PATCHLY

SECURITY ASSESSMENT

Penetration Testing Report

PREPARED FOR

VulnWeb

REPORT DATE

March 03, 2026

CONFIDENTIAL

Table of Contents

Executive Summary

Patchly AI conducted an External Attack Surface Assessment of VulnWeb's vulnweb.com domain on March 03, 2026. The assessment encompassed comprehensive web application security testing, DNS and email security configuration analysis, and SSL/TLS implementation review across the discovered infrastructure. During reconnaissance, the assessment identified 20 subdomains, 7 live hosts across 3 unique IP addresses, and 8 open ports, with some infrastructure sharing hosting resources. This broad attack surface provided multiple entry points for evaluation of the organization's external security posture.

The assessment uncovered significant security vulnerabilities requiring immediate attention. Four critical instances of SQL injection vulnerabilities were identified, representing the most severe findings that could allow attackers to extract, modify, or delete sensitive database information. Additionally, four high-severity findings were discovered, including two local file inclusion vulnerabilities that could expose sensitive system files, a publicly accessible WordPress database backup file, and the complete absence of DMARC email authentication. The web application layer demonstrated substantial weaknesses, with six instances of reflected cross-site scripting that could be leveraged for credential theft or session hijacking, alongside open redirect vulnerabilities that attackers could weaponize in phishing campaigns. Database dump files and exposed PHPinfo pages further compound the risk by providing attackers with detailed system configuration information.

Beyond application vulnerabilities, the assessment revealed systemic infrastructure weaknesses that amplify organizational risk. Seven subdomains completely lack SSL/TLS encryption, transmitting all data in cleartext and exposing user credentials and sensitive information to interception. Email security configurations present substantial gaps, with no DMARC record implemented, leaving the organization vulnerable to email spoofing and domain impersonation attacks. The absence of DKIM records and a permissive SPF soft fail policy further diminish email authentication capabilities. These email security deficiencies create significant exposure to business email compromise attacks and reputational damage from attackers impersonating the organization.

Immediate remediation priorities should focus on patching the four SQL injection vulnerabilities and implementing input validation across all web applications, followed by deploying SSL/TLS certificates across all subdomains to ensure encrypted communications. Establishing comprehensive DMARC, DKIM, and SPF policies will substantially reduce email-based threat vectors. The organization should conduct code-level security reviews of web applications, implement a secure software development lifecycle, and establish continuous vulnerability management processes to prevent recurrence of these fundamental security weaknesses.

Finding Summary

Critical	High	Medium	Low	Informational	Total
4	4	17	4	4	33

Scope & Methodology

This assessment targeted the domain vulnweb.com and its associated infrastructure. Reconnaissance identified 20 subdomains and 7 live hosts. Service discovery found 8 open ports across 3 unique IP addresses.

The assessment utilized an automated multi-phase scanning pipeline including subdomain enumeration (Subfinder), HTTP probing (Httpx), port scanning (Naabu), vulnerability scanning (Nuclei with community and custom templates), and web application scanning (OWASP ZAP). Findings were enriched with CVE, CVSS, and EPSS data from the National Vulnerability Database where applicable.

Discovered Subdomains

Subdomain	Status
antivirus1.vulnweb.com	Inactive
estphp.vulnweb.com	Inactive
odincovo.vulnweb.com	Inactive
phptest.vulnweb.com	Inactive
rest.vulnweb.com	Active
testasp.vulnweb.com	Active
testaspnet.vulnweb.com	Active
testhmtml5.vulnweb.com	Inactive
testhtml5.vulnweb.com	Active
testphp.vulnweb.com	Active
testpphp.vulnweb.com	Inactive
testsp.vulnweb.com	Inactive
tetphp.vulnweb.com	Inactive
u003etestasp.vulnweb.com	Inactive
virus.vulnweb.com	Inactive
viruswall.vulnweb.com	Inactive
vulnweb.com	Active
www.test.php.vulnweb.com	Inactive
www.virus.vulnweb.com	Inactive
www.vulnweb.com	Active

7 of 20 subdomains responded to HTTP probing and were included in active scanning.

Infrastructure Mapping

The following analysis maps discovered hostnames to their underlying IP addresses, revealing shared hosting relationships and the true scope of the infrastructure.

Reconnaissance identified 7 hostnames resolving to 3 unique IP addresses. 2 IP address(es) host multiple services, indicating shared infrastructure where a compromise of one service could impact others on the same server.

IP Address	Hostnames	Open Ports	Notes
44.228.249.3	testhtml5.vulnweb.com testphp.vulnweb.com vulnweb.com www.vulnweb.com	80	Shared hosting (4 hosts)
44.238.29.244	testasp.vulnweb.com testaspnet.vulnweb.com	80	Shared hosting (2 hosts)
18.215.71.186	rest.vulnweb.com	80, 8081	

Technology Profile

The following table summarizes the technology stack detected on each live host. Outdated or end-of-life software is flagged as it may contain known vulnerabilities and no longer receives security patches.

Host	Server	Technologies	Flags
rest.vulnweb.com	Apache/2.4.25 (Debian)	Apache HTTP Server:2.4.25, Debian, PHP:7.1.26	PHP 7.1.26 EOL
testaspnet.vulnweb.com	Microsoft-IIS/8.5	IIS:8.5, Microsoft ASP.NET, Microsoft Visual Studio, Windows Server	IIS 8.5, Microsoft-IIS 8.5 EOL
testasp.vulnweb.com	Microsoft-IIS/8.5	DreamWeaver, IIS:8.5, Microsoft ASP.NET, Windows Server	IIS 8.5, Microsoft-IIS 8.5 EOL
testhtml5.vulnweb.com	nginx/1.19.0	Amazon S3, AWS, AngularJS, Bootstrap:2.3.1, Google Hosted Libraries, Nginx:1.19.0 (+2 more)	AngularJS, jQuery 1.9.1 EOL
testphp.vulnweb.com	nginx/1.19.0	DreamWeaver, Nginx:1.19.0, PHP:5.6.40, Ubuntu	PHP 5.6.40 EOL
vulnweb.com	nginx/1.19.0	Nginx:1.19.0	–
www.vulnweb.com	nginx/1.19.0	Nginx:1.19.0	–

8 end-of-life component(s) detected. These versions no longer receive security updates and should be upgraded as a priority to reduce exposure to known vulnerabilities.

Detailed Findings

Web Application Findings

CRITICAL Error-Based SQL Injection (4 instances)

URL: http://testphp.vulnweb.com/product.php?pic=1' | +3 more instance(s) | Confidence: Medium

Description: Four separate pages on the application were found to be vulnerable to error-based SQL injection, where user-supplied input is directly incorporated into database queries without proper validation or sanitization. By appending a single quote character to parameter values in the URL, the application returned database error messages, confirming that malicious SQL commands could be injected and executed against the backend database. These vulnerabilities were identified across product display, search functionality, product listing, and artist pages, indicating a systemic issue with how user input is handled throughout the application.

Business Risk: An attacker exploiting these SQL injection vulnerabilities could gain unauthorized access to the entire database, allowing them to view, modify, or delete sensitive customer information, financial records, user credentials, and proprietary business data. Beyond data theft, attackers could potentially execute administrative commands on the database server itself, leading to complete system compromise, service disruption, or using the database as a pivot point to attack other internal systems.

Remediation: Immediately implement parameterized queries or prepared statements for all database interactions across the application, ensuring that user input is always treated as data rather than executable code. Review and remediate all four identified vulnerable endpoints as well as conduct a comprehensive code audit to identify and fix any similar patterns throughout the codebase. Deploy a web application firewall as a temporary protective measure while code-level fixes are being implemented, and establish secure coding standards that mandate the use of parameterized queries for all future development. After remediation, conduct thorough security testing to verify that the vulnerabilities have been properly addressed and cannot be exploited through alternative attack vectors.

HIGH LFI Detection – Keyed

URL: http://testasp.vulnweb.com/Templatize.asp?item=../../../../../../../windows/win.ini | Confidence: Medium

Description: A Local File Inclusion (LFI) vulnerability was identified on the target web application, allowing an attacker to access files on the web server that should not be publicly accessible. Testing revealed that the application accepts user-supplied input through the “item” parameter without proper validation, enabling path traversal to read the contents of the Windows system file “win.ini” located outside the web application’s intended directory structure. This finding represents one confirmed instance where sensitive server files can be accessed through manipulation of application parameters.

Business Risk: This vulnerability could allow an attacker to read sensitive configuration files, application source code, database credentials, or other confidential data stored on the web server. Beyond information disclosure, successful exploitation could lead to further system compromise if attackers obtain authentication credentials or discover additional vulnerabilities through access to application code, potentially resulting in data breaches, unauthorized system access, or complete application takeover.

Remediation: Implement strict input validation on the “item” parameter and all other user-controllable inputs that reference file paths, using an allowlist approach that only permits explicitly approved values rather than attempting to filter malicious patterns. Configure the application to use indirect file references such as numeric IDs mapped to legitimate file paths on the server side, preventing users from directly specifying file paths or names. Apply operating system-level restrictions to ensure the web application process runs with minimum necessary privileges and cannot access files outside its designated directory structure. Conduct a comprehensive code review to identify and remediate similar input validation issues throughout the application.

HIGH Local File Inclusion – Windows

URL: http://testasp.vulnweb.com/Templatize.asp?item=../../windows/win.ini | Confidence: Medium

Description: A Local File Inclusion vulnerability was identified on the target web application, specifically affecting the Templatize.asp page. This vulnerability allows an attacker to manipulate file path parameters to access arbitrary files on the Windows server, including sensitive system files like win.ini. The application fails to properly validate or sanitize user input used in file operations, permitting directory traversal sequences that navigate outside the intended web directory.

Business Risk: An attacker exploiting this vulnerability could read sensitive configuration files, source code, database credentials, and other confidential information stored on the server’s file system. This unauthorized access to internal files could lead to further system compromise, data breaches, exposure of intellectual property, and potential escalation to complete server takeover if credentials or other security information are discovered in accessible files.

Remediation: Implement strict input validation that rejects any user input containing directory traversal sequences such as “../” or encoded variants. Use a whitelist approach to define exactly which files or templates users are permitted to access, and reject any requests outside this approved list. Replace any direct file path operations with indirect references, such as mapping user input to predefined file identifiers rather than accepting file paths directly. Additionally, configure the web application to run with minimal file system permissions, ensuring it can only access files strictly necessary for its operation, and conduct a thorough code review of all file handling operations throughout the application.

HIGH WordPress Database Backup File – Exposure

URL: http://rest.vulnweb.com/db.sql | Confidence: High | CWE-200 | CVSS: 7.5

Description: A publicly accessible WordPress database backup file was discovered at http://rest.vulnweb.com/db.sql. This file contains a complete export of the website’s database, including user credentials, email addresses, private content, and other sensitive information that should never be accessible to unauthorized individuals. The backup file was left in a location where any internet user could download it without authentication.

Business Risk: An attacker who downloads this database backup gains immediate access to user passwords (which may be reusable across other systems), email addresses for phishing campaigns, and potentially confidential business data stored in the WordPress installation. This exposure could lead to account takeovers, data breaches affecting customers or employees, regulatory compliance violations under data protection laws, and significant reputational damage to the organization.

Remediation: Immediately remove the exposed database backup file from the web-accessible directory and verify that no other backup files exist in publicly accessible locations. Implement a policy requiring all database backups to be stored outside the web root directory, ideally in a separate backup system with appropriate

access controls. Reset passwords for all user accounts in the affected WordPress installation, as the exposed credentials must be considered compromised. Configure your web server to deny access to common backup file extensions such as .sql, .sql.gz, .bak, and .dump through directives in your web server configuration or .htaccess file.

MEDIUM MySQL – Dump Files

URL: http://rest.vulnweb.com/db.sql | Confidence: Medium | CWE-200 | CVSS: 5.3

Description: A MySQL database dump file was discovered publicly accessible at http://rest.vulnweb.com/db.sql. Database dump files typically contain complete exports of database contents, including table structures, configuration data, and potentially sensitive information such as user credentials, customer records, or proprietary business data. This file should not be accessible to unauthorized users on the public internet.

Business Risk: An attacker who accesses this dump file could extract sensitive information including passwords, email addresses, personal identifiable information, and internal system details. This exposure could lead to unauthorized access to user accounts, privacy violations requiring regulatory disclosure, reputational damage, and potential compliance penalties under data protection regulations such as GDPR or CCPA.

Remediation: Immediately remove the db.sql file from the web-accessible directory and verify that no other database dump files are publicly exposed. Implement access controls to ensure that backup files and database exports are stored in protected directories outside the web root with appropriate file permissions. Review the contents of the exposed dump file to determine what data was accessible and assess whether any credentials need to be rotated or if breach notification procedures need to be initiated. Establish a formal backup policy that defines secure storage locations for database dumps and regularly audit web directories to prevent similar exposures in the future.

MEDIUM Open Redirect Bypass

URL: http://testasp.vulnweb.com/Logout.asp?RetURL=//oast.me | Confidence: Medium

Description: An open redirect vulnerability was identified on the logout page of the web application, where the RetURL parameter can be manipulated to redirect users to arbitrary external domains. This bypass technique uses a double-slash notation (//oast.me) that successfully circumvents any basic filtering mechanisms that may have been implemented to prevent standard open redirect attacks. The application trusts user-supplied input in the redirect parameter without proper validation, allowing an attacker to craft malicious links that appear to originate from the legitimate domain.

Business Risk: This vulnerability enables attackers to create authentic-looking phishing campaigns by leveraging your organization’s trusted domain to redirect users to malicious websites. Attackers could craft convincing emails or messages that include links to your legitimate site, which then automatically redirect victims to fake login pages designed to harvest credentials, distribute malware, or conduct other social engineering attacks that exploit the trust users place in your domain.

Remediation: Implement strict validation on all redirect parameters by maintaining an allowlist of permitted internal redirect destinations rather than attempting to block malicious patterns. If external redirects are a necessary business requirement, display an interstitial warning page that clearly shows the destination URL and requires explicit user consent before proceeding to the external site. Additionally, consider using indirect references such as numeric IDs mapped to approved URLs on the server side, which eliminates the ability for users to specify arbitrary destinations. Review all other redirect functionality throughout the application to ensure consistent protection against similar bypass techniques.

MEDIUM Open Redirect Detection

URL: http://testasp.vulnweb.com/Logout.asp?RetURL=https://oast.me | Confidence: Medium

Description: An open redirect vulnerability was identified on the application's logout functionality at http://testasp.vulnweb.com/Logout.asp. The application accepts a return URL parameter (RetURL) and redirects users to the specified destination without validating whether it points to a trusted domain. This allows an attacker to craft a legitimate-looking link to your domain that ultimately sends users to a malicious external site.

Business Risk: Attackers can exploit this vulnerability to conduct sophisticated phishing campaigns by creating URLs that appear to originate from your trusted domain but redirect victims to fraudulent websites designed to steal credentials or distribute malware. This can damage your organization's reputation, erode customer trust, and potentially expose users to credential theft or other attacks that appear to be endorsed by your legitimate website.

Remediation: Implement a whitelist of approved redirect destinations and validate all RetURL parameter values against this list before performing any redirection. If the application requires flexibility in redirect destinations, use an indirect reference map where the RetURL parameter accepts only predefined tokens or identifiers that map to approved URLs server-side, rather than accepting arbitrary URLs directly. Additionally, consider displaying an interstitial warning page when redirecting to external domains, clearly informing users they are leaving your site. Review all other redirect functionality throughout the application to ensure similar vulnerabilities are not present elsewhere.

MEDIUM Reflected Cross-Site Scripting (6 instances)

URL: http://testphp.vulnweb.com/artists.php?artist=1'%22%3E%3C75393%3E | +5 more instance(s) | Confidence: Medium

Description: Six instances of reflected cross-site scripting vulnerabilities were identified across multiple pages of the application, including the artists, search, product, and parameter handling pages. These vulnerabilities occur when user-supplied input is immediately returned in the application's response without proper validation or encoding, allowing an attacker to inject malicious scripts that execute in the context of a victim's browser session.

Business Risk: An attacker could exploit these vulnerabilities to craft malicious URLs that, when clicked by authenticated users, execute arbitrary JavaScript code in their browsers to steal session cookies, capture sensitive information like credentials, perform unauthorized actions on behalf of the victim, or redirect users to phishing sites. This could lead to account compromise, data theft, and reputational damage to the organization if customers or employees become victims of targeted attacks.

Remediation: Implement proper output encoding for all user-supplied input that is reflected in HTTP responses, using context-appropriate encoding functions such as HTML entity encoding for content rendered in HTML contexts and JavaScript encoding for data inserted into JavaScript blocks. Apply input validation on the server side to reject or sanitize unexpected characters and patterns in URL parameters before processing. Consider implementing Content Security Policy headers to provide an additional layer of defense that restricts the execution of inline scripts and limits the sources from which scripts can be loaded. Review all affected pages identified in the finding and apply these controls consistently across the entire application to prevent similar vulnerabilities in other locations.

LOW PHPinfo Page – Detect

URL: http://rest.vulnweb.com/info.php | Confidence: Medium | CWE-200

Description: A publicly accessible PHPinfo page was discovered at http://rest.vulnweb.com/info.php that displays detailed configuration information about the web server's PHP environment. This diagnostic page, generated by PHP's phpinfo() function, is typically used by developers during application setup and troubleshooting but should never remain accessible in production environments. The page reveals extensive technical details including installed PHP modules, version numbers, file paths, environment variables, and server configuration settings.

Business Risk: An attacker who accesses this page gains valuable reconnaissance information that can be used to plan more sophisticated attacks against the application and underlying infrastructure. The exposed configuration details help attackers identify outdated software versions with known vulnerabilities, understand the server's architecture, and discover potential security weaknesses in the PHP configuration that could be exploited to compromise the system or access sensitive data.

Remediation: Immediately remove the info.php file from the web server's publicly accessible directories. If PHPinfo functionality is legitimately needed for system administration purposes, move the file outside the web root and implement strict access controls that require authentication and restrict access to authorized system administrators only. Conduct a comprehensive review of all web-accessible directories to identify and remove any other diagnostic, test, or development files that may have been inadvertently left in production. Establish deployment procedures that explicitly prevent diagnostic and development tools from being included in production releases.

INFORMATIONAL Server Version Disclosure

Hosts: rest.vulnweb.com, testasp.vulnweb.com, testaspnet.vulnweb.com, testphp.vulnweb.com | Confidence: High

Description: Multiple web servers return detailed version information in HTTP response headers, including specific software versions such as Apache/2.4.25, Microsoft-IIS/8.5, and nginx/1.19.0. This information assists attackers in identifying known vulnerabilities associated with specific software versions and tailoring exploits accordingly.

Remediation: Configure each web server to suppress or generalize version information in HTTP response headers. For Apache, set ServerTokens to Prod; for Nginx, enable server_tokens off; for IIS, remove the X-Powered-By header and configure URLScan or request filtering to suppress the Server header.

INFORMATIONAL Missing X-Frame-Options Header

Hosts: testphp.vulnweb.com, testasp.vulnweb.com, rest.vulnweb.com | Confidence: High

Description: Several hosts do not set the X-Frame-Options HTTP header, which prevents browsers from rendering the page inside a frame, iframe, or object element. Without this header, the application may be vulnerable to clickjacking attacks where an attacker overlays a transparent frame to trick users into clicking unintended elements.

Remediation: Add the X-Frame-Options header with a value of DENY or SAMEORIGIN to all HTTP responses. Alternatively, implement a Content-Security-Policy header with a frame-ancestors directive for more granular control over framing behavior.

INFORMATIONAL Missing Content-Security-Policy Header

Hosts: All tested hosts | Confidence: High

Description: None of the tested hosts implement a Content-Security-Policy (CSP) header. CSP is a defense-in-depth mechanism that mitigates the impact of cross-site scripting and other injection attacks by restricting the sources from which resources such as scripts, stylesheets, and images can be loaded. The absence of CSP leaves the application reliant solely on input validation and output encoding to prevent injection attacks.

Remediation: Develop and deploy a Content-Security-Policy header tailored to each application's resource requirements. Begin with a report-only policy (Content-Security-Policy-Report-Only) to identify violations without breaking functionality, then transition to an enforced policy once all legitimate resource sources have been whitelisted.

INFORMATIONAL

Missing X-Content-Type-Options Header

Hosts: All tested hosts | Confidence: High

Description: None of the tested hosts set the X-Content-Type-Options: nosniff header. Without this header, browsers may attempt to MIME-sniff the content type of responses, which could allow an attacker to trick the browser into interpreting a non-executable response as executable content, potentially leading to cross-site scripting or other injection attacks.

Remediation: Add the X-Content-Type-Options: nosniff header to all HTTP responses. This is a straightforward configuration change that provides meaningful defense-in-depth against MIME-type confusion attacks with no risk of breaking legitimate application functionality.

SSL/TLS Assessment

SSL/TLS Certificate Issues

MEDIUM **No SSL/TLS Available (rest.vulnweb.com)**

Host: rest.vulnweb.com

Description: The web service at rest.vulnweb.com does not have SSL/TLS encryption enabled, meaning all communication with this server occurs over unencrypted HTTP connections. When users or systems interact with this service, their data travels across the internet in plain text without any cryptographic protection. This represents a fundamental security gap in the application's communication layer.

Business Risk: Without SSL/TLS encryption, any data transmitted to or from this host can be intercepted and read by attackers positioned on the network path, including credentials, session tokens, API keys, and sensitive business data. This exposure could lead to account compromise, unauthorized access to systems, data theft, and potential regulatory compliance violations under standards like PCI DSS, HIPAA, or GDPR that mandate encryption of data in transit.

Remediation: Obtain and install a valid SSL/TLS certificate from a trusted certificate authority, then configure the web server to enable HTTPS on port 443 using modern TLS protocols (TLS 1.2 or higher). Redirect all HTTP traffic to HTTPS to ensure encryption is enforced for all connections. Implement HTTP Strict Transport Security (HSTS) headers to prevent browsers from making insecure connections. Regularly monitor certificate expiration dates and establish an automated renewal process.

MEDIUM **No SSL/TLS Available (testasp.vulnweb.com)**

Host: testasp.vulnweb.com

Description: The domain testasp.vulnweb.com does not have SSL/TLS encryption configured or the secure connection failed during testing. This means that any data transmitted between users and this server travels in plain text over the internet without encryption. Modern web applications are expected to enforce encrypted connections to protect sensitive information during transmission.

Business Risk: Without SSL/TLS encryption, any data transmitted to or from this host can be intercepted and read by attackers positioned on the network path, including credentials, session tokens, API keys, and sensitive business data. This exposure could lead to account compromise, unauthorized access to systems, data theft, and potential regulatory compliance violations under standards like PCI DSS, HIPAA, or GDPR that mandate encryption of data in transit.

Remediation: Obtain and install a valid SSL/TLS certificate from a trusted certificate authority, then configure the web server to enable HTTPS on port 443 using modern TLS protocols (TLS 1.2 or higher). Redirect all HTTP traffic to HTTPS to ensure encryption is enforced for all connections. Implement HTTP Strict Transport Security (HSTS) headers to prevent browsers from making insecure connections. Regularly monitor certificate expiration dates and establish an automated renewal process.

MEDIUM **No SSL/TLS Available (testaspnet.vulnweb.com)**

Host: testaspnet.vulnweb.com

Description: The web server hosting testaspnet.vulnweb.com does not offer SSL/TLS encryption, meaning all communication between users and the server occurs in plain text over unencrypted HTTP connections. When tested, no valid SSL certificate could be detected and the server either failed to establish a secure connection or does not support HTTPS at all.

Business Risk: Without SSL/TLS encryption, any data transmitted to or from this host can be intercepted and read by attackers positioned on the network path, including credentials, session tokens, API keys, and sensitive business data. This exposure could lead to account compromise, unauthorized access to systems, data theft, and potential regulatory compliance violations under standards like PCI DSS, HIPAA, or GDPR that mandate encryption of data in transit.

Remediation: Obtain and install a valid SSL/TLS certificate from a trusted certificate authority, then configure the web server to enable HTTPS on port 443 using modern TLS protocols (TLS 1.2 or higher). Redirect all HTTP traffic to HTTPS to ensure encryption is enforced for all connections. Implement HTTP Strict Transport Security (HSTS) headers to prevent browsers from making insecure connections. Regularly monitor certificate expiration dates and establish an automated renewal process.

MEDIUM No SSL/TLS Available (testhtml5.vulnweb.com)

Host: testhtml5.vulnweb.com

Description: The web server at testhtml5.vulnweb.com does not have SSL/TLS encryption configured, meaning all traffic between users and the server is transmitted in plain text over unencrypted HTTP connections. During security testing, the scanner was unable to establish an encrypted HTTPS connection to this host, confirming that no valid SSL/TLS certificate is present.

Business Risk: Without SSL/TLS encryption, any data transmitted to or from this host can be intercepted and read by attackers positioned on the network path, including credentials, session tokens, API keys, and sensitive business data. This exposure could lead to account compromise, unauthorized access to systems, data theft, and potential regulatory compliance violations under standards like PCI DSS, HIPAA, or GDPR that mandate encryption of data in transit.

Remediation: Obtain and install a valid SSL/TLS certificate from a trusted certificate authority, then configure the web server to enable HTTPS on port 443 using modern TLS protocols (TLS 1.2 or higher). Redirect all HTTP traffic to HTTPS to ensure encryption is enforced for all connections. Implement HTTP Strict Transport Security (HSTS) headers to prevent browsers from making insecure connections. Regularly monitor certificate expiration dates and establish an automated renewal process.

MEDIUM No SSL/TLS Available (testphp.vulnweb.com)

Host: testphp.vulnweb.com

Description: The web application at testphp.vulnweb.com is not configured to use SSL/TLS encryption, meaning all communication between users and the server occurs in plain, unencrypted HTTP. Attempts to establish an encrypted HTTPS connection to this host were unsuccessful.

Business Risk: Without SSL/TLS encryption, any data transmitted to or from this host can be intercepted and read by attackers positioned on the network path, including credentials, session tokens, API keys, and sensitive business data. This exposure could lead to account compromise, unauthorized access to systems, data theft, and potential regulatory compliance violations under standards like PCI DSS, HIPAA, or GDPR that mandate encryption of data in transit.

Remediation: Obtain and install a valid SSL/TLS certificate from a trusted certificate authority, then configure the web server to enable HTTPS on port 443 using modern TLS protocols (TLS 1.2 or higher). Redirect all HTTP traffic to HTTPS to ensure encryption is enforced for all connections. Implement HTTP Strict Transport Security (HSTS) headers to prevent browsers from making insecure connections. Regularly monitor certificate expiration dates and establish an automated renewal process.

MEDIUM No SSL/TLS Available (vulnweb.com)

Host: vulnweb.com

Description: The scanner detected that vulnweb.com does not have SSL/TLS encryption enabled, meaning all communications between users and this website are transmitted in plain text without any cryptographic protection. When users access this site, their web browsers cannot establish a secure HTTPS connection.

Business Risk: Without SSL/TLS encryption, any data transmitted to or from this host can be intercepted and read by attackers positioned on the network path, including credentials, session tokens, API keys, and sensitive business data. This exposure could lead to account compromise, unauthorized access to systems, data theft, and potential regulatory compliance violations under standards like PCI DSS, HIPAA, or GDPR that mandate encryption of data in transit.

Remediation: Obtain and install a valid SSL/TLS certificate from a trusted certificate authority, then configure the web server to enable HTTPS on port 443 using modern TLS protocols (TLS 1.2 or higher). Redirect all HTTP traffic to HTTPS to ensure encryption is enforced for all connections. Implement HTTP Strict Transport Security (HSTS) headers to prevent browsers from making insecure connections. Regularly monitor certificate expiration dates and establish an automated renewal process.

MEDIUM No SSL/TLS Available (www.vulnweb.com)

Host: www.vulnweb.com

Description: The website www.vulnweb.com was found to be accessible without SSL/TLS encryption, meaning all communication between users and the server occurs in plain text over unencrypted HTTP connections. No valid SSL certificate could be detected on this host.

Business Risk: Without SSL/TLS encryption, any data transmitted to or from this host can be intercepted and read by attackers positioned on the network path, including credentials, session tokens, API keys, and sensitive business data. This exposure could lead to account compromise, unauthorized access to systems, data theft, and potential regulatory compliance violations under standards like PCI DSS, HIPAA, or GDPR that mandate encryption of data in transit.

Remediation: Obtain and install a valid SSL/TLS certificate from a trusted certificate authority, then configure the web server to enable HTTPS on port 443 using modern TLS protocols (TLS 1.2 or higher). Redirect all HTTP traffic to HTTPS to ensure encryption is enforced for all connections. Implement HTTP Strict Transport Security (HSTS) headers to prevent browsers from making insecure connections. Regularly monitor certificate expiration dates and establish an automated renewal process.

Certificate Status by Host

Host	Status	Cert Valid	Expiry	Weak Ciphers
rest.vulnweb.com	No TLS	No	–	None
testasp.vulnweb.com	No TLS	No	–	None
testaspnet.vulnweb.com	No TLS	No	–	None
testhtml5.vulnweb.com	No TLS	No	–	None
testphp.vulnweb.com	No TLS	No	–	None
vulnweb.com	No TLS	No	–	None
www.vulnweb.com	No TLS	No	–	None

Status key – A/B: TLS present (letter reflects cipher strength). No TLS: could not establish any TLS connection (genuine gap). → host: domain redirects to HTTPS at that destination (not a finding).

DNS & Email Security

HIGH **No DMARC Record Found – vulnweb.com**

Check: DMARC

The domain vulnweb.com does not have a DMARC (Domain-based Message Authentication, Reporting, and Conformance) record. Without DMARC, there is no policy telling receiving mail servers how to handle emails that fail SPF or DKIM checks. This leaves the domain vulnerable to email spoofing and phishing attacks.

Recommended Fix: Publish a DMARC record. Start with monitoring mode: v=DMARC1; p=none; rua=mailto:dmarc-reports@yourdomain.com; then progressively move to p=quarantine and finally p=reject.

MEDIUM **No DKIM Records Found – vulnweb.com**

Check: DKIM

No DKIM (DomainKeys Identified Mail) records were found for common selectors on vulnweb.com. DKIM provides email authentication by signing outgoing messages with a cryptographic key. Without DKIM, email recipients cannot verify that messages were genuinely sent by your domain and have not been tampered with in transit. Note: DKIM selectors may use non-standard names that were not detected by this scan.

Recommended Fix: Configure DKIM signing for all email sending services. Your email provider will supply the selector name and DNS record to publish.

LOW **No MTA-STS Policy – vulnweb.com**

Check: MX

vulnweb.com does not have an MTA-STS (Mail Transfer Agent Strict Transport Security) policy. MTA-STS prevents TLS downgrade attacks on email delivery by requiring receiving servers to use encrypted connections.

Recommended Fix: Implement MTA-STS by publishing a policy at https://mta-sts.yourdomain.com/.well-known/mta-sts.txt and adding a _mta-sts TXT record.

LOW **DNSSEC Not Enabled – vulnweb.com**

Check: DNSSEC

DNSSEC is not enabled for vulnweb.com. Without DNSSEC, DNS responses can be spoofed through cache poisoning attacks, potentially redirecting users to malicious servers.

Recommended Fix: Enable DNSSEC through your DNS hosting provider. Most modern registrars and DNS providers support DNSSEC with minimal configuration.

LOW **SPF Soft Fail Policy – vulnweb.com**

Check: SPF

The SPF record uses "~all" (soft fail) instead of "-all" (hard fail). Soft fail marks unauthorized emails as suspicious but does not reject them, reducing protection against spoofing.

Recommended Fix: Change "~all" to "-all" once you have confirmed all legitimate senders are included in the SPF record.

Remediation Roadmap

The most urgent remediation priority is addressing the four instances of Error-Based SQL Injection vulnerabilities, which represent critical-severity findings that could allow attackers to extract, modify, or delete sensitive database information. These must be fixed immediately by implementing parameterized queries or prepared statements across all affected database interactions, ensuring that user input is never directly concatenated into SQL commands. Equally critical are the three high-severity file access vulnerabilities: the LFI Detection – Keyed finding, the Local File Inclusion – Windows vulnerability, and the exposed WordPress Database Backup File. The LFI vulnerabilities require immediate input validation and sanitization to prevent directory traversal attacks, implementing whitelist-based file access controls rather than attempting to blacklist dangerous patterns. The WordPress database backup file should be removed from the web-accessible directory immediately and stored securely offline, as this file likely contains credentials, user data, and complete database schema information that could facilitate further attacks. These critical and high-severity items should be resolved within 72 hours given their potential for immediate exploitation.

Within the next 30 days, attention should turn to the medium-severity findings that create substantial risk. The MySQL dump file exposure requires the same treatment as the WordPress backup – immediate removal from public access and implementation of proper backup storage procedures with encryption at rest. The six instances of Reflected Cross-Site Scripting vulnerabilities should be addressed by implementing context-aware output encoding throughout the application, ensuring all user-controllable data is properly sanitized before rendering in HTML, JavaScript, or other contexts. The two open redirect vulnerabilities (including one bypass variant) need remediation through whitelist-based URL validation, preventing attackers from using your domains to phish users or bypass security controls. Most significantly, the complete absence of SSL/TLS across all seven tested domains and subdomains (rest.vulnweb.com, testasp.vulnweb.com, testaspnet.vulnweb.com, testhtml5.vulnweb.com, testphp.vulnweb.com, vulnweb.com, and www.vulnweb.com) represents a systemic encryption failure that exposes all transmitted data to interception. Implement TLS 1.2 or higher across all properties with properly configured certificates, enable HSTS headers to prevent downgrade attacks, and redirect all HTTP traffic to HTTPS endpoints.

The lower-priority items should be addressed within 90 days as part of broader security hardening efforts. The exposed PHPinfo page should be removed from production environments, as it discloses detailed server configuration, installed modules, and potential attack surface information to reconnaissance efforts. While this represents a lower direct risk, it significantly aids attackers in crafting targeted exploits against your environment. This is an appropriate time to conduct a comprehensive review of all publicly accessible files and directories to ensure no other development artifacts, configuration files, or debugging tools remain exposed in production.

Following completion of these remediation efforts, we strongly recommend a follow-up penetration test within 60–90 days to validate that fixes were implemented correctly and have not introduced new vulnerabilities. This retest should particularly focus on verifying that SQL injection and LFI vulnerabilities have been comprehensively addressed across the entire application, not just at the specific endpoints identified in this assessment, and confirm that SSL/TLS implementation follows current best practices with strong cipher suites and proper certificate validation.

Conclusion

The external attack surface assessment of vulnweb.com reveals a security posture requiring immediate attention, with four critical SQL injection vulnerabilities and multiple high-severity exposures that could lead to complete system compromise. The presence of Error-Based SQL Injection across multiple instances represents an urgent threat, as these flaws allow attackers to extract sensitive database contents and potentially gain administrative access. Compounding this risk are the Local File Inclusion vulnerabilities affecting Windows systems, the exposed WordPress Database Backup File containing potentially sensitive credentials and configuration data, and the absence of a DMARC record that leaves the organization vulnerable to email spoofing attacks. While the assessment identified a reasonable attack surface with 20 subdomains and 7 live hosts, the concentration of critical and high-severity findings on production systems indicates gaps in secure development practices and vulnerability management processes.

Immediate remediation of the SQL injection vulnerabilities and local file inclusion flaws must be the top priority, followed by removal of the exposed database backup and implementation of proper email authentication controls. Patchly AI recommends addressing all critical and high-severity findings within 72 hours and conducting a focused re-test to validate remediation efforts. Our team remains available to provide remediation guidance, secure code review support, or additional penetration testing services as VulnWeb works to strengthen their security posture.

Appendix A – Detailed Instance Inventory

The following tables list every unique URL or endpoint affected by each multi-instance finding identified during this assessment. This inventory is provided to support remediation teams in scoping and validating fixes across all affected locations.

[CRITICAL] Error-Based SQL Injection – 4 instances

#	Affected URL / Target
1	http://testphp.vulnweb.com/product.php?pic=1’
2	http://testphp.vulnweb.com/search.php?test=query’
3	http://testphp.vulnweb.com/listproducts.php?artist=1’
4	http://testphp.vulnweb.com/artists.php?artist=1’

[MEDIUM] Reflected Cross-Site Scripting – 6 instances

#	Affected URL / Target
1	http://testphp.vulnweb.com/artists.php?artist=1’%22%3E%3C75393%3E
2	http://testphp.vulnweb.com/hpp/params.php?p=valid’%22%3E%3C75393%3E&pp=12
3	http://testphp.vulnweb.com/search.php?test=query’%22%3E%3C75393%3E
4	http://testphp.vulnweb.com/hpp/?pp=12’%22%3E%3C75393%3E
5	http://testphp.vulnweb.com/product.php?pic=1’%22%3E%3C75393%3E
6	http://testphp.vulnweb.com/listproducts.php?artist=1’%22%3E%3C75393%3E

About Patchly AI

Patchly AI delivers next-generation cybersecurity solutions powered by artificial intelligence, purpose-built for enterprises operating within Microsoft ecosystems. Our platform combines deep technical expertise with advanced AI capabilities to protect organizations against an evolving threat landscape, enabling security teams to move from reactive to proactive defense.

Our Services

Vulnerability Management

Continuous identification, classification, and prioritization of security vulnerabilities using AI-driven analysis and real-world threat intelligence.

Patch Management

Intelligent patch orchestration featuring our proprietary Patch Veracity technology, analyzing real deployment data and community sentiment for confident, risk-aware patching decisions.

Attack Surface Management

Comprehensive discovery and monitoring of your external and internal attack surface, providing continuous visibility into exposed assets and potential entry points.

Penetration Testing

Expert-led security assessments simulating real-world attack scenarios to identify exploitable vulnerabilities, validate controls, and deliver actionable remediation guidance.

Contact Us

www.patchly.ai
info@patchly.ai
St Petersburg, FL, USA