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
4. **Vulnerable JS Library (Outdated)**

Updated jQuery to the latest version to patch known vulnerabilities and prevent exploitation via outdated libraries.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

1. **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> |

1. **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(); |

1. **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 |

1. **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 |
| - echo $exception;  + echo "An error occurred. Please contact support."; |

1. **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 |
| <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> |

1. **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 |
| +php\_flag display\_errors Off |

1. **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: php.ini, .htaccess |
| --- |
| Risk=Low, Confidence=Medium (5) |
| **Vulnerabilities:** OWASP\_2021\_A01, OWASP\_2017\_A03, WSTG-v42-INFO-08, CWE-497 |
| // 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> |

1. **X-Content-Type-Options Header Missing**

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

| Modified files: .htaccess, httpd.conf |
| --- |
| Risk=Low, Confidence=Medium (5) |
| **Vulnerabilities:** CWE-693, OWASP\_2021\_A05, OWASP\_2017\_A06 |
| // .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';  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> |

1. **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. |

1. **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 |

1. **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 |

1. **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 |

1. **Disabled Weak protocols and Cypers**

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  SSLProtocol All -SSLv2 -SSLv3 -TLSv1 -TLSv1.1  # Disable weak ciphers  SSLCipherSuite HIGH:!aNULL:!MD5  SSLHonorCipherOrder On |

1. **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:

* 1. For SSL handshake: RSA, ECDSA, or DH
  2. For Data Transfer : AES (usually AES-128 or AES-256)

| Modified: httpd-ssl.conf |
| --- |
| Risk=Informational, Confidence=Medium (3) |
| **Vulnerabilities:** No CVE |
| # 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> |

1. 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 — like Slowloris.

| Modified: httpd.conf, httpd-default.conf |
| --- |
| Unknown |
| **Vulnerabilities: CVE-2007-6750** |
| # httpd.conf  - # LoadModule reqtimeout\_module modules/mod\_reqtimeout.so  + LoadModule reqtimeout\_module modules/mod\_reqtimeout.so |
| # httpd-default.conf  + KeepAlive On  + MaxKeepAliveRequests 100  + KeepAliveTimeout 5  + <IfModule reqtimeout\_module>  + RequestReadTimeout header=10-20,MinRate=500 body=10,MinRate=500  + </IfModule> |

1. **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 |

1. **Port Configuration (Firewall & Host Security)**

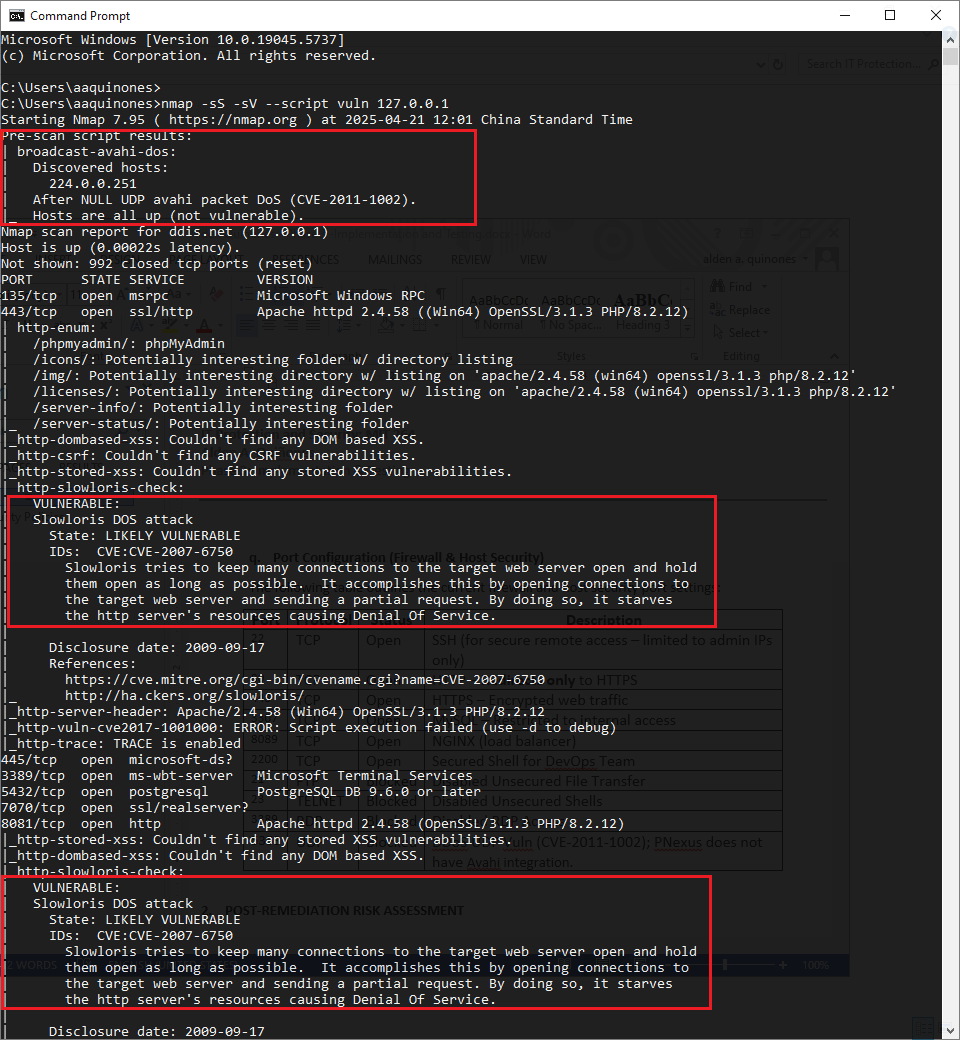
The following table outlines the current firewall and host security port settings:

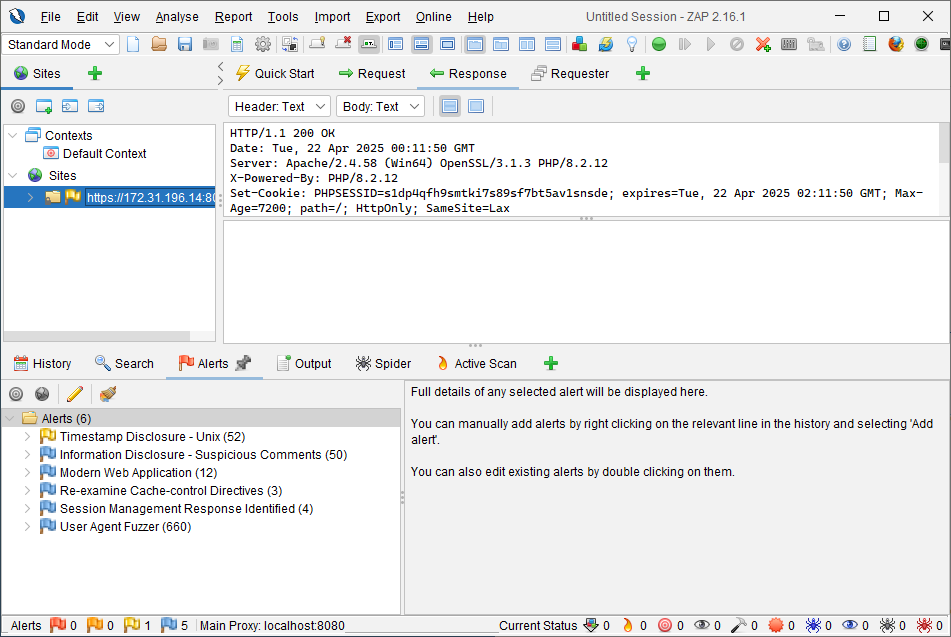
| **Port** | **Protocol** | **Status** | **Description** |
| --- | --- | --- | --- |
| 22 | TCP | Open | SSH (for secure remote access – limited to admin IPs only) |
| 80 | TCP | Open | HTTP – **Redirects only** to HTTPS |
| 443 | TCP | Open | HTTPS – Encrypted web traffic |
| 3307 | TCP | Open | MySQL – Restricted to internal access |
| 8089 | TCP | Open | NGINX (load balancer) |
| 2200 | TCP | Open | Secured Shell for DevOps Team |
| 21 | FTP | Blocked | Disabled Unsecured File Transfer |
| 23 | TELNET | Blocked | Disabled Unsecured Shells |
| 3389 | RDP | Blocked | Disabled RDP Access |
| 5353 | UDP | Blocked | DDoS UDP Vuln (CVE-2011-1002); PNexus does not have Avahi integration. |

1. **POST-REMEDIATION RISK ASSESSMENT**

The table below presents the number of security issues before and after applying fixes. All high-risk and six medium-risk issues were successfully resolved. One low-risk and one informational issue remain but are considered tolerable. Moreover, two new medium-risk and four new informational issues were identified during post-fix scans.

| **Risk Level** | **# of Issues (Before)** | **# of Issues (After)** |
| --- | --- | --- |
| High | 1 resolved | 0 |
| Medium | 6 resolved | 2 (newly detected) |
| Low | 4 (3 resolved) | 1 (Tolerated) |
| Informational | 2 (1 resolved) | 1 (Tolerated) + 4 (Newly detected) |





1. **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:

1. Mitigation Strategies

| **Category** | **Vulnerability Addressed** | **Mitigation Strategy** |
| --- | --- | --- |
| **Web Application Security** | Vulnerable JS Libraries, Missing Security Headers, CSRF, Clickjacking | - 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) | - 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 | - Sanitized all session-related outputs (no exposure in JS console) - Implemented CSRF tokens for all forms |
| **Server/Infrastructure** | Open Ports, Service Fingerprinting | - 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 | - Enabled application logging - Integrated Prometheus and Grafana for real-time monitoring and alerts - Regularly audit logs for suspicious activity |

1. Security Policies

| **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 (e.g., database not exposed to the public internet). |
| **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** | Requires encrypted backup of application databases with offsite storage and regular testing of backup integrity. |
| **Audit and Compliance Policy** | Establishes quarterly internal audits and annual external penetration testing to validate compliance with industry standards (e.g., OWASP Top 10). |