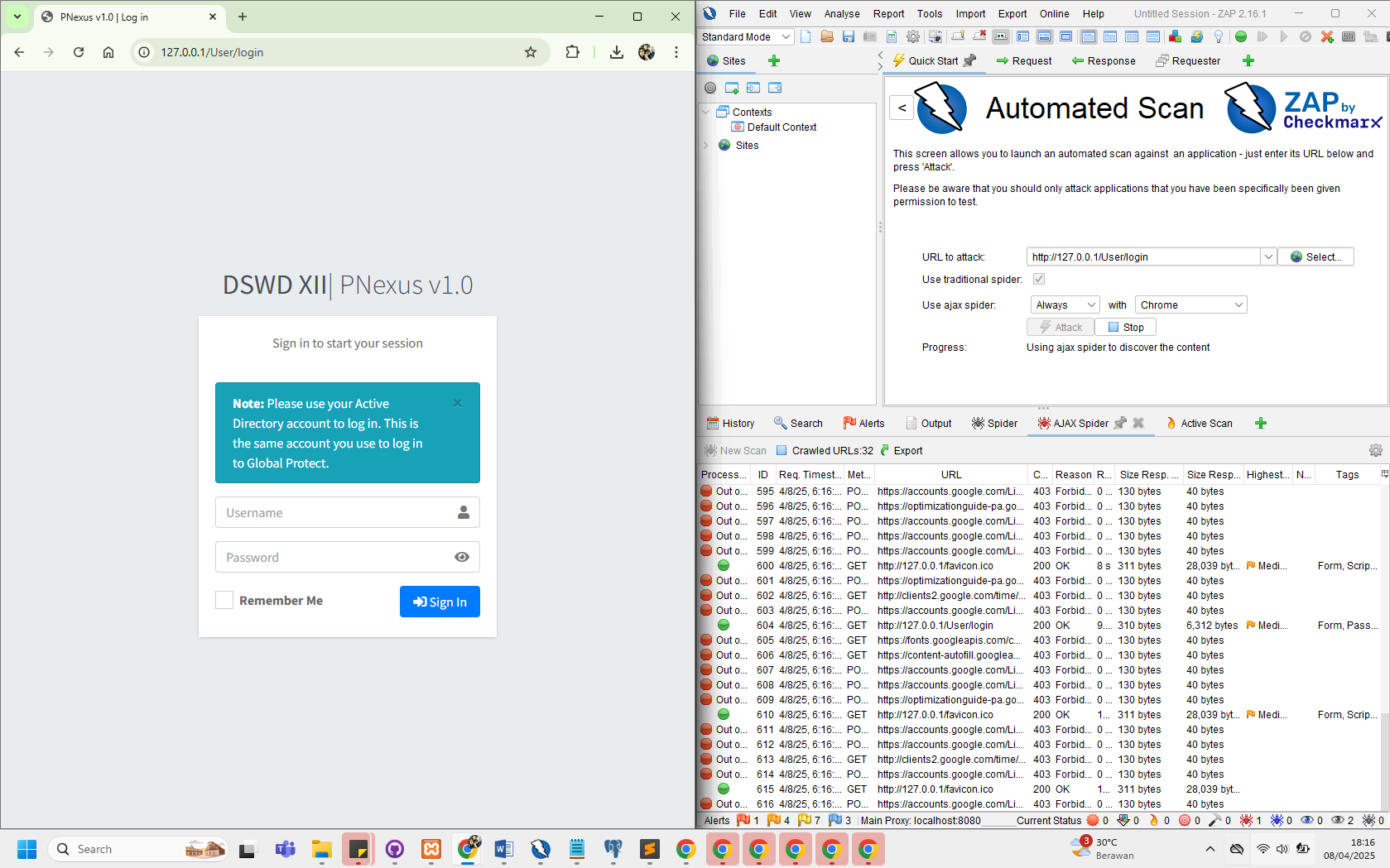
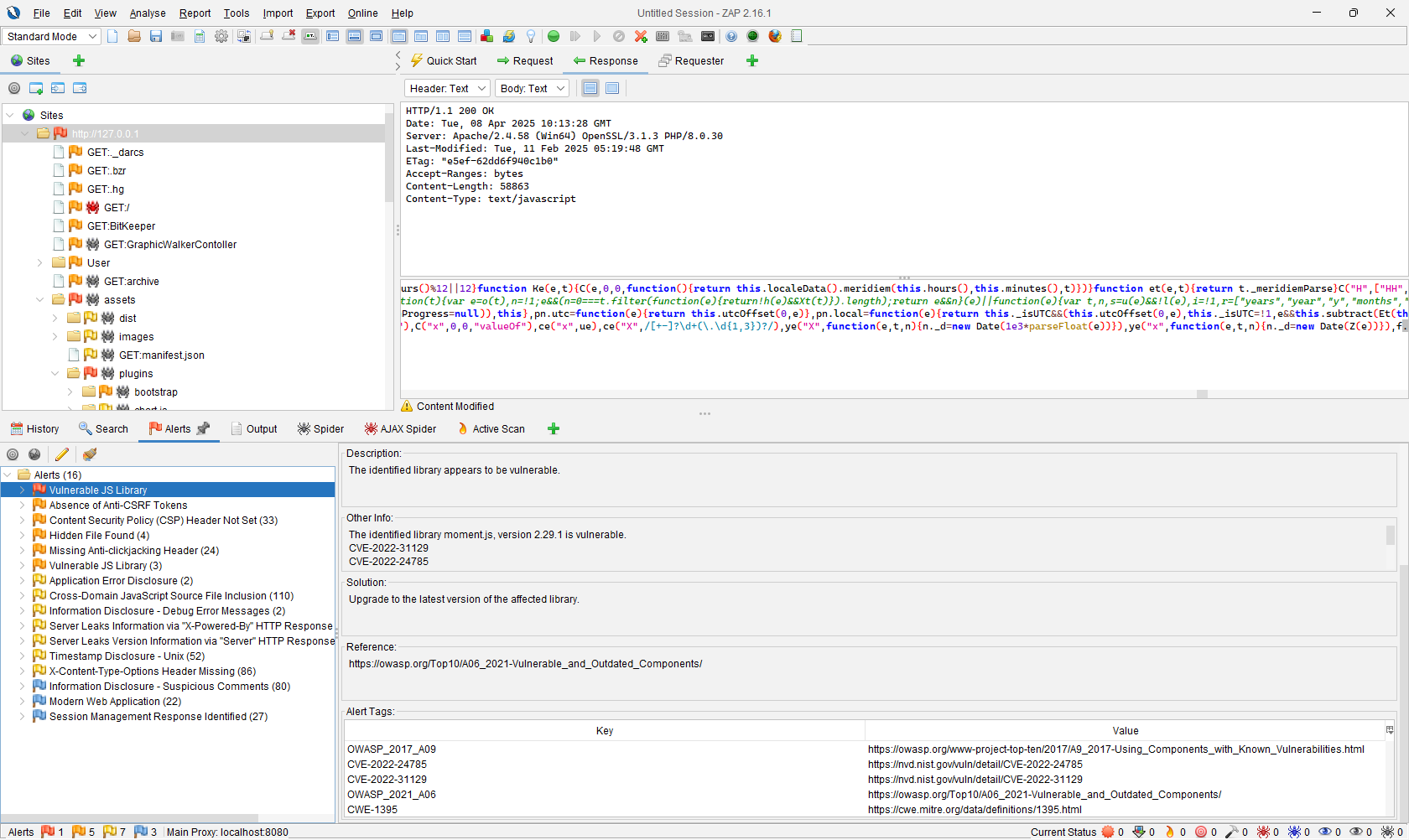


Figure 7: Screenshot showing high-risk vulnerability assessment result.



1. Comparison with Industry Security Standards
2. Infrastructure Level

The table provides a detailed risk assessment for vulnerabilities found on a system through an Nmap scan. It lists 14 vulnerabilities related to different services like Apache HTTPD, Microsoft RPC, etc.

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Vulnerability** | **Description** | **Industry Security Standard** |
| 1 | **CVE-2024-38476 (Apache HTTPD)** | Remote code execution vulnerability in Apache HTTPD 2.4.58. | **NIST SP 800-53 (System and Communications Protection - SC)**: Ensures secure configuration and patch management practices to prevent exploits like these. |
| 2 | **CVE-2024-38474 (Apache HTTPD)** | Remote code execution vulnerability in Apache HTTPD 2.4.58. | **OWASP Top 10 (A9 - Using Components with Known Vulnerabilities)**: Using vulnerable components like outdated versions of HTTPD increases security risks. |
| 3 | **CVE-2024-38475 (Apache HTTPD)** | Remote code execution vulnerability in Apache HTTPD 2.4.58. | **NIST SP 800-53 (Access Control - AC)**: Ensure strong access controls and update systems promptly to mitigate risks from known vulnerabilities. |
| 4 | **Slowloris DoS (Port 80)** | **Slowloris** attack can be used to keep many HTTP connections open and starve the server of resources, causing a Denial of Service (DoS). | **ISO/IEC 27001:2022 (A.14.2.5 - Protection from Malicious Code)**: Ensure effective protection against DoS attacks by implementing server resource limitations. |
| 5 | **MSRPC (Port 135)** | Microsoft RPC service exposed to the network, commonly exploited in remote attacks. | **NIST SP 800-53 (System and Communications Protection - SC)**: Ensure proper network segmentation and access control to prevent unauthorized access. |
| 6 | **CVE-2024-40898 (Apache HTTPD)** | Vulnerability in Apache HTTPD 2.4.58 that could allow for remote code execution. | **CIS Control 3.1**: Regularly apply patches to mitigate risks associated with known vulnerabilities. |
| 7 | **CVE-2024-39573 (Apache HTTPD)** | Vulnerability in Apache HTTPD 2.4.58. | **ISO/IEC 27001:2022 (A.12.6.1 - Management of Technical Vulnerabilities)**: Implement a patch management process to minimize exposure to critical vulnerabilities. |
| 8 | **CVE-2024-38477 (Apache HTTPD)** | Vulnerability in Apache HTTPD 2.4.58 that could allow an attacker to execute arbitrary code. | **NIST SP 800-53 (Risk Assessment - RA)**: Assess and mitigate vulnerabilities in all deployed services and software. |
| 9 | **CVE-2024-27316 (Apache HTTPD)** | Vulnerability in Apache HTTPD 2.4.58 that could allow a remote attacker to execute arbitrary code. | **CIS Control 4.8**: Regular patching of software components to minimize risk from known vulnerabilities. |
| 10 | **CVE-2024-38476, CVE-2024-38475 (Apache HTTPD)** | Multiple vulnerabilities identified in Apache HTTPD 2.4.58. | **ISO/IEC 27001:2022 (A.10.1.1 - Cryptographic Controls)**: Apply encryption and security patches to mitigate the impact of vulnerabilities like these. |
| 11 | **RDP (Port 3389)** | Remote Desktop Protocol exposed, vulnerable to brute-force and credential stuffing attacks. | **OWASP Top 10 (A8 - Insecure Deserialization)**: Ensure secure access to sensitive services like RDP, with strong authentication mechanisms. |
| 12 | **PostgreSQL DB (Port 5432)** | Exposed **PostgreSQL** database potentially allowing unauthorized access. | **NIST SP 800-53 (Access Control - AC)**: Implement strict access controls for databases and sensitive services. |
| 13 | **Grafana (Port 3000)** | Exposed Grafana service without proper authentication. | **OWASP Top 10 (A2 - Broken Authentication)**: Proper authentication is required for accessing sensitive administrative interfaces like Grafana. |
| 14 | **TRACE Enabled (Port 80)** | The **TRACE** HTTP method is enabled, potentially exposing sensitive data and increasing vulnerability to Cross-Site Tracing (XST) attacks. | **OWASP Top 10 (A5 - Security Misconfiguration)**: Misconfiguration of web services like HTTP methods creates unnecessary attack vectors. |

1. Application Level

The identified vulnerabilities were assessed in relation to widely recognized cybersecurity frameworks and standards to determine their severity, prevalence, and recommended remediation strategies. Notably, many findings align with critical areas outlined in the OWASP Top 10 (2021), Common Weakness Enumeration (CWE), and Web Application Security Consortium (WASC) Threat Classification.

One of the most critical findings—the use of vulnerable and outdated JavaScript libraries such as moment.min.js and bootstrap.bundle.min.js—falls under OWASP A06:2021 – Vulnerable and Outdated Components. Using known-vulnerable components in a production environment can lead to severe exploits, including remote code execution or cross-site scripting. This issue also maps to CWE-1104: Use of Unmaintained Third-Party Components.

The table below shows how the findings relate to industry security standards:

|  |  |  |  |
| --- | --- | --- | --- |
| **Vulnerability** | **OWASP**  **Top 10** | **CWE by MITRE**  *Common Weakness Enumeration* | **WASC**  *Web Application Security Consortium* |
| Vulnerable JS Library | A06: Vulnerable Components | CWE-1104 | WASC-20: Improper Input Handling |
| Missing CSP Header | A05: Security Misconfiguration | CWE-693 | WASC-15: Application Misconfiguration |
| Missing X-Frame-Options | A05: Security Misconfiguration | CWE-1021 | WASC-15 |
| Missing Anti-CSRF Tokens | A01: Broken Access Control | CWE-352: Cross-Site Request Forgery | WASC-9: CSRF |
| Hidden .hg Directory | A05: Security Misconfiguration | CWE-200: Exposure of Sensitive Info | WASC-13: Info Leakage |
| Suspicious Comments/Debug Info | A03: Injection / A05: Misconfiguration | CWE-615: Debug Info Exposure | WASC-13 |

The analysis clearly shows that multiple findings correspond to major categories of application vulnerabilities commonly exploited in real-world attacks. These gaps not only pose technical risks but also reduce compliance with standards such as ISO 27001:2013 (Annex A.14 – System Acquisition, Development, and Maintenance) and NIST SP 800-53 (System and Information Integrity SI-10, SI-2). Aligning the system with industry standards requires addressing these vulnerabilities through component updates, HTTP header configuration, and secure development lifecycle (SDLC) improvements. Regular vulnerability assessments and automated dependency checks should be integrated into the development and deployment pipeline to maintain compliance and reduce risk exposure over time.