Comprehensive Web Application Security Assessment Report

Prepared For: Connected Pakistan

Dated: July 17, 2025

Prepared By: Haroon Allahdad (Auditor)

1. Executive Summary

- This report presents a comprehensive assessment of the web application security posture of Connected Pakistan's primary web assets. The assessment leveraged a multi-tool approach, incorporating findings from Nikto, Nessus, a custom web pentesting application, and OWASP ZAP.
- The findings indicate several critical and high-risk vulnerabilities, alongside numerous medium, low, and informational findings. Key areas of concern include:
  - **Critical Data Leakage:** Direct exposure of Personally Identifiable Information (PII), including a credit card number, in web responses.
  - **Widespread CSRF Vulnerabilities:** A significant number of forms lack Anti-CSRF tokens, making the application highly susceptible to Cross-Site Request Forgery attacks.
  - **Missing Essential Security Headers:** Critical headers such as Content Security Policy (CSP), X-Frame-Options (for clickjacking protection), and Strict-Transport-Security (HSTS) are either entirely absent or improperly configured across various web assets, significantly increasing the risk of client-side attacks and downgrade attacks.
  - **Weak Cryptographic Configurations:** SSH services are configured with deprecated CBC mode ciphers and weak key exchange algorithms. Several SSL/TLS services advertise discouraged cipher suites and use certificates signed with weak hashing algorithms (SHA-1).
  - **Information Disclosure:** Various instances of sensitive information disclosure were identified, including robots.txt entries, sitemap.xml content, directory listings, server version banners, and Unix timestamps.

  - **WordPress-Specific Vulnerabilities:** Identification of WordPress installation with exposed default files and potential for version enumeration.
- Addressing these vulnerabilities is paramount to safeguarding Connected Pakistan's data, maintaining user trust, and ensuring regulatory compliance. Immediate action is recommended for critical and high-risk findings, followed by a systematic remediation plan for all identified weaknesses.

2. Introduction

2.1. Purpose of the Assessment

The purpose of this comprehensive web application security assessment is to identify, analyze, and report on security vulnerabilities and weaknesses present within Connected Pakistan's web application infrastructure. This assessment aims to provide actionable recommendations to enhance the overall security posture, reduce the attack surface, and mitigate potential risks associated with web-based threats.

2.2. Scope of the Assessment

The scope of this assessment primarily focused on the following web assets associated with Connected Pakistan:

- **Primary Domain:** https://connectedpakistan.pk/
- **Associated Subdomains/Ports:** Services running on ehostpk.net (IP: 5.9.177.100), including administrative interfaces (cPanel, WHM, Webmail) and mail services.
- **Web Application Components:** WordPress installation, underlying web server configurations, and custom application logic (as identified by the custom pentesting tool).

2.3. Methodology

A multi-faceted approach was employed to ensure comprehensive coverage, combining automated scanning with specialized tools:

Automated Vulnerability Scanning:

- **Nessus:** Utilized for comprehensive network and web application vulnerability scanning, including host-level and service-level checks, cryptographic configurations, and compliance.
- **Nikto:** Employed for web server and web application enumeration, focusing on common misconfigurations, outdated software, and known vulnerabilities.
- **OWASP ZAP (Zed Attack Proxy):** Used for dynamic application security testing (DAST), identifying vulnerabilities such as injection flaws, broken authentication, and security misconfigurations by actively interacting with the web application.
- **Custom Pentesting Application:** A specialized application which incorporates custom tools which are in-house developed Python scripts designed for reconnaissance (OSINT lookups), device/network scanning (port scanning, network discovery), and web application vulnerability scanning for comprehensive security assessments. It was used to identify unique vulnerabilities not typically detected by generic scanners.
- **Information Gathering and Analysis:** Manual review of scan outputs, header analysis, and reconnaissance data to correlate findings and assess their cumulative impact.

3. Web Application Security Findings

The findings are categorized by severity (Critical, High, Medium, Low, Informational) and grouped by the tool that primarily identified them, with cross-references where multiple tools reported the same issue.

3.1. Critical Risk Vulnerabilities

3.1.1. PII Disclosure (CWE-359)

- **Source:** OWASP ZAP
- **Description:** The web application is disclosing Personally Identifiable Information (PII) in its responses. Specifically, a Mastercard credit card number

(5108880199188264) along with its associated Bank Identification Number (BIN), Brand, and Issuer details were found.

- **Affected URL:** https://connectedpakistan.pk/cp-alumni/
- **Instances:** 1
- **Impact:** This is a severe data breach vulnerability. Exposure of credit card information can lead to immediate financial fraud, identity theft, and significant legal and reputational damage for Connected Pakistan. This constitutes a direct violation of data privacy principles and potentially compliance regulations (e.g., PCI DSS if cardholder data is being processed).
- **Recommendations:**
- **Immediate Action:**
  - Investigate the source of this PII disclosure. This data should never be present in unencrypted or unmasked form in web responses.
  - Implement strict data handling policies to ensure sensitive information is never leaked.
  - Review all data flows and ensure that credit card numbers and other PII are properly masked, encrypted, or tokenized at all stages, and only transmitted over secure channels when absolutely necessary.

3.2. High Risk Vulnerabilities

3.2.1. Absence of Anti-CSRF Tokens (CWE-352)

- **Source:** OWASP ZAP
- **Description:** Multiple HTML submission forms across the application lack proper Anti-CSRF tokens. This vulnerability allows an attacker to craft a malicious request that, when triggered by an authenticated user, can perform actions on the user's behalf without their consent.
- **Affected URLs (Examples):** https://connectedpakistan.pk/become-a-member/, https://connectedpakistan.pk/blogstory/, https://connectedpakistan.pk/contact-us/, and 66 other instances on various event/form pages.

- **Instances:** 69
- **Impact:** High risk of unauthorized actions, including data modification, account takeover, or arbitrary content submission, if an authenticated user is tricked into clicking a malicious link. This can severely compromise data integrity and user trust.
- **Recommendations:**
  - Implement a robust Anti-CSRF mechanism for all state-changing forms and requests.
  - Generate unique, unpredictable, and cryptographically strong tokens for each user session and embed them in forms.
  - Ensure tokens are validated on the server-side for every sensitive request.
  - Avoid using GET requests for any state-changing operations.

3.2.2. Content Security Policy (CSP) Header Not Set

- **Source:** OWASP ZAP, Nikto (implied by missing headers)
- **Description:** The Content Security Policy (CSP) HTTP header is missing from web responses. CSP is a crucial security mechanism that helps detect and mitigate attacks such as Cross-Site Scripting (XSS) and data injection by allowing web developers to control which resources (JavaScript, CSS, images, etc.) a user agent is allowed to load.
- **Affected URLs (Examples):** https://connectedpakistan.pk/, https://connectedpakistan.pk/1-on-1-consultation-with-syed-arsalan-ali-shah/, and 461 other instances.
- **Instances:** 463
- **Impact:** Without a CSP, the application is highly susceptible to XSS attacks. Attackers can inject malicious scripts, deface the website, steal user session cookies, or redirect users to phishing sites. This significantly compromises data confidentiality and integrity.
- **Recommendations:**

- Implement a strong Content Security Policy (CSP) header. Define trusted sources for all types of content (scripts, styles, images, fonts, etc.).
- Start with a Content-Security-Policy-Report-Only header to monitor violations before enforcing the policy.
- Gradually refine the CSP to Content-Security-Policy with a strict whitelist of allowed sources.

3.3. Medium Risk Vulnerabilities

3.3.1. Missing Anti-clickjacking Header (X-Frame-Options)

- **Source:** OWASP ZAP, Nikto
- **Description:** The X-Frame-Options HTTP header is missing from web responses. This header prevents clickjacking attacks, where a malicious site embeds the target website in an invisible iframe to trick users into clicking on hidden elements.
- **Affected URLs (Examples):** https://connectedpakistan.pk/, and 460 other instances.
- **Instances:** 461
- **Impact:** Users can be tricked into performing unintended actions (e.g., clicking buttons, submitting forms) on the legitimate site while believing they are interacting with another site. This can lead to unauthorized transactions or data manipulation.
- **Recommendations:**
  - Implement the X-Frame-Options header on all web pages.
  - Set it to DENY if the page should never be framed.
  - Set it to SAMEORIGIN if the page can only be framed by pages from the same origin.
  - Alternatively, use the frame-ancestors directive within a Content Security Policy.

3.3.2. Cross-Domain Misconfiguration (CORS/Other Policy)

- **Source:** OWASP ZAP
- **Description:** A cross-domain misconfiguration was identified, specifically related to a JavaScript file loaded from an external domain. This could indicate overly permissive Cross-Origin Resource Sharing (CORS) policies or other cross-site policy issues.
- **Affected URL:** https://www.winaffiliatepro.com/wp-content/plugins/wp-affiliate-platform/js/jquery.dataTables.min.js
- **Instances:** 1
- **Impact:** Potential for unauthorized access to resources or data leakage if the cross-domain policy is not properly restricted. Malicious websites could potentially interact with Connected Pakistan's resources.
- **Recommendations:**
  - Review and correctly configure Cross-Origin Resource Sharing (CORS) policies on the web server.
  - Ensure that Access-Control-Allow-Origin headers are set only for explicitly trusted domains.
  - If the resource is not intended for cross-domain access, restrict it accordingly.

3.3.3. HSTS Missing From HTTPS Server (RFC 6797)

- **Source:** Nessus, Nikto, OWASP ZAP
- **Description:** The remote web server is not enforcing HTTP Strict Transport Security (HSTS). HSTS is a security mechanism that forces user agents (browsers) to interact with the server only over secure HTTPS connections.
- **Affected Ports/URLs:** tcp/2083/www (cPanel), tcp/443/www (main website), and numerous pages on https://connectedpakistan.pk/.
- **Instances:** 3518 (ZAP), multiple (Nessus)
- **Impact:** This absence allows downgrade attacks (where attackers force a connection over insecure HTTP) and SSL-stripping man-in-the-middle attacks, and weakens cookie-hijacking protections. Users can initially connect over HTTP before being redirected to HTTPS, creating a window for attack.

- **Recommendations:**
  - Configure the web server to send the Strict-Transport-Security header with an appropriate max-age directive (e.g., max-age=31536000 for one year) and includeSubDomains if applicable.
  - Ensure all subdomains also support HTTPS before implementing includeSubDomains.
  - Consider submitting the domain to the HSTS preload list for maximum protection.

3.4. Low Risk Vulnerabilities

3.4.1. SSH Server CBC Mode Ciphers Enabled (CVE-2008-5161)

- **Source:** Nessus
- **Description:** The SSH server is configured to support Cipher Block Chaining (CBC) encryption modes (aes128-cbc, aes256-cbc). While not as severe as ECB, CBC mode can be vulnerable to plaintext recovery attacks under certain conditions (e.g., if padding oracles exist).
- **Affected Port:** tcp/22/ssh
- **Impact:** Potential for an attacker to recover plaintext messages from encrypted SSH traffic, compromising confidentiality.
- **Recommendations:**
  - Disable CBC mode ciphers on the SSH server.
  - Prioritize and enable stronger, authenticated encryption modes like CTR (Counter Mode) or GCM (Galois/Counter Mode).

3.4.2. SSH Weak Key Exchange Algorithms Enabled (RFC 9142)

- **Source:** Nessus
- **Description:** The remote SSH server is configured to allow weak key exchange algorithms, specifically diffie-hellman-group-exchange-sha1. These algorithms are considered cryptographically weak and are not recommended.
- **Affected Port:** tcp/22/ssh

- **Impact:** Weakens the key exchange process, potentially allowing sophisticated attackers to compromise the initial key agreement and decrypt SSH sessions.
- **Recommendations:**
  - Disable all weak key exchange algorithms on the SSH server.
  - Configure the SSH server to use strong, modern key exchange algorithms such as curve25519-sha256, diffie-hellman-group14-sha256, ecdh-sha2-nistp256, or stronger.

3.4.3. Cookie No HttpOnly Flag

- **Source:** OWASP ZAP
- **Description:** Several cookies are missing the HttpOnly flag. When this flag is not set, client-side scripts (e.g., JavaScript) can access these cookies.
- **Affected URLs (Examples):** https://connectedpakistan.pk/, https://connectedpakistan.pk/wp-comments-post.php
- **Instances:** 4
- **Impact:** If an XSS vulnerability exists on the website, an attacker could exploit it to steal user session cookies, leading to session hijacking and unauthorized access to user accounts.
- **Recommendations:**
  - Set the HttpOnly flag for all cookies that do not need to be accessed by client-side scripts, especially session cookies and those containing sensitive information.

3.4.4. Cookie without SameSite Attribute

- **Source:** OWASP ZAP
- **Description:** Several cookies are missing the SameSite attribute. This attribute helps protect against Cross-Site Request Forgery (CSRF) attacks by controlling when cookies are sent with cross-site requests.
- **Affected URLs (Examples):** https://connectedpakistan.pk/, https://connectedpakistan.pk/wp-comments-post.php
- **Instances:** 4

- **Impact:** Weakens the application's defense against CSRF attacks, particularly when combined with the absence of Anti-CSRF tokens.
- **Recommendations:**
  - Implement the SameSite attribute for all cookies.
  - Consider SameSite=Lax as a default for most cookies, and SameSite=Strict for cookies associated with highly sensitive operations.

3.4.5. Cross-Domain JavaScript Source File Inclusion (CWE-829)

- **Source:** Custom Pentest App
- **Description:** The web application includes JavaScript files from external, third-party domains. While common for analytics or CDNs, this introduces a supply chain risk if the external domain is compromised.
- **Affected URLs (Examples):** https://stats.wp.com/e-202529.js, https://connect.facebook.net/signals/config/21235123123.js, https://www.google.com/recaptcha/api.js
- **Instances:** 462
- **Impact:** If a third-party script source is compromised, an attacker could inject malicious code into Connected Pakistan's website, leading to XSS, data theft, or other client-side attacks.
- **Recommendations:**
  - Thoroughly vet all third-party script providers.
  - Where possible, host critical JavaScript files locally.
  - For external scripts, implement Subresource Integrity (SRI) to ensure that the fetched resource has not been tampered with.
  - Ensure that Content Security Policy (CSP) is configured to only allow scripts from explicitly trusted domains.

3.4.6. Timestamp Disclosure - Unix

- **Source:** Custom Pentest App
- **Description:** Unix timestamps are being disclosed in URLs or web content.

- **Affected URLs (Examples):** https://connectedpakistan.pk/wp-content/cache/wpo-minify/1752648739/assets/...
- **Instances:** 597
- **Impact:** While generally low risk, this information disclosure can provide attackers with insights into file creation/modification times or internal system processes, potentially aiding in further reconnaissance or targeted attacks.
- **Recommendations:**
  - Review the necessity of exposing raw Unix timestamps in public-facing URLs or content.
  - If timestamps are required, consider obfuscating or hashing them to prevent direct interpretation.

3.4.7. X-Content-Type-Options Header Missing

- **Source:** Nikto, OWASP ZAP
- **Description:** The X-Content-Type-Options HTTP header is missing from web responses. This header prevents browsers from "sniffing" the content type of a response, which can lead to MIME-sniffing attacks.
- **Affected URLs (Examples):** Numerous https://connectedpakistan.pk/ pages, including images and JS files.
- **Instances:** 2863
- **Impact:** Allows older versions of Internet Explorer and Chrome to interpret content as a different MIME type than declared, potentially leading to XSS or other client-side injection attacks if an attacker can upload a malicious file with a misleading content type.
- **Recommendations:**
- **Implement the X-Content-Type-Options:** nosniff header for all web pages and static assets.

3.5. Informational Risk Findings

3.5.1. Web Server robots.txt Information Disclosure

- **Source:** Nessus, Nikto

- **Description:** The robots.txt file is publicly accessible and contains entries that specify directories and files intended to be disallowed for web crawlers. While standard practice, a malicious user can use this information to discover sensitive or interesting paths.
- **Affected Paths:** /*.html, /*.shtml, /*.htm, /cgi-bin/ (on tcp/443/www); / (on tcp/2083/www).
- **Impact:** Provides attackers with a roadmap of potentially interesting areas to investigate, even if they are not directly accessible.
- **Recommendations:**
  - Review the contents of robots.txt to ensure no truly sensitive or exploitable paths are inadvertently disclosed.
  - For sensitive material, rely on proper access controls (authentication, authorization) rather than robots.txt directives.
  - Consider using X-Robots-Tag HTTP header for more granular control over indexing.

3.5.2. Web Application Cookies Are Expired

- **Source:** Nessus
- **Description:** Several HTTP cookies (e.g., roundcube_sessid, cprelogin, PPA_ID, roundcube_sessauth) are being set with an Expires attribute in the past (e.g., Thu, 01-Jan-1970 00:00:01 GMT). This means these cookies are immediately removed by the browser.
- **Affected Ports/URLs:** tcp/80/www, tcp/443/www, tcp/2083/www.
- **Impact:** While not a direct vulnerability, this indicates a misconfiguration in cookie management. If these cookies are intended to persist or hold session information, their immediate expiry can lead to functional issues or unexpected behavior. If sensitive data is briefly stored, it's immediately purged, which is a positive side effect, but the underlying intent should be clarified.
- **Recommendations:**
  - Review the purpose and intended lifespan of these cookies.

- Configure the Expires attribute correctly based on the cookie's function (e.g., set a future date for persistent cookies, or remove the Expires attribute for session cookies).

3.5.3. Web Server No 404 Error Code Check

- **Source:** Nessus
- **Description:** The web server does not consistently return 404 Not Found HTTP status codes for non-existent files. Instead, it sometimes returns HTTP code 301 (Moved Permanently). This can hinder automated scanning tools and potentially mask the true status of requested resources.
- **Affected Ports/URLs:** tcp/80/www, tcp/443/www, tcp/2083/www.
- **Impact:** Can lead to inaccurate scan results and makes it harder for automated tools to distinguish between non-existent content and redirected content, potentially prolonging reconnaissance for attackers.
- **Recommendations:**
  - Configure the web server to return standard HTTP 404 Not Found responses for all non-existent resources.

3.5.4. SSL Certificate 'commonName' Mismatch

- **Source:** Nessus
- **Description:** The commonName (CN) attribute in the SSL certificate presented by the FTP service (server.domaincontrol.pk) does not match the hostname being accessed (connectedpakistan.pk).
- **Affected Port:** tcp/21/ftp
- **Impact:** While often just a warning for browsers, it can lead to user confusion, trust issues, and in some cases, could be indicative of a misconfigured load balancer or a man-in-the-middle attempt if not properly managed.
- **Recommendations:**
  - Ensure that the SSL certificate used for the FTP service includes connectedpakistan.pk in its Common Name (CN) or Subject Alternative Names (SANs) list.

3.5.5. SSL Certificate Chain Contains Certificates Expiring Soon / SSL Certificate Expiry - Future Expiry

- **Source:** Nessus
- **Description:** Multiple SSL certificates in the chain are expiring soon (within 60 days).
- **Affected Ports:** tcp/110/pop3, tcp/143/imap, tcp/993/imap, tcp/995/pop3, tcp/2083/www.
- **Output:** Certificate CN=connectedpakistan.pk expires Aug 27 16:54:18 2025 GMT.
- **Impact:** Failure to renew these certificates before their expiration date will result in service outages, browser warnings for users, and a complete breakdown of secure communication, leading to denial of service and loss of trust.
- **Recommendations:**
  - Initiate the renewal process for all expiring SSL certificates immediately.
  - Implement an automated certificate management system or a robust process to track and renew certificates well in advance of their expiry dates.

3.5.6. SSL Certificate Signed Using Weak Hashing Algorithm (Known CA)

- **Source:** Nessus
- **Description:** A root CA certificate in the SSL certificate chain (specifically "AAA Certificate Services") is signed using a cryptographically weak hashing algorithm (SHA-1) and is valid until 2028.
- **Affected Port:** tcp/21/ftp
- **Impact:** While SHA-1 is formally deprecated for digital signatures, its continued use, especially by a CA, can raise concerns about the overall cryptographic hygiene. Although the security of HMAC does not rely on collision resistance, the presence of SHA-1 in the certificate chain can be flagged by modern security tools and browsers.
- **Recommendations:**

- Migrate to certificates signed with stronger hashing algorithms (e.g., SHA-256 or SHA-384) from trusted Certificate Authorities.
- Ensure all intermediate and root certificates in the chain also use modern cryptographic standards.

3.5.7. SSL/TLS Recommended Cipher Suites

- **Source:** Nessus
- **Description:** Various SSL/TLS services advertise discouraged cipher suites, including older DHE-RSA, RSA-AES, and Camellia-CBC ciphers, which are not part of recommended modern TLSv1.2/TLSv1.3 configurations.
- **Affected Ports:** tcp/21/ftp, tcp/110/pop3, tcp/143/imap, tcp/443/www, tcp/993/imap, tcp/995/pop3, tcp/2083/www.
- **Impact:** While TLSv1.2 and TLSv1.3 are supported, the presence of weaker ciphers can allow attackers to force the use of less secure encryption, potentially leading to easier decryption of traffic.
- **Recommendations:**
  - Configure all SSL/TLS services to support only strong, modern, and recommended cipher suites (e.g., those listed by Mozilla's SSL Configuration Generator for "Modern" compatibility).
  - Disable all weak, deprecated, and less secure cipher suites.

3.5.8. Server Leaks Version Information via "Server" HTTP Response Header Field (CWE-497)

- **Source:** OWASP ZAP
- **Description:** The web/application server is leaking specific version information via the "Server" HTTP response header.
- **Affected URLs (Examples):** https://tracking-protection.cdn.mozilla.net/ads-track-digest256/128.0/1732308867
- **Instances:** 2

- **Impact:** Aids attackers in reconnaissance by providing specific software versions, allowing them to quickly identify known vulnerabilities associated with those versions.
- **Recommendations:**
  - Configure the web server to suppress or generalize the "Server" header to avoid disclosing specific version numbers.

3.5.9. Information Disclosure - Suspicious Comments

- **Source:** OWASP ZAP
- **Description:** Suspicious comments (e.g., containing "admin", "from") were found in the source code of web pages.
- **Affected URLs:** https://connectedpakistan.pk/ (various pages)
- **Instances:** 244
- **Impact:** While not a direct vulnerability, such comments can provide attackers with internal system details, development notes, or hints about sensitive areas, aiding in further attacks.
- **Recommendations:**
  - Review all source code comments and remove any that contain sensitive information or could provide clues to an attacker. Ensure comments are purely for code readability and not for operational notes.

3.5.10. User Controllable HTML Element Attribute (Potential XSS)

- **Source:** OWASP ZAP
- **Description:** A user-controlled value was identified being reflected into an HTML attribute without proper sanitization or encoding.
- **Instances:** 165
- **Impact:** This indicates a potential Cross-Site Scripting (XSS) vulnerability. If an attacker can inject malicious script into an HTML attribute, they could execute arbitrary JavaScript in the user's browser, leading to session hijacking, defacement, or redirection.
- **Recommendations:**

- Implement strict input validation for all user-supplied data.
- Apply context-aware output encoding when reflecting user-controlled data into HTML attributes to neutralize any malicious characters.

3.5.11. Re-examine Cache-control Directives

- **Source:** OWASP ZAP
- **Description:** The Cache-control header is either not set properly or is missing, potentially allowing sensitive content to be cached by browsers or proxies.
- Affected URLs: Various pages on https://connectedpakistan.pk/
- **Instances:** 454
- **Impact:** If sensitive information is cached, it could be retrieved by unauthorized users from shared caches or local browser caches, compromising data confidentiality.
- **Recommendations:**
  - For sensitive content, ensure the Cache-Control HTTP header is set with no-cache, no-store, must-revalidate, private.
  - For static assets intended for caching, use appropriate directives like public, max-age, immutable.

3.5.12. WordPress Specific Files & Installation Confirmed

- **Source:** Nikto
- **Description:** The presence of wp-links-opml.php and license.txt files directly reveals the WordPress installation and its version.
- **Impact:** Provides attackers with specific version information, simplifying the process of finding and exploiting known vulnerabilities in that particular WordPress version, themes, or plugins.
- **Recommendations:**
  - Regularly update WordPress core, themes, and all plugins to their latest versions.
  - Remove or rename wp-links-opml.php and license.txt if they are not essential or if their information disclosure poses a risk.

- Implement WordPress security plugins and hardening best practices (e.g., limiting login attempts, strong user roles, file integrity monitoring).

4. Risk Assessment and Vulnerability Metrics

This section provides an overview of the vulnerability distribution and associated risk levels.

4.1. Vulnerability Summary by Severity

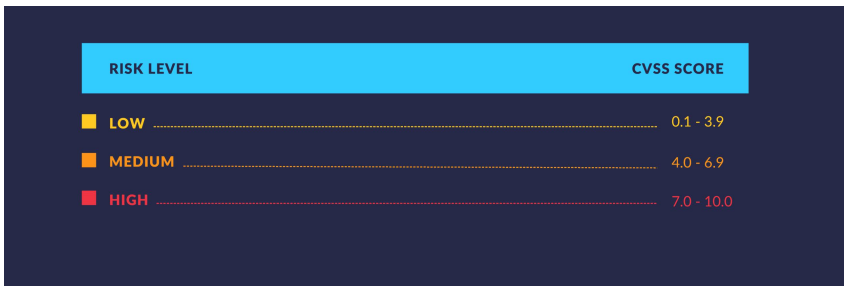
4.2. CVSS (Common Vulnerability Scoring System) Diagram

CVSS is an open framework for communicating the characteristics and impacts of IT vulnerabilities. It provides a numerical score that can be translated into a qualitative representation (low, medium, high, critical).

Rating	CVSS Score
None	0.0
Low	0.1-3.9
Medium	4.0-6.9
High	7.0-8.9
Critical	9.0-10.0

(credit of this diagram belongs to BENQ)

This is one standard and the other is in which White is interchangable to Blue and some systems consider High and Critical as one with red as the label, while Medium Goes as orange, and yellow as Low Risk.



(credit of this diagram goes to NowSecure)

5. Consolidated Recommendations for Web Application Security Enhancement

This section provides a consolidated list of prioritized recommendations to address the identified vulnerabilities.

5.1. Immediate Action (Critical & High Risk)

- **Eliminate PII Disclosure:** Immediately identify and remediate the source of credit card number disclosure on https://connectedpakistan.pk/cp-alumni/. Implement data masking, encryption, or removal of sensitive data from public responses.
- **Implement Anti-CSRF Tokens:** Integrate robust Anti-CSRF token mechanisms into all web forms and state-changing requests to prevent Cross-Site Request Forgery attacks.
- **Enforce Content Security Policy (CSP):** Configure a strict CSP header to whitelist trusted content sources, significantly mitigating XSS and data injection vulnerabilities. Begin with Report-Only mode to refine the policy before full enforcement.
- **Implement HSTS:** Configure the Strict-Transport-Security header with an appropriate max-age for all HTTPS services, including the main website and administrative panels, to enforce secure connections and prevent downgrade attacks.

5.2. High Priority Remediation (Medium Risk)

- **Implement Anti-Clickjacking Headers:** Add X-Frame-Options: DENY or SAMEORIGIN to all web pages to protect against clickjacking attacks.
- **Correct Cross-Domain Configurations:** Review and restrict Cross-Origin Resource Sharing (CORS) policies to only explicitly trusted domains to prevent unauthorized resource access.

5.3. Medium Priority Remediation (Low Risk)

- **Harden SSH Configuration:** Disable CBC mode ciphers and weak key exchange algorithms on the SSH server. Prioritize modern, strong cryptographic algorithms.
- **Secure Cookie Attributes:** Ensure all sensitive cookies have the HttpOnly and SameSite attributes set appropriately (Lax or Strict) to mitigate XSS and CSRF risks.
- **Review Cross-Domain Script Inclusions:** Vet all third-party script providers. Host critical scripts locally where possible, or implement Subresource Integrity (SRI) for external scripts from trusted CDNs.
- **Suppress Server Version Information:** Configure web servers to suppress or generalize the "Server" HTTP response header to avoid disclosing specific version numbers.
- **Address Timestamp Disclosure:** Review the necessity of exposing raw Unix timestamps. If required, consider obfuscating them.
- **Implement X-Content-Type-Options:** Add X-Content-Type-Options: nosniff header to all web responses to prevent MIME-sniffing attacks.

5.4. Ongoing Maintenance & Best Practices (Informational Risk & General)

- **Review robots.txt and 404 Handling:** Audit robots.txt for unintended disclosures and ensure consistent HTTP 404 Not Found responses for non-existent resources.
- **Manage SSL/TLS Certificates:**
  - Address SSL certificate commonName mismatches.
  - Implement a robust certificate lifecycle management process to track and renew certificates well in advance of expiry.
  - Replace certificates signed with weak hashing algorithms (SHA-1) with stronger alternatives.
- **Harden Cipher Suites:** Configure all SSL/TLS services to support only strong, modern, and recommended cipher suites, disabling all weak and deprecated ones.

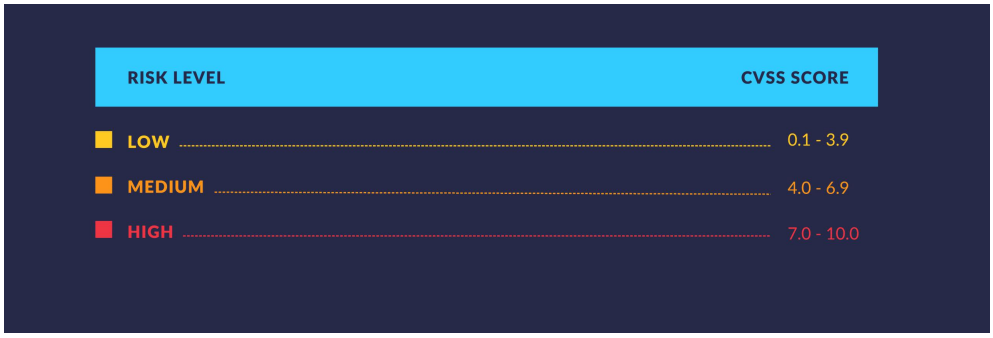
- **WordPress Hardening:** Regularly update WordPress core, themes, and all plugins. Remove or secure wp-links-opml.php and license.txt. Implement WordPress security plugins and hardening best practices.Input Validation and Output Encoding: Continuously enforce comprehensive input validation and context-aware output encoding across the entire application to prevent injection and XSS vulnerabilities.
- **Error Handling:** Configure the application and web server to provide generic error messages to users, avoiding the disclosure of sensitive system information.
- **Implement a Web Application Firewall (WAF):** Consider deploying a WAF to provide an additional layer of protection against common web attacks.
- **Security Development Lifecycle (SDL):** Integrate security best practices into the entire SDLC for any custom application development.

6. Conclusion

- This web application security assessment has identified a range of vulnerabilities, from critical data leakage to various configuration weaknesses and informational disclosures. The presence of PII disclosure, widespread CSRF vulnerabilities, and missing crucial security headers represents significant risks that demand immediate attention.
- By systematically addressing the recommendations outlined in this report, Connected Pakistan can significantly enhance its web application security posture, reduce its attack surface, protect sensitive data, and build greater trust with its users. Continuous monitoring, regular security assessments, and adherence to secure development practices are essential for maintaining a strong security posture in the evolving threat landscape.
- Nonetheless, it was also observed that Connected Pakistan has implemented commendable security mechanisms and procedures, particularly concerning the administration side of their website.

7. Appendices

7.1. Vulnerability Metrics Diagrams



(Copied from NowSecure)

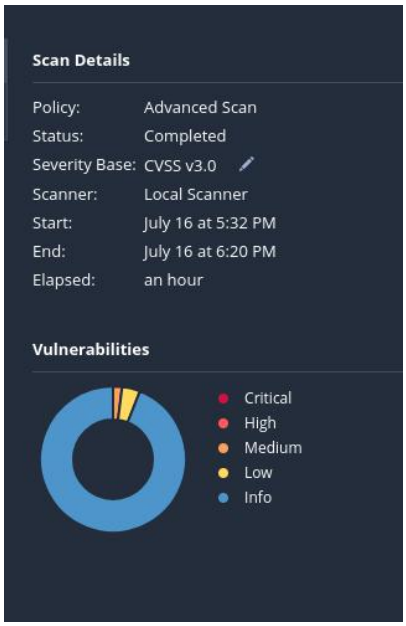
This is the Dictionary Figure of what is the CVSS Score and Color Relation to the Risk (Likelihood of an Attack x Impact of an attack).

Figure 7.1.1: Vulnerability Distribution by Severity (Zaproxy Chart)

Risk Level	Number of Alerts
High	1
Medium	4
Low	7
Informational	8

This is a chart showing the count of Critical, High, Medium, Low, and Informational vulnerabilities identified in Connected Pakistan's Official Website.

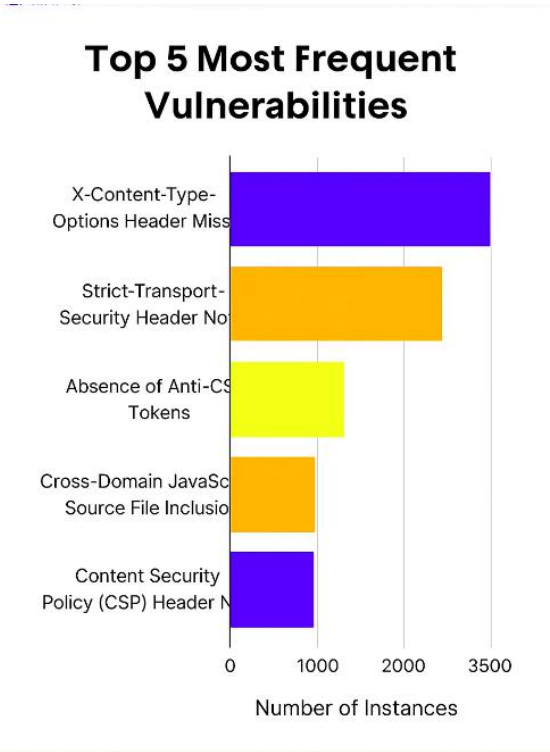
Figure 7.1.2: Vulnerability Distribution by CVSS Base Score (Pie Chart)



94% Informational CVSS Risk (No Risk), 4% Low CVSS Risks, 2% Medium CVSS Risks

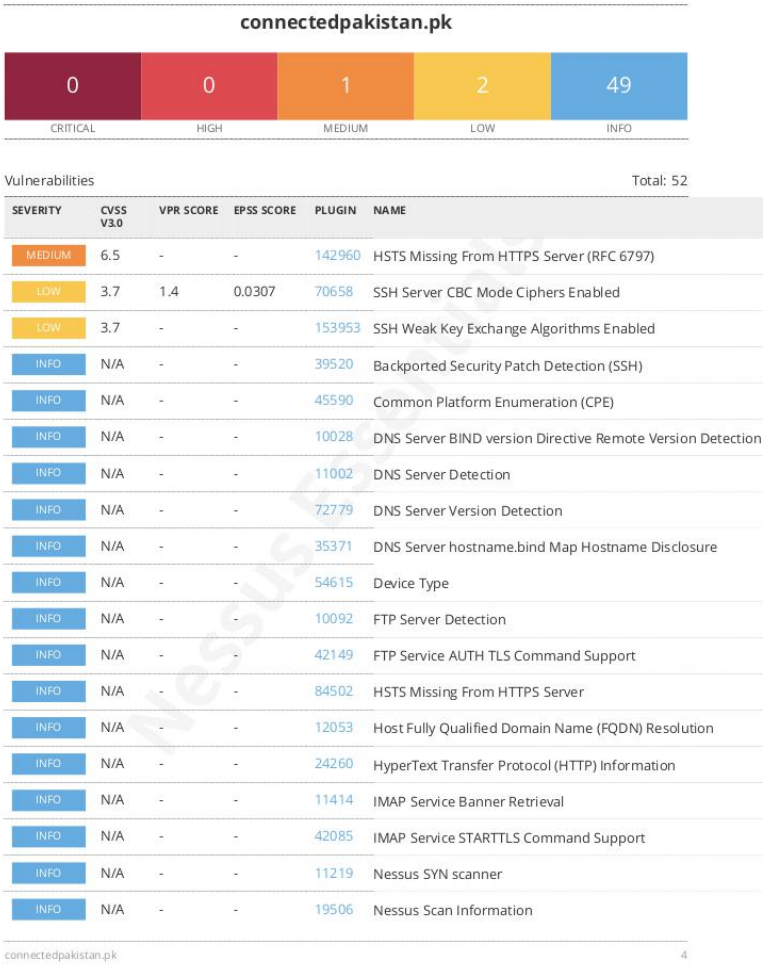
This is basically a pie chart illustrating the percentage of vulnerabilities falling into different CVSS Base Score ranges (e.g., Low: 0.0-3.9, Medium: 4.0-6.9, High: 7.0-8.9, Critical: 9.0-10.0).

Figure 7.1.3: Top 5 Most Frequent Vulnerabilities (Bar Chart)



This is a bar chart showing the count of instances for the most frequently occurring vulnerabilities. It also shows their risk, by which the client company (Connected Pakistan can decide, which vulnerability to recover first).

Figure 7.1.4: Nessus Vulnerability Summary



connectedpakistan.pk

0	0	1	2	49
CRITICAL	HIGH	MEDIUM	LOW	INFO

Vulnerabilities Total: 52

SEVERITY	CVSS V3.0	VPR SCORE	EPSS SCORE	PLUGIN	NAME
MEDIUM	6.5	-	-	142960	HSTS Missing From HTTPS Server (RFC 6797)
LOW	3.7	1.4	0.0307	70658	SSH Server CBC Mode Ciphers Enabled
LOW	3.7	-	-	153953	SSH Weak Key Exchange Algorithms Enabled
INFO	N/A	-	-	39520	Backported Security Patch Detection (SSH)
INFO	N/A	-	-	45590	Common Platform Enumeration (CPE)
INFO	N/A	-	-	10028	DNS Server BIND version Directive Remote Version Detection
INFO	N/A	-	-	11002	DNS Server Detection
INFO	N/A	-	-	72779	DNS Server Version Detection
INFO	N/A	-	-	35371	DNS Server hostname.bind Map Hostname Disclosure
INFO	N/A	-	-	54615	Device Type
INFO	N/A	-	-	10092	FTP Server Detection
INFO	N/A	-	-	42149	FTP Service AUTH TLS Command Support
INFO	N/A	-	-	84502	HSTS Missing From HTTPS Server
INFO	N/A	-	-	12053	Host Fully Qualified Domain Name (FQDN) Resolution
INFO	N/A	-	-	24260	HyperText Transfer Protocol (HTTP) Information
INFO	N/A	-	-	11414	IMAP Service Banner Retrieval
INFO	N/A	-	-	42085	IMAP Service STARTTLS Command Support
INFO	N/A	-	-	11219	Nessus SYN scanner
INFO	N/A	-	-	19506	Nessus Scan Information

connectedpakistan.pk 4

A high-level summary diagram from the Nessus report.

7.2. Raw Scan Outputs (External Reference)

Full raw outputs from the following tools are available for detailed review and are referenced in this report:

Nessus Scan Report:

[https://drive.google.com/file/d/1xLf8nI7WsEaXiAR8JsZWxiAiO8BUFk_8/view?usp=sharing]

Nikto Scan Output:

[https://drive.google.com/file/d/19x30A1B7E64i0GcA8hUu7cOiV2jk14c7/view?usp=sharing]

OWASP ZAP Report:

[https://drive.google.com/file/d/1JKbSVhOPJoNOYmKnqOW32OJBgrpkZe1j/view?usp=sharing]

Custom Web Pentesting Application Logs:

[https://drive.google.com/file/d/1xLf8nI7WsEaXiAR8JsZWxiAiO8BUFk_8/view?usp=sharing]