IT Protection and Security: MIT 264
Alden A. Quiñones
Phase 3: Implementation & Testing

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.

a. Vulnerable JS Library (Outdated)

Updated jQuery to latest version to patch known vulnerabilities and prevent exploitation via outdated libraries.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

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

Modified: .htaccess

Risk=Medium, Confidence=High (1)

Vulnerabilities: CWE-693, OWASP_2021_A05, OWASP_2017_A06

```
+<IfModule mod_headers.c>
+    Header always set Content-Security-Policy "frame-ancestors 'self';"
+</IfModule>
```

c. Missing Anti-clickjacking Header

Included X-Frame-Options header to block embedding in external frames, mitigating clickjacking attacks.

Modified: .htaccess

Risk=Medium, Confidence=Medium (2)

Vulnerabilities: WSTG-v42-CLNT-09, OWASP_2021_A05, OWASP_2017_A06, CWE-1021

```
<IfModule mod_headers.c>
+    Header always set X-Frame-Options "SAMEORIGIN"
   Header always set Content-Security-Policy "frame-ancestors 'self';"
</IfModule>
```

d. 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();
```

e. 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
```

f. 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.";
```

g. Cross-Domain JS Source File Inclusion

Strengthened CSP policy to restrict all resource types to 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 X-Frame-Options "SAMEORIGIN"
-   Header always set Content-Security-Policy "frame-ancestors 'self';"
+ Header always set Content-Security-Policy "
  default-src 'none';
  script-src 'self';
  style-src 'self';
  img-src 'self';
  font-src 'self';
  connect-src 'self';
  object-src 'none';
  frame-ancestors 'self';
  base-uri 'self';
  form-action 'self';
  upgrade-insecure-requests;
"
</IfModule>
```

h. 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
```

i. 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
Header always set Content-Security-Policy "
  default-src 'none';
  script-src 'self';
  style-src 'self';
  img-src 'self';
  font-src 'self';
  connect-src 'self';
  object-src 'none';
  frame-ancestors 'self';
  base-uri 'self';
  form-action 'self';
  upgrade-insecure-requests;
"
</IfModule>
```

j. 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>
+ Header set X-Content-Type-Options "nosniff"
Header unset X-Powered-By
Header always set Content-Security-Policy "
  default-src 'none';
  script-src 'self';
  style-src 'self';
  img-src 'self';
  font-src 'self';
  connect-src 'self';
  object-src 'none';
  frame-ancestors 'self';
  base-uri 'self';
  form-action 'self';
  upgrade-insecure-requests;
"
</IfModule>
```

```
// httpd.conf
+<FilesMatch "\.(html|css|js|png|jpg|jpeg|gif)$">
+   Header set X-Content-Type-Options "nosniff"
+</FilesMatch>
```

k. 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.
```

l. 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
```

m. 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
```

n. 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
```

o. 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
```

p. 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

```
# 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>
```

q. Port Configuration (Firewall & Host Security)

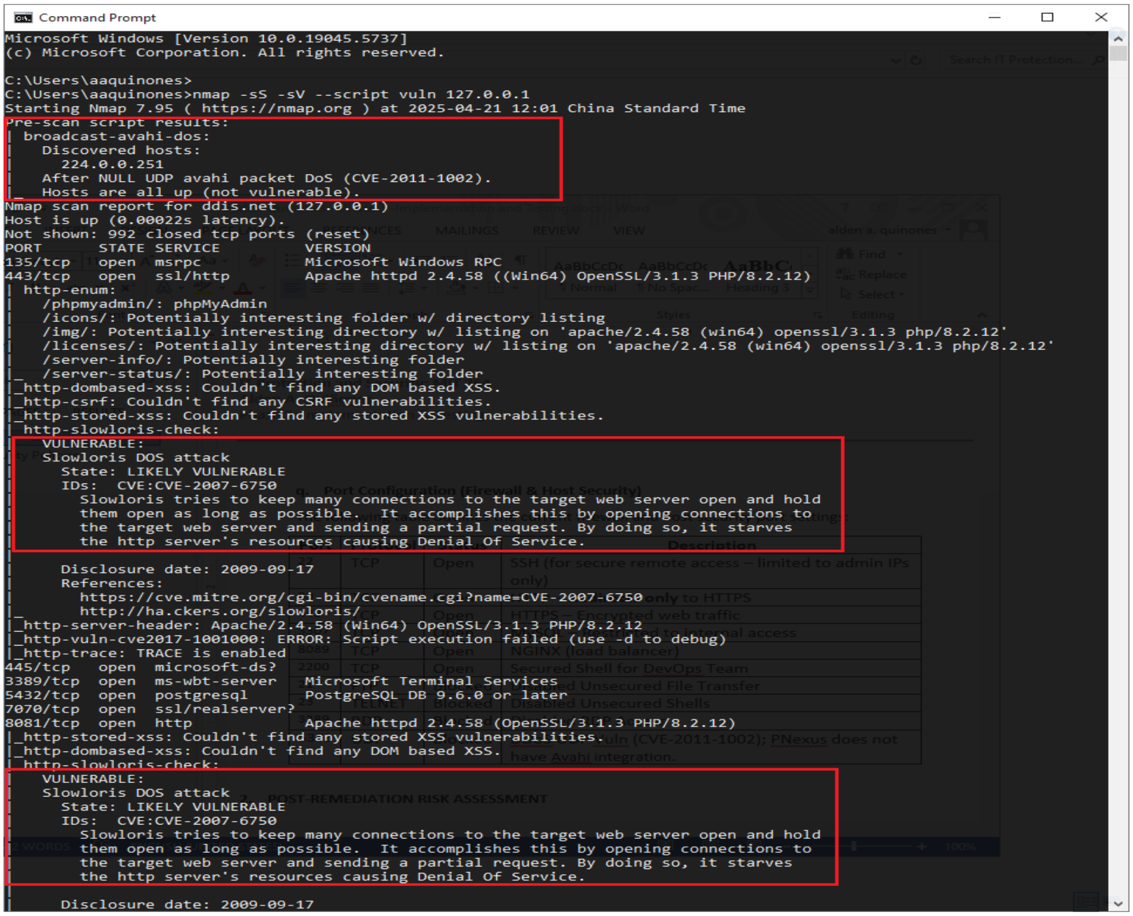
The following table outlines the current firewall and host security port settings:

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.

2. POST-REMEDIATION RISK ASSESSMENT

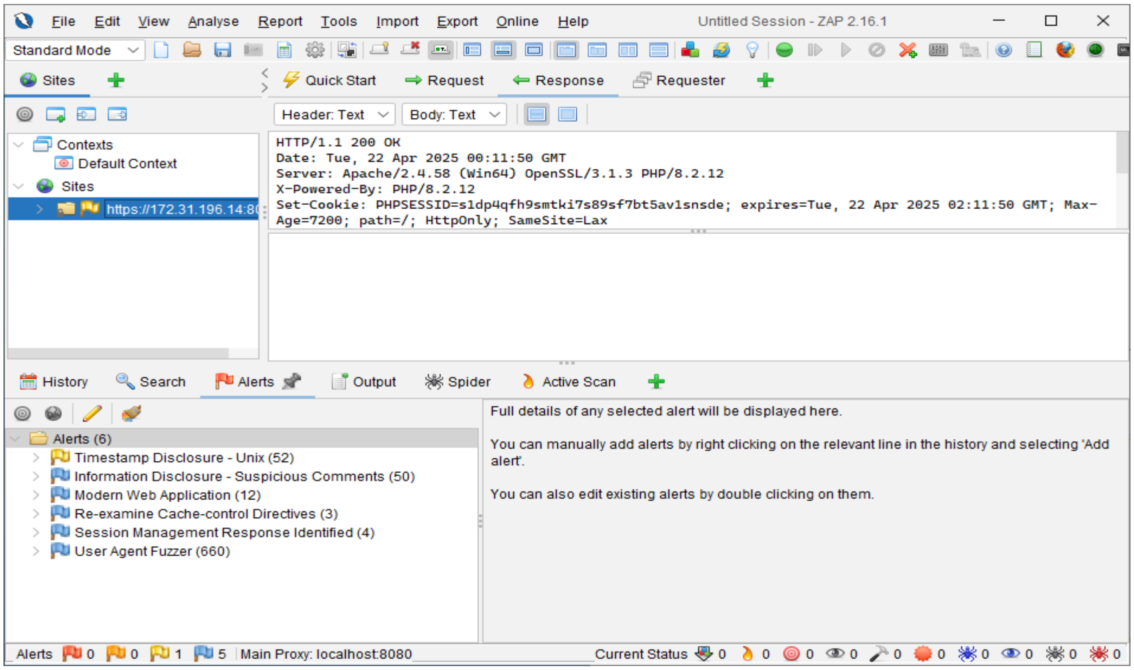
The table below presents the number of security issues before and after applying fixes. All high-risk and six medium-risk issues were successfully resolved. One low-risk and one informational issue remain but are considered tolerable. Moreover, two new medium-risk and four new informational issues were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



```
Microsoft Windows [Version 10.0.19045.5737]
(c) Microsoft Corporation. All rights reserved.

C:\Users\aaquinones>
C:\Users\aaquinones>nmap -sS -sV --script vuln 127.0.0.1
Starting Nmap 7.95 ( https://nmap.org ) at 2025-04-21 12:01 China Standard Time
Pre-scan script results:
| broadcast-avahi-dos:
|   Discovered hosts:
|     224.0.0.251
|   After NULL UDP avahi packet DoS (CVE-2011-1002).
|_  Hosts are all up (not vulnerable).
Nmap scan report for ddis.net (127.0.0.1)
Host is up (0.00022s latency).
Not shown: 992 closed tcp ports (reset)
PORT     STATE SERVICE        VERSION
135/tcp  open  msrpc          Microsoft Windows RPC
443/tcp  open  ssl/http       Apache httpd 2.4.58 ((Win64) OpenSSL/3.1.3 PHP/8.2.12)
| http-enum:
|   /phpmyadmin/: phpMyAdmin
|   /icons/: Potentially interesting folder w/ directory listing
|   /img/: Potentially interesting directory w/ listing on 'apache/2.4.58 (win64) openssl/3.1.3 php/8.2.12'
|   /licenses/: Potentially interesting directory w/ listing on 'apache/2.4.58 (win64) openssl/3.1.3 php/8.2.12'
|   /server-info/: Potentially interesting folder
|_  /server-status/: Potentially interesting folder
|_http-dombased-xss: Couldn't find any DOM based XSS.
|_http-csrf: Couldn't find any CSRF vulnerabilities.
|_http-stored-xss: Couldn't find any stored XSS vulnerabilities.
| http-slowloris-check:
|   VULNERABLE:
|   Slowloris DOS attack
|     State: LIKELY VULNERABLE
|     IDs:  CVE:CVE-2007-6750
|       Slowloris tries to keep many connections to the target web server open and hold
|       them open as long as possible.  It accomplishes this by opening connections to
|       the target web server and sending a partial request. By doing so, it starves
|       the http server's resources causing Denial Of Service.
|
|     Disclosure date: 2009-09-17
|     References:
|       https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-6750
|_      http://ha.ckers.org/slowloris/
|_http-server-header: Apache/2.4.58 (Win64) OpenSSL/3.1.3 PHP/8.2.12
|_http-vuln-cve2017-1001000: ERROR: Script execution failed (use -d to debug)
|_http-trace: TRACE is enabled
445/tcp  open  microsoft-ds?
3389/tcp open  ms-wbt-server  Microsoft Terminal Services
5432/tcp open  postgresql     PostgreSQL DB 9.6.0 or later
7070/tcp open  ssl/realserver?
8081/tcp open  http           Apache httpd 2.4.58 (OpenSSL/3.1.3 PHP/8.2.12)
|_http-stored-xss: Couldn't find any stored XSS vulnerabilities.
|_http-dombased-xss: Couldn't find any DOM based XSS.
| http-slowloris-check:
|   VULNERABLE:
|   Slowloris DOS attack
|     State: LIKELY VULNERABLE
|     IDs:  CVE:CVE-2007-6750
|       Slowloris tries to keep many connections to the target web server open and hold
|       them open as long as possible.  It accomplishes this by opening connections to
|       the target web server and sending a partial request. By doing so, it starves
|       the http server's resources causing Denial Of Service.
|
|     Disclosure date: 2009-09-17
```



```
HTTP/1.1 200 OK
Date: Tue, 22 Apr 2025 00:11:50 GMT
Server: Apache/2.4.58 (Win64) OpenSSL/3.1.3 PHP/8.2.12
X-Powered-By: PHP/8.2.12
Set-Cookie: PHPSESSID=s1dp4qfh9smtki7s89sf7bt5av1snsde; expires=Tue, 22 Apr 2025 02:11:50 GMT; Max-
Age=7200; path=/; HttpOnly; SameSite=Lax
```

Alerts (6)
- Timestamp Disclosure - Unix (52)
- Information Disclosure - Suspicious Comments (50)
- Modern Web Application (12)
- Re-examine Cache-control Directives (3)
- Session Management Response Identified (4)
- User Agent Fuzzer (660)

Full details of any selected alert will be displayed here.

You can manually add alerts by right clicking on the relevant line in the history and selecting 'Add alert'.

You can also edit existing alerts by double clicking on them.

Alerts 0 0 1 5 Main Proxy: localhost:8080 Current Status 0 0 0 0 0 0 0 0 0 0

3. 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	- 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

b. 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).