

SECURITY AUDIT OF PNEXUS WEB APPLICATION AND ITS DEPLOYMENT ENVIRONMENT.

Project Type: Security Audit

This proposal outlines a comprehensive security audit of the PNexus Web Application used by the Department of Social Welfare and Development (DSWD) under the Pantawid Pamilyang Pilipino Program (4Ps). The audit will focus on identifying vulnerabilities within the application, deployment environment, and infrastructure, specifically using OWASP ZAP and Wireshark as the primary tools for testing and analysis.

Scope and Limitations

1. The primary focus will be on identifying vulnerabilities within the PNexus Web Application and its associated security components, which are part of the Pantawid Pamilyang Pilipino Program. This will include a comprehensive evaluation of the application's front-end, back-end, authentication mechanisms, and its integration with the underlying infrastructure.
2. The audit will focus on the security of the PNexus application and the OCP Server where the application is deployed, including the server's network infrastructure and security configurations.
3. The audit will not cover potential data inconsistencies or security issues related to data imported from the official Pantawid information system.
4. The audit will be limited strictly to the OCP Server hosting the PNexus application and will not include other systems or infrastructure within DSWD FO12.
5. The audit will not assess the backup policy, as it is not implemented in the current infrastructure.

Objectives

The main objectives of the security audit are as follows:

1. To identify possible vulnerabilities within the PNexus Web Application and deployment environment.
 - a) This includes identifying potential vulnerabilities in the codebase, user authentication, and front-end technologies.
 - b) Assess the security of the server environment, including issues such as insecure configurations, improper server hardening, and other potential deployment flaws.
2. To provide recommend best practices and security enhancements to mitigate identified vulnerabilities.

Target System or Case Study

The case study for this audit focuses on the PNexus Web Application, a system used by the Department of Social Welfare and Development (DSWD) as part of the Pantawid Pamilyang Pilipino Program. This application is built using several key technologies, which include CodeIgniter version 3 for the back-end, with the system lacking built-in ORM support, potentially exposing it to SQL injection vulnerabilities. The system currently operates over HTTP, an insecure protocol that poses a risk to data integrity and confidentiality. On the front-end, the application utilizes jQuery, HTML5, and Bootstrap 4, which can be susceptible to security issues like cross-site scripting (XSS) or cross-site request forgery (CSRF) if not properly configured.

Regarding security mechanisms, the application lacks CAPTCHA, most like it more vulnerable to bot attacks, and there is no implementation of Two-Factor Authentication (2FA), which increases the probability of account compromises. The system integrates with Active Directory for user authentication, which seems to be secure but the way it integrates with the web-application should also be checked. Access to the system is limited to the GlobalProtect VPN, which adds a layer of security but requires thorough assessment to identify potential vulnerabilities. The ICT infrastructure also includes a VPN network that provides secure remote access, though the configuration and security of this network need to be reviewed. In deployment environment, the system runs in a Hyper-V virtualized environment on an OCP server, which requires a detailed security review for potential vulnerabilities. One possible critical issue is the possible absence of a formal backup policy; thus critical data may be at risk of loss in the event of a breach or system failure.

Tools and Frameworks to be Used

1. OWASP ZAP - An open-source web application security scanner will be the primary tool used to perform dynamic application security testing (DAST) on the PNexus Web Application. ZAP will help identify issues such as SQL injection, Cross-Site Scripting (XSS), insecure HTTP methods, and other common vulnerabilities in web applications.
2. Wireshark - A network protocol analyzer will be used to capture and analyze network traffic to check for unencrypted data transmission and other potential vulnerabilities in communication between the client and server.
3. Nmap (Network Mapper) - will be used to perform network reconnaissance and vulnerability scanning. With its Nmap Scripting Engine (NSE), it can detect open ports, service versions, operating system fingerprints, and common vulnerabilities in running services (e.g., outdated Apache, weak SSL ciphers, known CVEs).

Ethical and Legal Considerations

The security audit will be conducted in accordance with ethical standards, ensuring that no data is compromised, and privacy is respected throughout the process. The following legal and ethical guidelines will be adhered to:

1. Philippine Data Privacy Act of 2012 (Republic Act No. 10173) - A law mandates that the collection, processing, and storage of personal data must be done securely, and it provides guidelines for data protection. All actions during the audit will comply with the provisions of this law to ensure the protection of personal and sensitive data.
2. Prior to starting the audit, consent will be obtained from the relevant authorities at the DSWD to ensure that the audit is conducted with full knowledge and approval.

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.

2. Use of Security Tools for Data Gathering

A. 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.

B. 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.

3. Risk and Vulnerability Analysis

A. 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 was 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:
 - o Modular Gateway Router
 - o Palo Alto Ingress Firewall
 - o Palo Alto Intrusion Detection System (IDS)
 - o PA-Series Next-Generation Firewall (NGFW)
 - o Distribution Router with two connected switches
 - o Patch panels for structured distribution

b. Office-Specific Network Cabinets

- ✓ Each office has a dedicated cabinet containing:
 - o Repeater
 - o Router
 - o Switches
 - o 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:

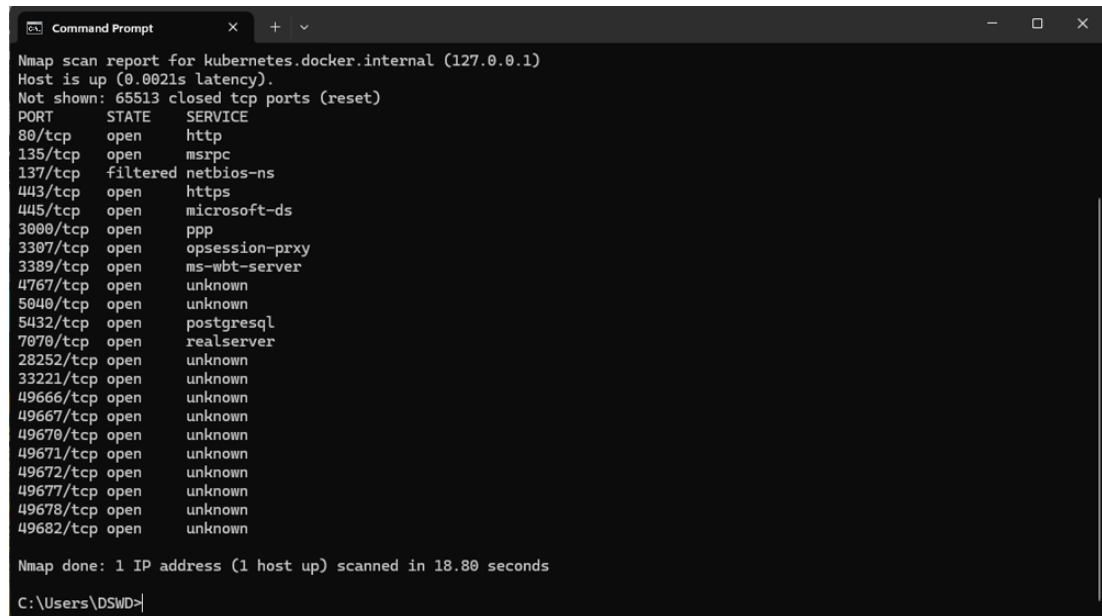
- ✓ Heavy Duty UPS
- ✓ Automatic transfer switches
- ✓ Backed with industrial heavy duty generator

Risk Assessment

#	Potential Risk	Risk Level	Impact	Recommendation
1	Absence of VLANs 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 VLAN.
2	Absence of DMZ.	Low	As technology evolved, malicious technique also evolved.	The current infrastructure and security feature of the network is really impressive but it is recommended the maximum/latest security technology as possible. Creation of DMZ is highly recommended.

B. Deployment Environment

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



```
Command Prompt
Nmap scan report for kubernetes.docker.internal (127.0.0.1)
Host is up (0.0021s latency).
Not shown: 65513 closed tcp ports (reset)
PORT      STATE SERVICE
80/tcp    open  http
135/tcp    open  msrpc
137/tcp    filtered netbios-ns
443/tcp    open  https
445/tcp    open  microsoft-ds
3000/tcp   open  ppp
3307/tcp   open  opsession-prxy
3389/tcp   open  ms-wbt-server
4767/tcp   open  unknown
5040/tcp   open  unknown
5432/tcp   open  postgresql
7070/tcp   open  realserver
28252/tcp  open  unknown
33221/tcp  open  unknown
49666/tcp  open  unknown
49667/tcp  open  unknown
49670/tcp  open  unknown
49671/tcp  open  unknown
49672/tcp  open  unknown
49677/tcp  open  unknown
49678/tcp  open  unknown
49682/tcp  open  unknown

Nmap done: 1 IP address (1 host up) scanned in 18.80 seconds
C:\Users\DSWD>
```

Risk Assessment

#	Vulnerability	Risk Level	Impact	Recommendation
1	Port 80 is Open	Medium	Potential information leakage before HTTPS redirect; vulnerable to downgrade attacks.	Force unencrypted request redirect to https; disable port 80 if not strictly required.
2	Port 135 (RPC) is Open	High	Remote code execution risk; commonly targeted by malware like WannaCry.	Restrict to internal/trusted IPs only; Consider blocking externally.
3	Port 137 (NetBIOS)	High	Info disclosure of network names; exploited for reconnaissance and lateral movement.	Disable NetBIOS over TCP/IP;

4	Port 443 (HTTPS)	Low	Secure if TLS is enforced, but vulnerable to weak cipher suites or bad certificates.	Enforce TLS 1.2/1.3, valid certs, strong ciphers, HSTS, and regular vulnerability scanning.
5	Port 445 (SMB)	High	Known SMB vulnerabilities; targeted in ransomware attacks.	Block if not needed; disable SMBv1; restrict access via firewall;
6	Port 3000	Medium	Often exposes dev environments (e.g., Node.js); may contain misconfigured services.	Restrict to internal use;
7	Port 3307 (MySQL alt)	Medium	May expose database to attacks if not secured; not default MySQL port.	Use strong auth, restrict to internal IPs, enable SSL, monitor access logs.
8	Port 3389 (RDP)	High	High-profile attack vector for brute-force and remote access exploitation.	Should be blocked / GUI based remote access is prohibited (Only RPC/SSL Allowed)
9	Port 5432 (PostgreSQL)	Medium	May expose DB to unauthorized access or injection if insecure.	Use SSL, strong passwords, limit external access;
10	Port 7070 (RealServer)	Medium	Often used by old streaming services; might be outdated or vulnerable.	disable or replace with modern alternatives; apply latest patches.
11	Ports 4767	Medium	Global Protect Portal	Can only be access using active directory (ldap) account
12	Port 5040 (PRC)	Medium	PRC Port	Restricted/Internal access only
13	Ports 28252+	Medium	High ports may indicate dynamic RPC or malicious backdoors.	These ports are unknown and no-known application using it. It should be blocked

2. Comparison to International Standards

#	Port/Service	Description	International Standards Reference
1	Port 80 (HTTP)	Should redirect to HTTPS; exposed port may allow downgrade attacks.	OWASP ASVS, ISO 27001 A.10.1, NIST SP 800-95

2	Port 135 (RPC)	Should be blocked or IP-restricted; enables remote execution.	NIST SP 800-53 AC-4, CIS Benchmark Windows
3	Ports 137/445 (NetBIOS/SMB)	Should be disabled unless needed; often exploited.	Microsoft Security Baseline, PCI DSS, CIS
4	Port 443 (HTTPS)	Secure if TLS 1.2+ and proper certs are enforced.	ISO 27001 A.10, OWASP, NIST 800-52r2
5	Port 3000	Should not be publicly exposed unless secured.	OWASP Top 10 A5:2021 (Security Misconfiguration)
6	Port 3307 (MySQL)	DB ports must be firewalled and monitored.	OWASP, NIST SP 800-171, CIS Benchmarks
7	Port 3389 (RDP)	Should be behind VPN, with NLA and MFA.	NIST SP 800-46, CIS Control 9, ISO 27001 A.13
8	Port 5432 (PostgreSQL)	Encrypt DB traffic; strong auth; restrict access.	CIS PostgreSQL Benchmark, NIST CM-6
9	Ports 4767–49680	Unknown or dynamic RPC ports — should be documented, restricted, monitored.	CIS Control 9, NIST CM-7
10	Port 7070	RealServer often outdated and insecure.	ISO 27001 A.12.6.1 (Malware Controls)

The table helps prioritize remediation efforts, starting with the most critical vulnerabilities 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.

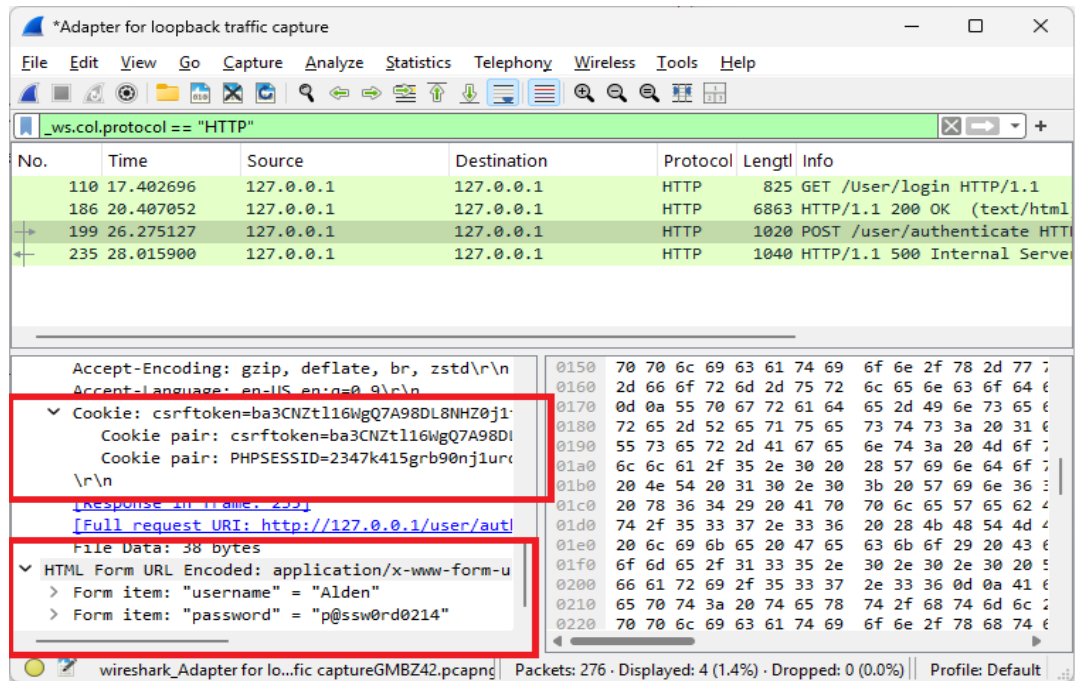
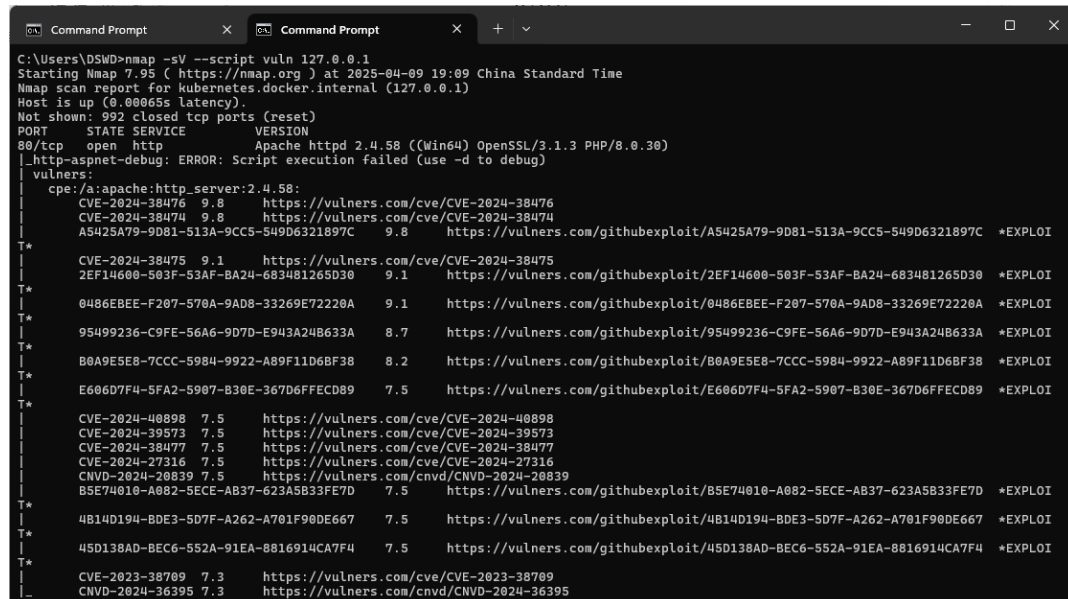


Figure 6: Screenshot of server vulnerability scan using NMAP



C. 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.

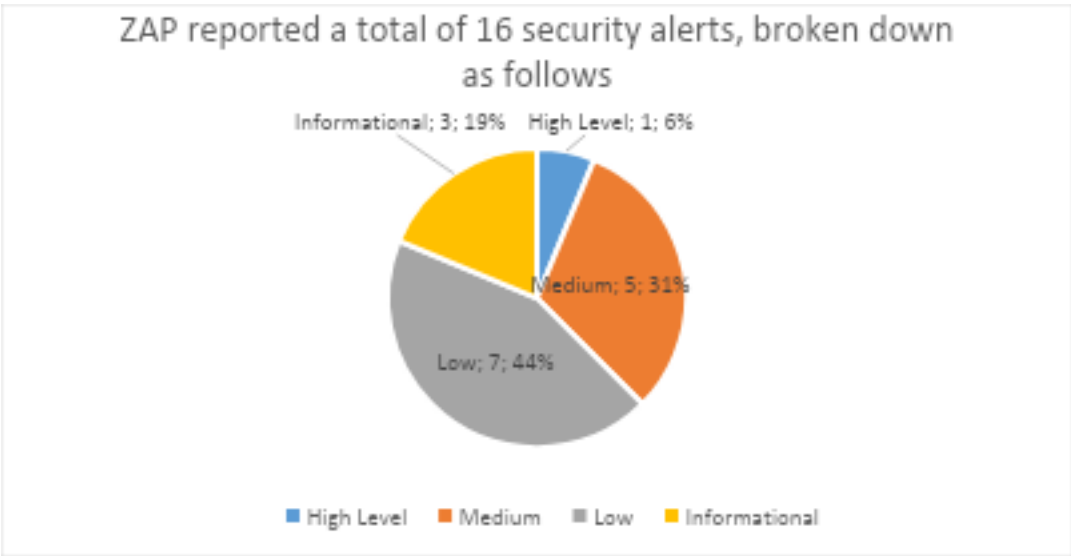
Both the Medium Risk and Low Risk categories have similar distributions. While the Medium Risk category has a mix of Low Confidence (12.5%) and User Confirmed (12.5%) alerts, the Low Risk category contains a larger portion of Low Confidence alerts (31.2%). Despite this, Low Risk still represents the 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)	It is recommended to replace this

		<ul style="list-style-type: none"> - 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. 	library with the latest, secure version to mitigate the risk.
Medium	5	<ul style="list-style-type: none"> ✓ 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	<ul style="list-style-type: none"> ✓ 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 	Review and remove sensitive information in headers and responses. Investigate and address the cross-

		<p>with information about the system that can be exploited. These messages should be suppressed in the production environment.</p> <ul style="list-style-type: none"> ✓ 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. 	domain script inclusion.
Informational	3	<ul style="list-style-type: none"> ✓ 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

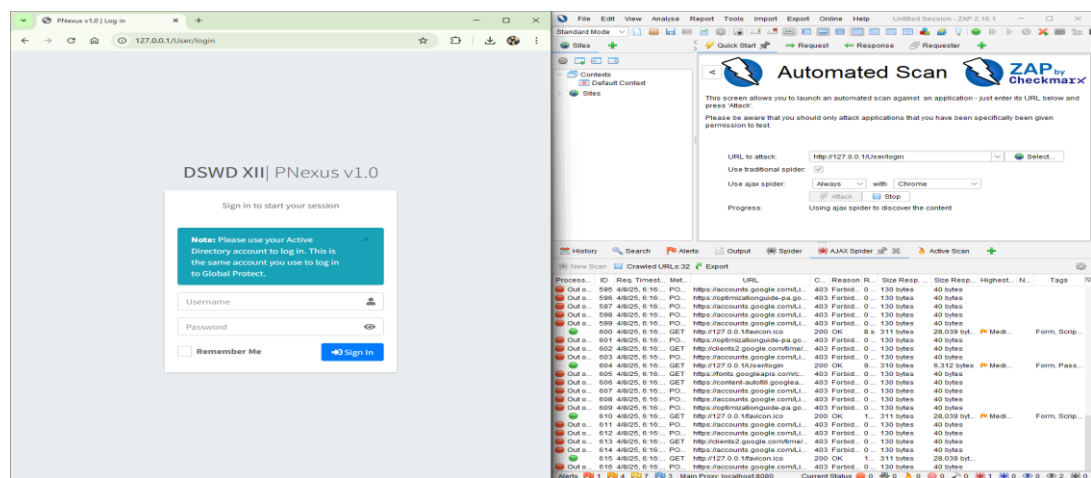
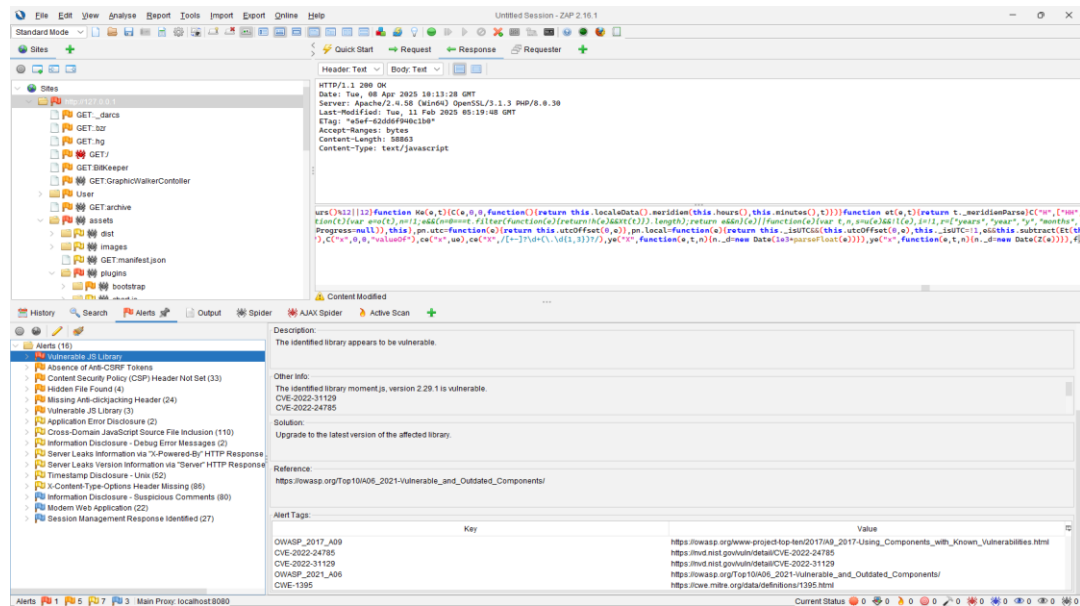


Figure 7: Screenshot showing high-risk vulnerability assessment result.



4. Comparison with Industry Security Standards

A. 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.

#	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-	OWASP Top 10 (A8 - Insecure Deserialization): Ensure secure access to sensitive services like

		force and credential stuffing attacks.	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.

B. 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 <i>Common Weakness Enumeration</i>	WASC <i>Web Application Security Consortium</i>
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.

PHASE 3: IMPLEMENTATION & TESTING

3.1 SUMMARY OF APPLIED PATCHES AND FIXES

This section covered the security patches and configuration changes applied to the PNexus Web Application environment in response to vulnerabilities identified during the recent network vulnerability scan using Nmap and NSE scripts.

Code Legends:

1. Codes with negative sign tells that the line(s) were removed
2. Codes with positive sign tells that the line(s) were added
3. Codes without signs means that there were no changes made

a. Vulnerable JS Library (Outdated)

Updated jQuery to the latest version to patch known vulnerabilities and prevent exploitation via outdated libraries.

Modified: header.php
Risk=High, Confidence=Medium (1)
Vulnerabilities: CVE-2022-24785, CVE-2022-31129, OWASP_2021_A06, CWE-1395
- <script src="js/jquery-1.12.4.min.js"></script> + <script src="https://code.jquery.com/jquery-3.7.1.min.js"></script>

b. CSP Header Not Set & Missing Anti-clickjacking Header

Added Content-Security-Policy (CSP) header to restrict frame embedding and reduce risk of clickjacking and content injection.

Modified: .htaccess
(a) Risk=Medium, Confidence=High (1) ; # CSP Header Not Set
(a) Vulnerabilities: CWE-693, OWASP_2021_A05, OWASP_2017_A06
(b) Risk=Medium, Confidence=Medium (2); # Missing Anti-clickjacking Header
(b) Vulnerabilities: WSTG-v42-CLNT-09, OWASP_2021_A05, OWASP_2017_A06, CWE-1021
+<IfModule mod_headers.c> + Header always set Content-Security-Policy " + frame-ancestors 'self'; #prevent changing frame-source to external URI + " +</IfModule>

c. Absence of Anti-CSRF Tokens

Enabled CSRF protection in CodeIgniter config to safeguard forms from cross-site request forgery attacks.

Modified: application/config/config.php
Risk=Medium, Confidence=Low (2)
Vulnerabilities: OWASP_2021_A01, WSTG-v42-SESS-05, OWASP_2017_A05, CWE-352
+\$config['csrf_protection'] = TRUE; +\$config['csrf_token_name'] = 'csrf_token_pnexus'; +\$config['csrf_cookie_name'] = 'csrf_cookie_pnexus'; +\$config['csrf_expire'] = 7200; +\$config['csrf_regenerate'] = TRUE; +\$config['csrf_exclude_uris'] = array();

d. Server Leaks "Server" Header

Configured Apache to suppress the "Server" header to prevent revealing backend technology details to potential attackers.

Modified: apache/conf/httpd.conf
Risk=Low, Confidence=High (1)
Vulnerabilities: OWASP_2021_A05, OWASP_2017_A06, WSTG-v42-INFO-02, CWE-497
+ServerTokens Prod +ServerSignature Off

e. Application Error Disclosure

Replaced detailed error output with generic message to prevent leaking sensitive debug info to users.

Modified: application/view/errors/error_handler.php
Risk=Low, Confidence=Medium (5)
Vulnerabilities: WSTG-v42-ERRH-02, WSTG-v42-ERRH-01, CWE-550, OWASP_2021_A05, OWASP_2017_A06
<pre>- echo \$exception; + echo "An error occurred. Please contact support.";</pre>

f. Cross-Domain JS Source File Inclusion

Strengthened CSP policy to restrict all resource types to the same origin, blocking potential XSS through cross-domain inclusions.

Modified: .htaccess
Risk=Low, Confidence=Medium (5)
Vulnerabilities: OWASP_2021_A08, CWE-829
<pre><IfModule mod_headers.c> Header always set Content-Security-Policy " + default-src 'none'; # Block all content by default; + img-src 'self'; # Allow images only from the same origin + connect-src 'self'; # Allow XHR, WebSocket, and EventSource connections only to the same origin + object-src 'none'; # Prevent embedding external objects (e.g., Flash, PDFs, videos) frame-ancestors 'self'; + base-uri 'self'; # Only allow the original server url tag to reference same-origin URLs; + form-action 'self'; # Allow form submissions only to the same origin; + upgrade-insecure-requests; # redirect all http (insecure) requests to https (secure) " </IfModule></pre>

g. Debug Error Messages Disclosure

Disabled PHP error display to hide internal application messages that could aid attackers in crafting attacks.

Modified: apache/conf/httpd.conf
Risk=Low, Confidence=Medium (5)
Vulnerabilities: OWASP_2021_A01, WSTG-v42-ERRH-01, OWASP_2017_A03, CWE-1295
<pre>+php_flag display_errors Off</pre>

h. Server Leaks "X-Powered-By" Header

Disabled `expose_php` and unset X-Powered-By header to conceal PHP usage and version from attackers.

Modified files: <code>php.ini</code> , <code>.htaccess</code>
Risk=Low, Confidence=Medium (5)
Vulnerabilities: OWASP_2021_A01, OWASP_2017_A03, WSTG-v42-INFO-08, CWE-497
<pre>// php.ini -expose_php = On +expose_php = Off // .htaccess <IfModule mod_headers.c> + Header unset X-Powered-By # prevents apache from displaying PHP versions in http response. Header always set Content-Security-Policy " default-src 'none'; img-src 'self'; connect-src 'self'; object-src 'none'; frame-ancestors 'self'; base-uri 'self'; form-action 'self'; upgrade-insecure-requests; " </IfModule></pre>

i. X-Content-Type-Options Header Missing

Added `nosniff` header to prevent MIME type sniffing, which could allow execution of malicious files.

Modified files: <code>.htaccess</code> , <code>httpd.conf</code>
Risk=Low, Confidence=Medium (5)
Vulnerabilities: CWE-693, OWASP_2021_A05, OWASP_2017_A06
<pre>// .htaccess <IfModule mod_headers.c> # developer should explicitly specify the mime type (server should not guess it) + Header set X-Content-Type-Options "nosniff" Header unset X-Powered-By </IfModule mod_headers.c> Header unset X-Powered-By Header always set Content-Security-Policy " default-src 'none';</pre>

```

img-src 'self';
connect-src 'self';
object-src 'none';
frame-ancestors 'self';
base-uri 'self';
form-action 'self';
upgrade-insecure-requests;
"
</IfModule>

// httpd.conf
# 'nosniff' prevents browsers from interpreting files based on their content
# and forces them to adhere strictly to the declared Content-Type header.
# This protects against attacks where an attacker may try to disguise a malicious file with a
# misleading extension.
+<FilesMatch "\.(html|css|js|png|jpg|jpeg|gif|php)$">
+  Header set X-Content-Type-Options "nosniff"
+</FilesMatch>

```

j. Timestamp Disclosure – Unix

Identified as low-risk; no fix applied since exposed timestamps are not considered sensitive in current context.

Modified: None
Risk=Low, Confidence=Low (1)
Vulnerabilities: OWASP_2021_A01, OWASP_2017_A03, CWE-497
//No configuration made because the risk level is tolerable.

k. Comments in Javascripts

No sensitive data found in JS comments; retained for code readability and development documentation purposes.

Modified: None
Risk=Informational, Confidence=Medium (3)
Vulnerabilities: OWASP_2021_A01, WSTG-v42-INFO-05, OWASP_2017_A03, CWE-615
//No change made because comments don't contains sensitive information //these also helps developer tracks code easily //it is also enforce developer to add doc strings on javascripts

l. Web Crawling Enabled

Removed robots.txt to avoid exposing paths or structure of the web application to web crawlers.

Modified: robots.txt
Risk=Informational, Confidence=Medium (3)
Vulnerabilities: No CVE
- User-agent: * - Disallow: / + # robots.txt removed to prevent unintended crawling

m. Session Info in JS Console

Removed session information from console logs to prevent exposure of user data in browser developer tools.

Modified: main.js
Risk=Informational, Confidence=Medium (3)
Vulnerabilities: No CVE
- consoe.log (response.session.username) + // Removed logging of session info

n. Disabled Weak protocols and Cyphers

Disables outdated and vulnerable SSL/TLS versions. Only allows strong ciphers for encrypted communication. Forces server-preferred ciphers to enhance security.

Modified: httpd-ssl.conf
Risk=Informational, Confidence=Medium (3)
Vulnerabilities: No CVE
httpd-ssl.conf # Disable weak protocols # allow all (including TLS: 1.3 which uses AES-128 and AES-256) # recommended to use -TLSv1.3 SSLProtocol All -SSLv2 -SSLv3 -TLSv1 -TLSv1.1 # Disable weak ciphers SSLCipherSuite HIGH:!aNULL:!MD5 SSLHonorCipherOrder On

o. Enforce SSL Encryption

Ensure encrypted communication between clients and the PNexus Web Application, an SSL certificate was installed and configured on the Apache server.

Encryption Type:

- a. For SSL handshake: RSA, ECDSA, or DH
- b. For Data Transfer : AES (usually AES-128 or AES-256)

Modified: httpd-ssl.conf
Risk=Informational, Confidence=Medium (3)
Vulnerabilities: No CVE
<pre># httpd-ssl.conf + <VirtualHost _default_:443> + DocumentRoot "C:/xampp/htdocs/pnexus" + ServerName entdswd.local + SSLEngine on + SSLCertificateFile "conf/ssl/pnexus.crt" + SSLCertificateKeyFile "conf/ssl/pnexus.key" + <Directory "C:/xampp/htdocs/pnexus"> + AllowOverride All + Require all granted + </Directory> + </VirtualHost></pre>

- p. Resolution for CVE-2007-6750 detected CVE after patched applied

Limits how long Apache will wait for a client (browser, script, or attacker) to send the HTTP request headers and body. It's a defense against slow clients.

Modified: httpd.conf, httpd-default.conf
Unknown
Vulnerabilities: CVE-2007-6750 (Slowloris)
<pre># httpd.conf - # LoadModule reqtimeout_module modules/mod_reqtimeout.so + LoadModule reqtimeout_module modules/mod_reqtimeout.so</pre>
<pre># httpd-default.conf + KeepAlive On + MaxKeepAliveRequests 100 + KeepAliveTimeout 5 + <IfModule reqtimeout_module> + RequestReadTimeout header=10-20,MinRate=500 body=10,MinRate=500 + </IfModule></pre>

- q. Blocked Port Used by Avahi

Modified: Firewall configuration

Unknown
Vulnerabilities: CVE-2011-1002
! /bin.bash sudo ufw deny proto udp from any to any port 5353 sudo ufw reload

r. Port Configuration (Firewall & Host Security)

The following table outlines the current firewall and host security port settings:

#	Port	Protocol/Service	Status	Mitigating Action
1	80	HTTP	Redirect only	Configured webserver to redirect unsecured request https (443).
2	135	RPC	Allow (restricted)	Allow only from trusted IPs for remote execution; Monitor and log usage (Network Administrator).
3	137	NetBIOS Name Service	Block	Disable unless needed for legacy support;
4	443	HTTPS	Open	Secure with TLS 1.3; Maintain valid certificates. Disabled Weak Cyphers
5	445	SMB	Block	Block /disable SMB; Monitor for signs of exploitation like EternalBlue.
6	3000	https	Allow (restricted)	Used for staging as preparation for Vulnerability Assessment
7	3307	MySQL alt port	Blocked	The application don't use MySQL
8	3389	RDP	Blocked	blocked - GUI based remote access is prohibited (Only RPC/SSL Allowed internally)
9	5432	PostgreSQL	Open (restricted)	Allow only specific IP Address; Block access outside DSWD Intranet;
10	7070	RealServer (Streaming)	Block	Blocked – Does not use streaming service.
11	4767	Global Protect	Allow (restricted)	Restrict access to DSWD resources using firewall roles base on the active directory group assigned.
12	5040	PRC	Allow (restricted)	Restricted only to access by Network Administrator and IDS/IPS
13	28252+	Unknown Ports	Block	Unknown Ports are blocked by default (whitelisting);

a. Infrastructure enhancement recommendation

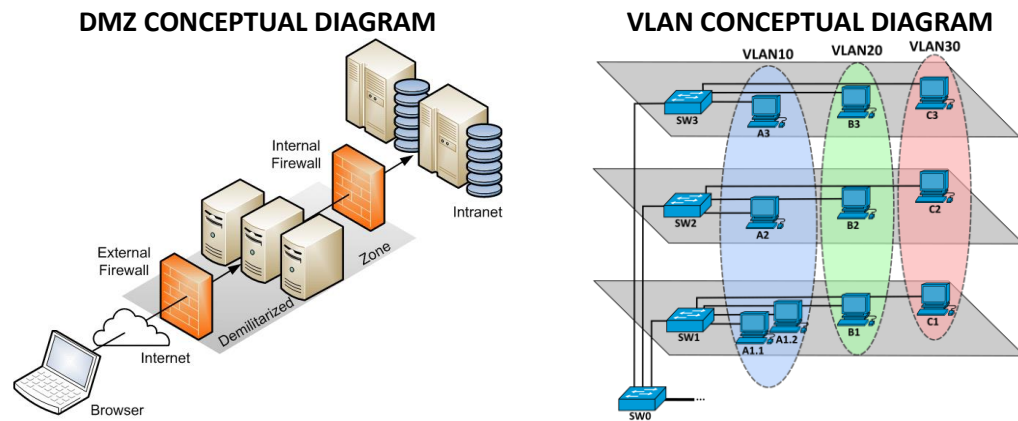
The current network structure of DSWD Field Office XII is highly secure, operating within a VPN/Intranet environment fortified by multiple layers of internal network security. However, in light of growing cyber threats and the demand for a more scalable, segmented, and responsive infrastructure, it is recommended to upgrade the existing setup to a more advanced and future-ready network architecture.

Proposed Enhancements

Proposed Enhancement	Description	Advantages
Deployment of a Demilitarized Zone (DMZ)	A DMZ is a physical or logical subnetwork that separates the internal network from untrusted external networks	1. Isolation of Public Services: Keeps public-facing servers separate from the internal network. 2. Enhanced Security: Adds a control layer for traffic flow. 3. Controlled Access: Restricts external access only to DMZ resources, not internal systems.
Implementation of VLANs per Division/Unit	VLANs logically segment the network at Layer 2, isolating traffic per division/unit even over shared infrastructure.	1. Improved Network Performance: Minimizes broadcast traffic. 2. Enhanced Security: Prevents inter-division threats by isolating traffic. 3. Simplified Management: Facilitates easier monitoring, policy application, and troubleshooting.

Action Taken

1. Collaboration with ICTMS
Coordinated with the Information and Communications Technology Management Section (ICTMS) to identify the necessary ICT equipment through market research and technical feasibility studies.
2. Preparation of Justification Document
ICTMS to drafting a formal justification for the proposed structural enhancements, detailing benefits, risk mitigations, and compliance with national ICT standards.
3. Deployment Plan Development
A step-by-step deployment plan is being prepared, including hardware acquisition, implementation phases and testing.
4. Integration into ISSP (2026–2028)
ICTMS will incorporate the proposed network architecture, deployment roadmap, and budgetary requirements into the Information Systems Strategic Plan (ISSP) as part of the three-year ICT modernization agenda.

Conceptual Diagrams:**1. POST-REMEDIATION RISK ASSESSMENT**

The table below summarizes the number of security issues before and after applying fixes. All high-risk issues and six medium-risk issues were successfully resolved. After remediation, no high-risk or medium-risk issues remain. Regarding low-risk issues, four were initially identified, with three resolved. One low-risk issue remains, but it is considered tolerable. For informational issues, two were initially detected, and one was resolved. Four additional informational issues were found, but only one remains, which is also deemed tolerable.

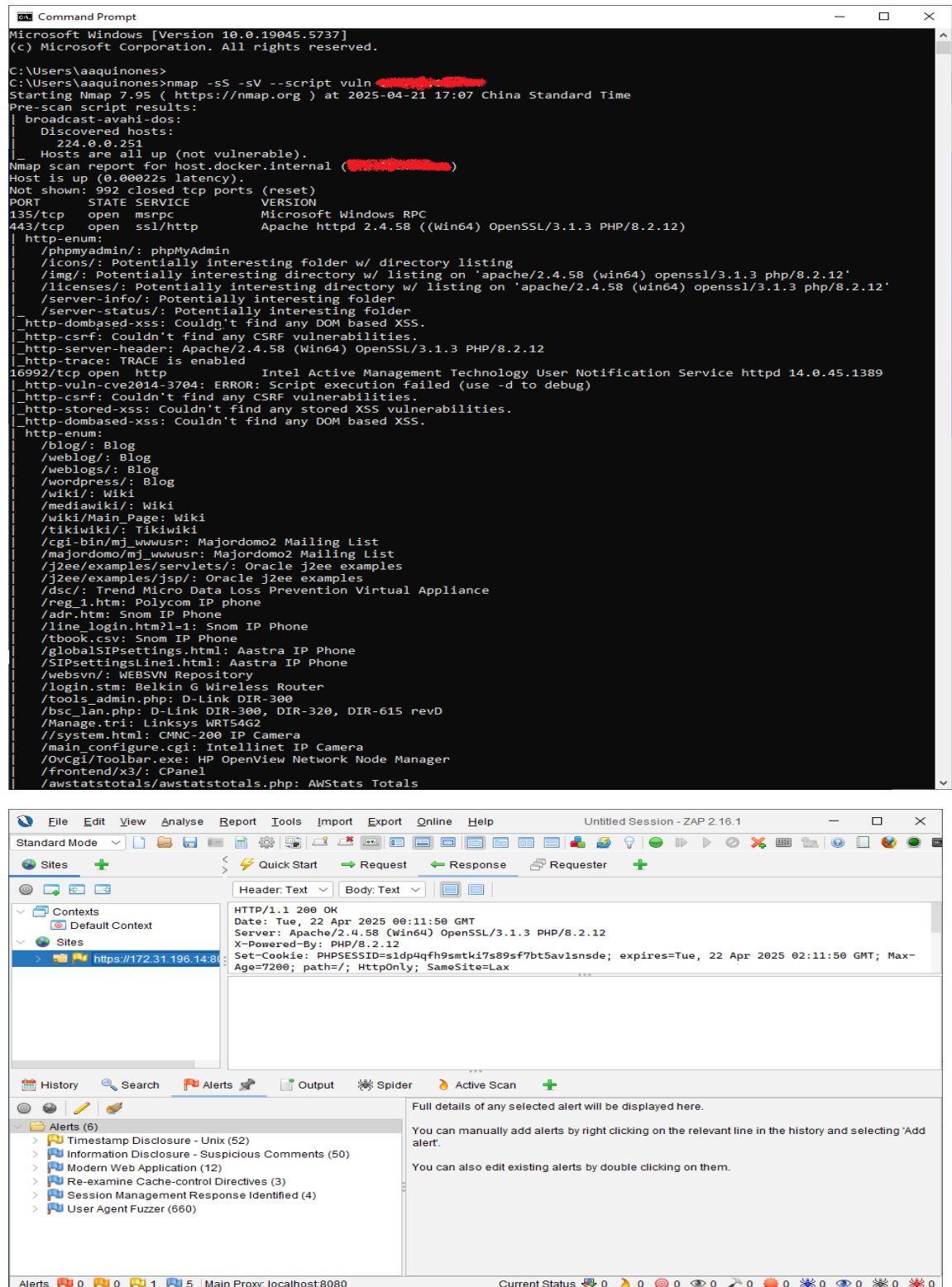
Risk Level	# of Issues (Before)	# of additional issued detected after post remediation	# of Issues (After final remediation)
High	1 resolved		0
Medium	6 resolved + 2 resolved	2 - (resolved) after migration to dockerized container	0
Low	4 (3 resolved)	1 (Tolerated)	1 (Tolerated)
Informational	2 (1 resolved)	4 (resolved)	1 (Tolerated)

IT Protection and Security: MIT 264

Alden A. Quiñones

Project: Security Audit

Screenshot 3.1: Screenshot of Post vulnerability scan after all the patches applied.



2. MITIGATION STRATEGIES AND SECURITY POLICIES

As part of Phase 3 of the security remediation process, the following mitigation strategies and security policies were developed and/or updated to address the identified vulnerabilities, minimize risk exposure, and ensure long-term protection of the system:

a. Mitigation Strategies

Category	Vulnerability Addressed	Mitigation Strategy
Web Application Security	Vulnerable JS Libraries, Missing Security Headers, CSRF, Clickjacking	<ul style="list-style-type: none">- Updated all outdated libraries using CDN versions- Enforced secure HTTP response headers via .htaccess and httpd.conf- Enabled CSRF protection in server configuration- Implemented proper error handling and custom error pages
Data Protection	Information Disclosure (X-Powered-By, Debug Mode, Server Version)	<ul style="list-style-type: none">- Disabled debug mode in production- Removed server version banners and powered-by headers- Minified JS files and removed developer comments
Access Control	Session Exposure, Absence of Anti-CSRF Tokens	<ul style="list-style-type: none">- Sanitized all session-related outputs (no exposure in JS console)- Implemented CSRF tokens for all forms
Server/Infrastructure	Open Ports, Service Fingerprinting	<ul style="list-style-type: none">- Disabled unused services- Restricted access to necessary ports only (via firewall rules)- Enforced internal-only access to MySQL and Redis
Monitoring & Logging	Application Errors, Misconfigurations	<ul style="list-style-type: none">- Enabled application logging- Integrated Prometheus and Grafana for real-time monitoring and alerts- Regularly audit logs for suspicious activity

b. Security Policies

IT Protection and Security: MIT 264

Alden A. Quiñones

Project: Security Audit

Policy Title	Description
Web Application Security Policy	Defines secure coding standards (input validation, CSRF/XSS protection), use of HTTPS, required headers (CSP, X-Content-Type-Options), and version management for libraries.
Access Control Policy	Specifies role-based access control for admin and user levels, password policy enforcement, and session timeout guidelines.
Patch Management Policy	Requires regular scanning using tools like Nmap and OWASP ZAP, and monthly review of CVEs and dependencies. Hotfix timelines are defined based on risk level.
Network Security Policy	Details port management strategy, firewall configurations, VPN access, and segregation of services (all ports/and IP should be whitelisted).
Incident Response Policy	Outlines procedures for breach identification, immediate containment, impact analysis, reporting, and recovery. Logs are preserved for forensic analysis.
Data Backup and Recovery Policy	<p>Requires encrypted backup of application databases with offsite storage and regular testing of backup integrity.</p> <p>Conduct regular/scheduled backups:</p> <ol style="list-style-type: none">1. Full backup - every first day of the month2. Differential Backup - daily at 6pm <p>Conduct a backup quarterly integrity:</p> <ol style="list-style-type: none">1. Generate MD5 Sub of all backup files2. Check backup integrity using md5Sum checker
Audit and Compliance Policy	<p>Obtain ISO Certifications:</p> <ul style="list-style-type: none">- Database Management Standards- Network & Security Standards- Server Maintenance standards<ul style="list-style-type: none">✓ Predictive Maintenance✓ Corrective Maintenance✓ Preventive Maintenance <p>Establishes annual internal audits and annual external security audits to validate compliance with industry standards like OWASP Top 10.</p>