A

Web Vulnerability and Bug Hunting Report

JUNAID QUADRI

quadrijunaid369258@gmail.com

Contents

Executive Summary

The results of a web vulnerability assessment and exploit demonstration conducted as part of the SECT organization's educational cybersecurity internship assignment on the publicly accessible web application [i.e., http://tendermines.com] are presented in this document. Finding, validating, and reporting common web application security flaws, particularly those listed in the OWASP Top 10 vulnerabilities was the main goal.

Assessment Objectives

By deploying common web-based attacks and examining the overall exposure to that exploitation, this assessment will attempt to perform a thorough examination of the domain's web application security posture, specifically http://tendermines.com
Goals:

1. Determine Typical Internet Vulnerabilities
2. Carry out simulated exploit demonstrations
3. Verify and Expand the Data from Reconnaissance (Week 1)
4. Evaluate the Risk Severity and Impact
5. Offer Remedial Suggestions

Scope of Work

The scope of this vulnerability assessment is clearly defined to ensure that testing efforts are structured, targeted, and ethically conducted within agreed-upon boundaries. The main focus is on identifying, simulating, and reporting vulnerabilities in the publicly accessible web infrastructure of the target domain: http://tendermines.com.

3.1 In-Scope Components

The following components are explicitly included within the scope of this assessment:

- Primary Target URL
- Web Pages
- Forms & Inputs
- Subdomains
- Public DNS Records
- Reconnaissance Artifacts

3.2 Out-of-Scope Components

To maintain ethical boundaries and prevent any unintentional disruption to services, the following activities are considered out of scope for this project:

- Denial of Service (DoS) / DDoS attacks
- Phishing/Social Engineering simulations
- Exploit of production systems
- Brute Force or Password Spraying
- Data Deletion or Modification
- Third-party integrations testing

Testing Methodology

Phases	Description	Tools Used in testing
Reconnaissance	Recon is a passive and active information gathering technique.	Google Dorks Sublist3r IntelX
Threat Modelling	It is used identifies attack vectors It is used to identify critical entry points	Manual
Vulnerability Scan	The OWASP Top 10	Burp Suite Firefox Dev Tools
Exploitation	A Safe Proof of Concept attack (i.e. XSS, SQLi)	TryHackMe PortSwigger Labs
Post Exploitation	A privilege escalation Data exposure on darknet	Manual
Risk Reporting	Document the risks with CVSS score and OWASP mapping	NIST/NVD/CVSS method

Techniques Used

Technique	Purpose	Scope
Google Dorking	Find the hidden files Hidden directories Hidden emails	tendermines.com
XSS Payload Testing	Detect any vulnerable script injection point	Search/contact
SQL Injection	Test the login by using SQL query	Login/admin page
Header Inspection	Deeply analyse the response header	Burp Suite Browser Dev Tools
Dark Web Search	Check for any leaked database	intelx.io
Security Header Testing	Check the HSTS, CSP, X-Frame	HTTP response inspection

Tools & Technologies Used

Category	Tool / Platform	Purpose
Reconnaissance	Google Dorking	Discover exposed directories, emails, login panels
	IntelX.io	Search for leaked credentials and dark web traces
	theHarvester	OSINT on emails/domains/subdomains
	Shodan.io	Port scanning and banner grabbing
	Sublist3r	Subdomain enumeration
Vulnerability Testing	Burp Suite Community Edition	Intercept HTTP traffic, test inputs, modify headers
	Firefox Developer Tools	Inspect DOM, test XSS injection points manually
Lab Simulations	The TryHackMe Labs	The XSS Room, and OWASP Top 10 room
	The PortSwigger	XSS & SQLi exploitation lab
	GeeksforGeeks	Understanding SQLi and XSS injection
Passive Analysis	WhatWeb	Web stack fingerprinting
	SSL Labs / SecurityHeaders.io	Header misconfiguration detection
	BuiltWith	Discover backend technologies
DNS/Email Security	MX Toolbox	{SPF, DKIM, DMARC,} and DNS records

Vulnerability Reports

Reflected Cross-Site Scripting (XSS)

By inserting potentially harmful scripts into form inputs, search functionality, and URL parameters, a thorough test was carried out to determine whether the application is susceptible to Reflected or Stored XSS. The application safely filtered or sanitized every tested payload.

Parameter	Details
Vulnerability Type	Reflected Cross-Site Scripting (XSS)
Testing Outcome	No XSS vulnerability found
Status	Not Vulnerable
Severity	N/A
OWASP Category	A03:2021 – Injection
Affected Components	contact form search inputs URL parameters

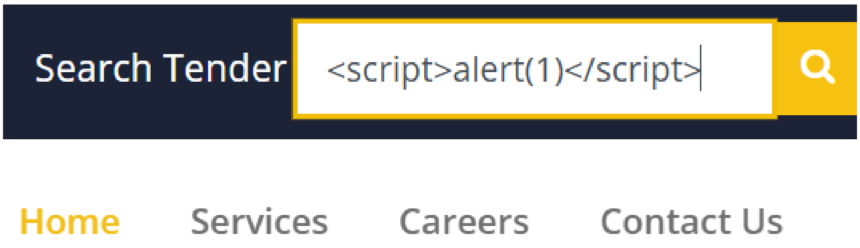
Payloads Tested

Input Vector	Payload Example	Result
/contact?msg=	<script>alert(1)</script>	Filtered → [removed]alert(1)
/keyword (search bar)	<img src=x onerror=alert(1)>	Filtered → <img> only
Contact Form	"><script>alert(1)</script>	Sanitized before rendering
Global	<svg/onload=alert(1)>	Sanitized

Recommendations

Control	Purpose
Input Allowlisting	Continue the strict sanitization of user inputs
Output Encoding	Use a trusted encoding libraries like (e.g., OWASP Java Encoder)
Implement CSP	Add a Content-Security-Policy header so as to restrict executable code
Regular Testing	Maintain a well periodic XSS scanning

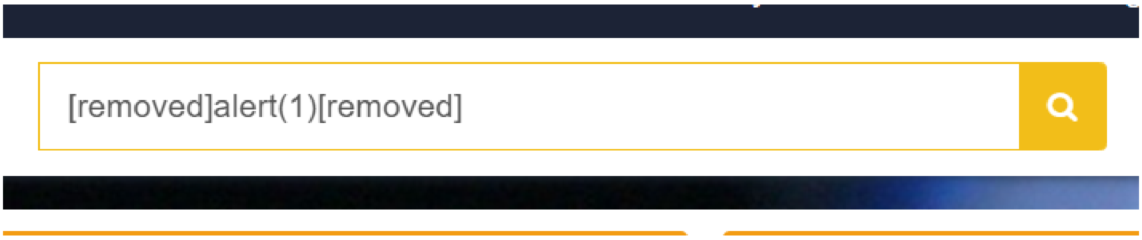
Before:



Search Tender

<script>alert(1)</script>

After:



[removed]alert(1)[removed]

SQL Injection (SQLi)

An error-based SQL injection vulnerability exists in the http://tendermines.com/login login endpoint. When a specially constructed input is entered in the username field, the application returns a raw SQL error message that reveals:

- Database table and column names;
- SQL query logic for the backend
- The server's file paths, which display the WAMP stack

Parameter	Details
Vulnerability Type	SQL Injection (Error-Based)
OWASP Category	A01:2021 – Broken Access Control / Injection
Severity	High
Affected Component	Login Form – username field
Error Triggered	"Mentorname" is an unknown column in "field list."
Database Used	MySQL (inferred from error format and file path)

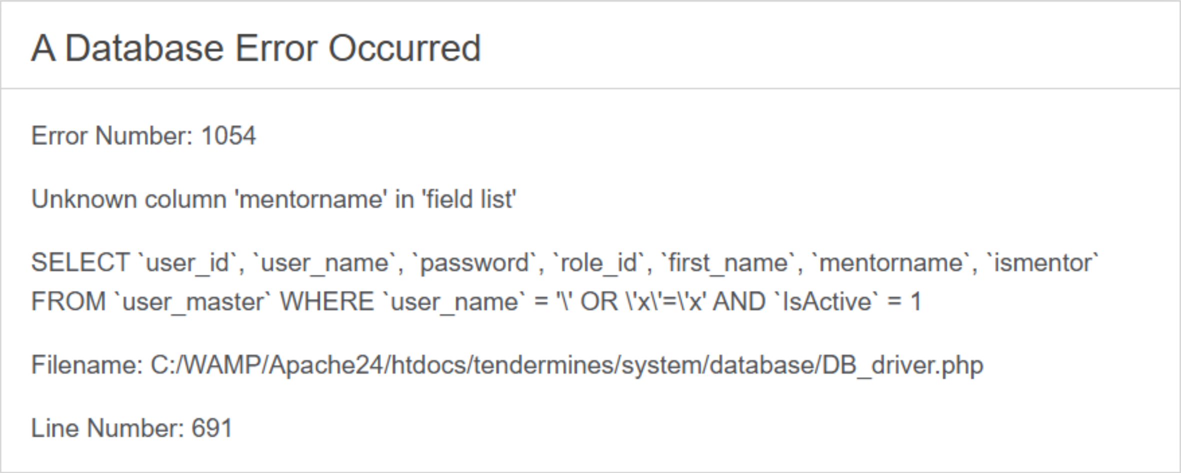
Steps to Reproduce

Step	Action
1	Go to http://tendermines.com/login
2	In Username field, enter: { ' OR 'x'='x }, {' OR 1=1 --}, {' OR '1'='1}, {admin' --}, {" OR ""="}
3	In Password field enter anything, for example enter: password123
4	Submit the form
5	Observe SQL error exposing query and internal structure

Payload Used

' OR 1=1 --
' OR '1'='1
admin' --
" OR ""="
' OR 'x'='x

Screenshot



A Database Error Occurred

```
Error Number: 1054

Unknown column 'mentorname' in 'field list'

SELECT `user_id`, `user_name`, `password`, `role_id`, `first_name`, `mentorname`, `ismentor`
FROM `user_master` WHERE `user_name` = '\' OR \'x\'=\'x' AND `IsActive` = 1

Filename: C:/WAMP/Apache24/htdocs/tendermines/system/database/DB_driver.php

Line Number: 691
```

Missing HTTPS Implementation

The website http://tendermines.com is served entirely over unsecured HTTP, without support for HTTPS (SSL/TLS) encryption. This exposes all traffic between the user and the server to clear-text transmission, which can be intercepted, manipulated, or stolen.

Parameter	Details
Vulnerability Type	An Insecure Transport {Missing HTTPS / TLS Encryption}
OWASP Category	A05:2021 – Security Misconfiguration
Severity	High
Affected Component	Entire Web Application (http://tendermines.com)
Verification Method	Live testing & passive reconnaissance

Steps to Identify

Step	Action
1	Accessed http://tendermines.com via browser
2	No automatic redirection to https://tendermines.com
3	Attempted to load https://tendermines.com → site is unreachable
4	Used tools like curl, Burp, and SecurityHeaders.io
5	No HSTS or TLS-related response headers is being detected

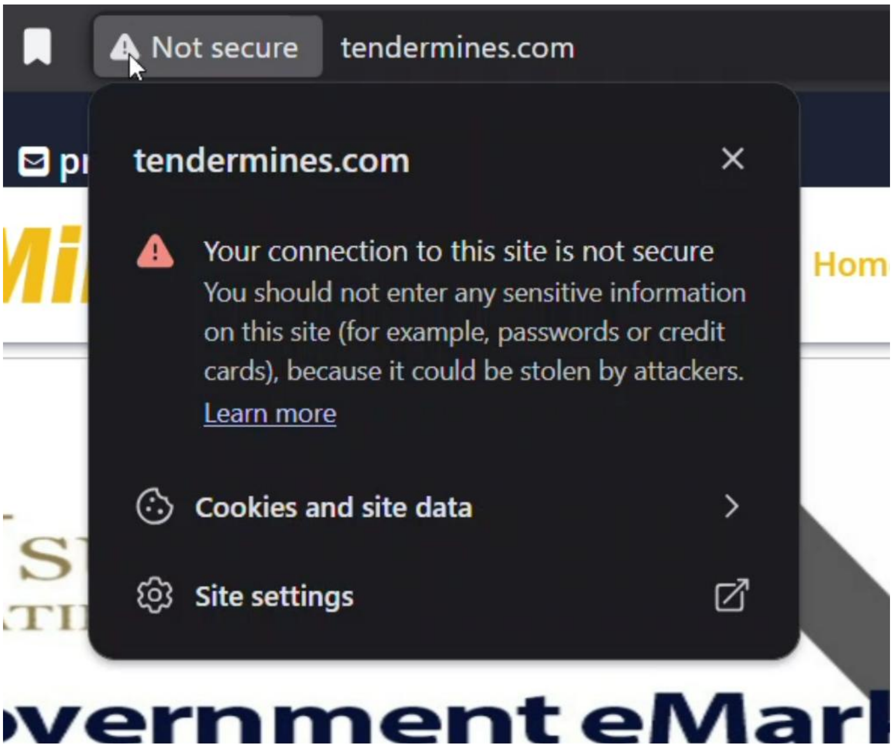
Impact

Threat Scenario	Impact Description
MITM Attacks	An Attackers can intercept or modify login sessions and form submissions
Credential Theft	Many plaintext credentials may be exposed
SEO/Trust Issues	Browsers mark the site as “Not Secure” → Loss of user trust
Cookie Hijacking	Session tokens can be stolen via packet sniffing
No HSTS Header	Clients are unable to enforce secure-only connections

Recommendations

Control	Description
Implement HTTPS	Get an SSL/TLS certificate (for example, through Let's Encrypt).
Redirect HTTP to HTTPS	Use web server config to auto-redirect all http:// to https://
Enforce HSTS	Add Strict-Transport-Security header with proper max-age
Secure Cookies	Set Secure and HttpOnly flags on session cookies
Certificate Renewal Automation	Use certbot or cron-based script to ensure renewal of SSL certs

Screenshot



Not secure tendermines.com

tendermines.com

Your connection to this site is not secure

You should not enter any sensitive information on this site (for example, passwords or credit cards), because it could be stolen by attackers.

Learn more

Email Spoofing via Missing SPF/DKIM/DMARC

The email domain associated with tendermines.com does not implement the standard email authentication records, DKIM (DomainKeys Identified Mail), DMARC (Domain-based Message Authentication, Reporting, and Conformance), and SPF (Sender Policy Framework).

Parameter	Details
Vulnerability Type	Email Spoofing / DNS Misconfiguration
OWASP Category	A05:2021 – Security Misconfiguration
Severity	High
Affected Component	Email Domain – @tendermines.com
Verification Method	Using Google Admin Tools and MXToolbox for passive DNS enumeration

Steps to Identify (for reference read the reconnaissance report)

Step	Tool Used	Result
1	MXToolbox	Searched tendermines.com → No SPF, DKIM, or DMARC found
2	Google Admin Toolbox	Verified absence of DNS TXT records for auth headers
3	dig / nslookup	The authoritative servers did not return any TXT/SPF records.

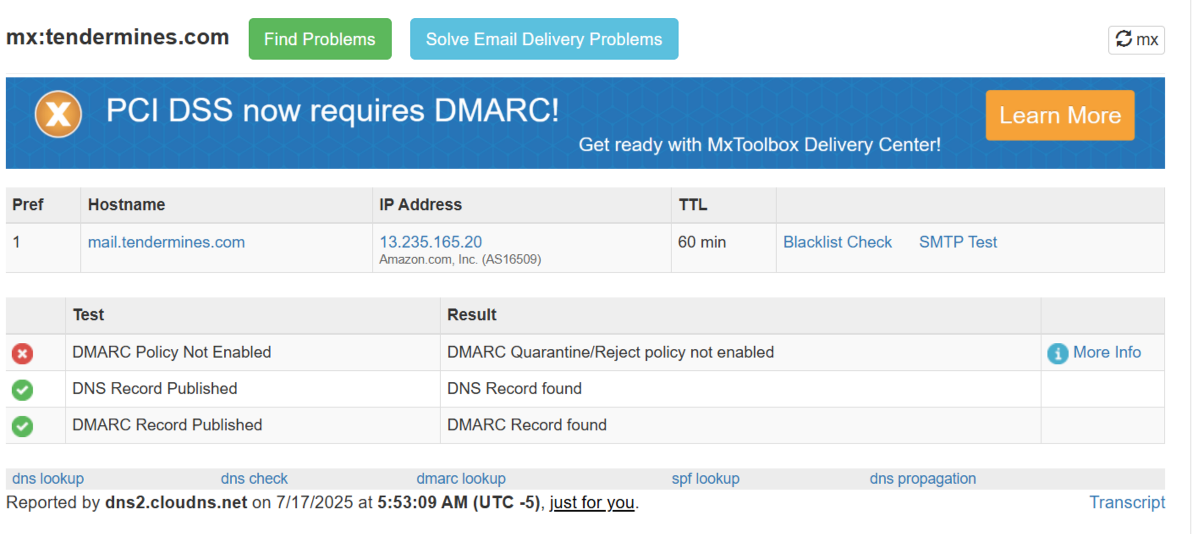
Impact

Risk Vector	Threat Description
Phishing and Social Engineering	Phishing emails posing as executives, administrators, or support
Brand & Reputation Damage	Targeted users may associate scams with the real domain
Mail Delivery Failures	Legitimate mails may be marked as spam due to no validation
No Accountability	Email recipients are unable to confirm who is permitted to send emails

Recommendations

Control	Description
Add SPF Record	Example: v=spf1 include:_spf.google.com ~all
Generate DKIM Keys	Generate domain key pair and add DNS TXT for selector
Create DMARC Policy	Example: v=DMARC1; p=quarantine; rua=mailto:admin@domain.com
Use Email Monitoring Tools	Use aggregate reports to identify spoofing attempts.
Test Regularly	Use MXToolbox, Dmarcian, Google Toolbox for continuous DNS validation

Screenshot


Find Problems
Solve Email Delivery Problems
mx
PCI DSS now requires DMARC!
Get ready with MxToolbox Delivery Center!
Learn More
dns lookup
dns check
dmarc lookup
spf lookup
dns propagation
Transcript

mx:tendermines.com

Pref	Hostname	IP Address	TTL	
1	mail.tendermines.com	13.235.165.20 Amazon.com, Inc. (AS16509)	60 min	Blacklist Check SMTP Test

	Test	Result	
✗	DMARC Policy Not Enabled	DMARC Quarantine/Reject policy not enabled	More Info
✓	DNS Record Published	DNS Record found	
✓	DMARC Record Published	DMARC Record found	

Reported by **dns2.cloudns.net** on 7/17/2025 at **5:53:09 AM (UTC -5)**, just for you.

Sensitive Information Disclosure

During the engagement, multiple instances of unintended information disclosure were identified. Disclosure can be identifies in the screenshot with the line:
Filename: C:/WAMP/Apache24/htdocs/tendermines/system/database/DB_driver.php

Parameter	Details
Vulnerability Type	Information Disclosure / Verbose Error Messages
OWASP Category	A01:2021 – Broken Access Control / A05:2021 – Security Misconfiguration
Severity	Medium to High (based on publicly available data)
Affected Components	Login Page, Contact Form, Server Error Pages
Verification Method	Manual input testing, error monitoring

Steps to Identify

Action Performed	Observation
SQLi Payload: ' OR 'x'='x	MySQL syntax and schema data caused a raw SQL error.
Contact form fuzzing	No sensitive error displayed
Attempting invalid login with payloads	Returned raw database errors with **WAMP path** + **filename** + **line number**

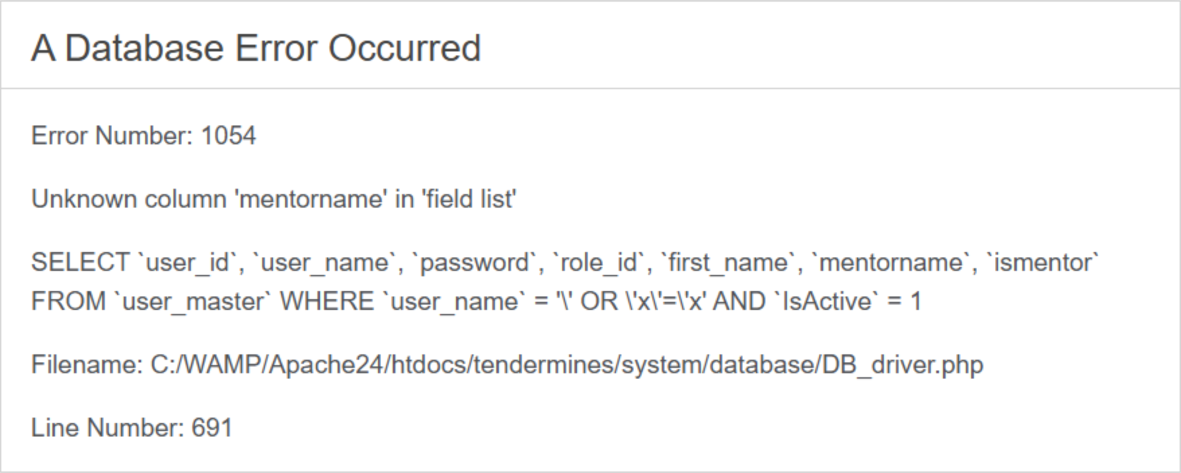
Impact

Threat Scenario	Impact Description
Information Enumeration	Attackers gain knowledge of the internal database structure, tables, and field names.
Stack Fingerprinting	WAMP stack revealed (OS, web server, framework)
Targeted Exploitation	Enables creation of more precise SQLi, LFI, or RCE payloads
Trust and Compliance Issues	violates the standards for a production environment and secure coding.

Recommendations

Remediation Step	Description
Suppress Debug/Error Output	Substitute generic user-facing responses for error messages.
Configure Error Handling	For 400/500 errors, use custom error pages.
Remove Verbose Debug Logs	Disable stack traces, SQL dumps, and file path displays in production
Harden Server Configuration	Prevent exposure of backend paths or debug stack info via Apache/PHP
Use Try-Catch with Logging	Internally log errors but show users generic messages
Conduct Code Review	Ensure exception handling does not expose sensitive data

Screenshot



```
A Database Error Occurred

Error Number: 1054

Unknown column 'mentorname' in 'field list'

SELECT `user_id`, `user_name`, `password`, `role_id`, `first_name`, `mentorname`, `ismentor`
FROM `user_master` WHERE `user_name` = '\' OR \'x\'=\'x' AND `IsActive` = 1

Filename: C:/WAMP/Apache24/htdocs/tendermines/system/database/DB_driver.php

Line Number: 691
```

Admin/Login Pages Indexed

The authentication interface (/login, /admin) was discovered to be openly available and possibly search engine indexed during the evaluation. These pages are specifically used for backend or admin control.

Parameter	Details
Vulnerability Type	Sensitive Page Exposure / Directory Indexing
OWASP Category	A05:2021 – Security Misconfiguration
Severity	Medium
Affected Endpoint	http://tendermines.com/login
Verification Method	Google dorking combined with manual access (site:tendermines.com)

Steps to Identify

Step	Method / Tool	Result
1	Visited http://tendermines.com/login http://tendermines.com/admin	Login form displayed, no CAPTCHA or rate-limiting
2	Ran Google Dork: site:tendermines.com	Page is indexed (or can be, if not restricted)

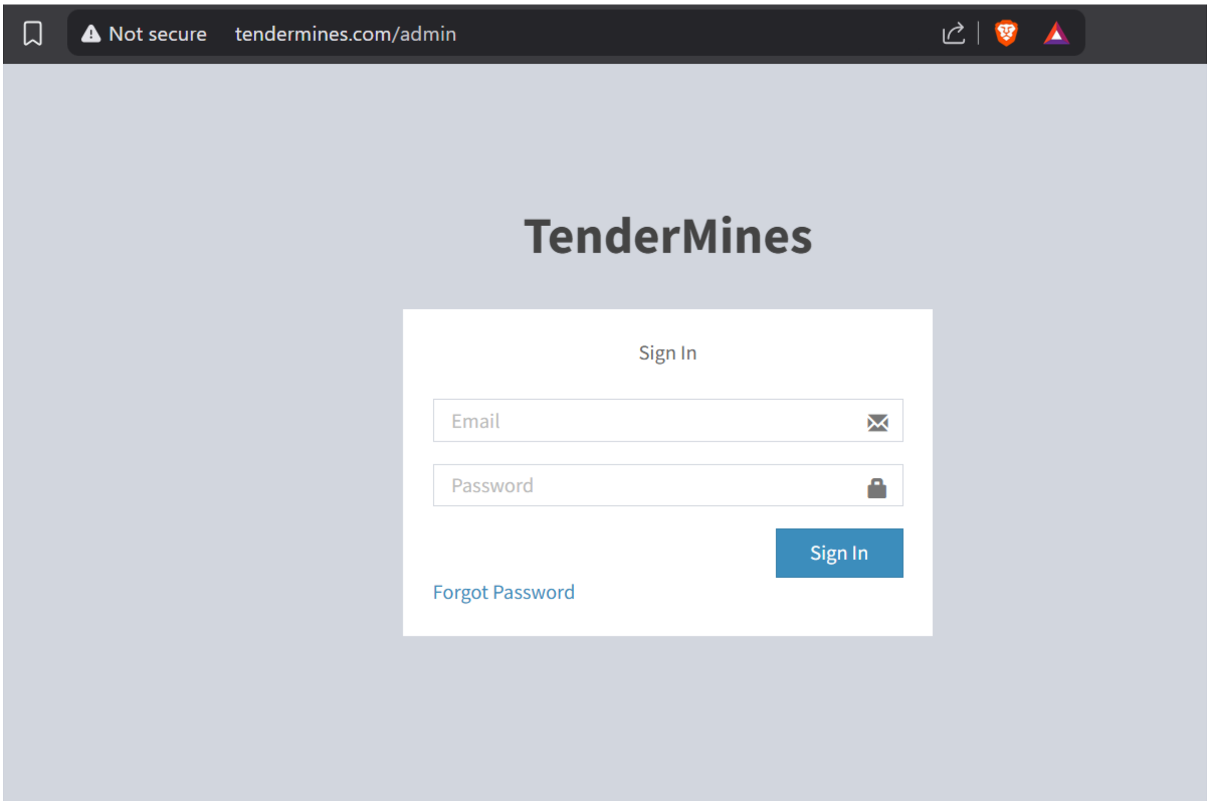
Impact

Threat Scenario	Description
Credential Brute Force	Attackers can script login attempts
Automated Recon Tools	Tools like Shodan, Google Dorks, FOFA can find the page
Account Risk	If the login is weak or is used on multiple portals.
Exploitation Increased	The page could be an injection point for SQLi and XSS.

Recommendations

Control	Description
Restrict Login Interface	Use IP allowlists, VPN, or geofencing for admin portals
Implement robots.txt Rules	Disallow indexing of sensitive URLs
Adding CAPTCHA and Rate Limiting	Use tools like Google reCAPTCHA and Fail2Ban to prevent brute-force attacks.
Monitor Auth Logs	To keep an eye on login trends and irregularities, use WAF or SIEM.
Rename Admin Paths	Use non-default admin paths (e.g., /admin-panel → /secure-area)
Two-Factor Authentication	Enforce MFA to reduce password-only attack surface

Screenshot



Not secure tendermines.com/admin

TenderMines

Sign In

Lack of Security Headers

Several essential HTTP security headers are not implemented by the web application at http://tendermines.com, making users susceptible to a range of browser-based attacks. These headers are necessary to guarantee the security of contemporary browsers.

Parameter	Details
Vulnerability Type	Insecure HTTP Response Headers
OWASP Category	A05:2021 – Security Misconfiguration
Severity	Medium
Affected Pages	Entire application (http://tendermines.com)
Test Methodology	Manual testing + SecurityHeaders.com
Missing Headers	CSP, HSTS, X-Content-Type-Options, Referrer-Policy, Permissions-Policy

Steps to Identify

Step	Tool Used	Result
1	curl -I http://tendermines.com	Manual response inspection showed only X-Frame-Options
2	SecurityHeaders.com	Grade E: The majority of important headers are absent
3	Browser DevTools – Network Tab	No CSP, HSTS, Referrer-Policy, XCTO, or Permissions-Policy

Headers

Header Name	Status	Risk Description
Content-Security-Policy	Missing	Allows inline scripts → XSS risk
Strict-Transport-Security (HSTS)	Missing	No HTTPS → vulnerable to SSL stripping
X-Content-Type-Options	Missing	MIME sniffing risk
Referrer-Policy	Missing	Referrer data leakage possible
Permissions-Policy	Missing	Browser API abuse not restricted
X-Frame-Options	Present	Protects against clickjacking

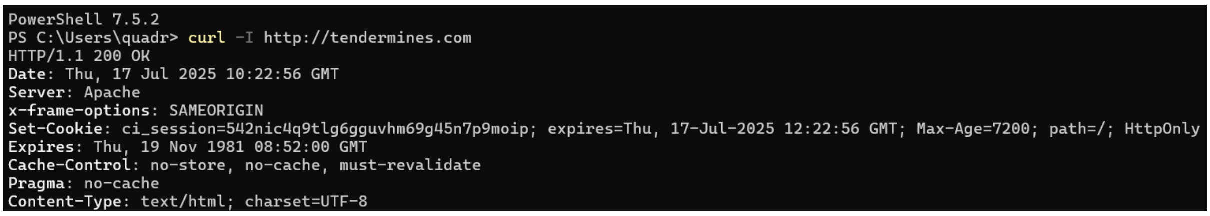
Impact

Attack Scenario	Description
XSS	No CSP → Unsafe scripts are permitted by the browser.
Content Sniffing Attacks	No XCTO → Browser may guess file types
Referrer Leakage	No policy → Full URLs with tokens may be leaked
Browser API Abuse	No Permissions-Policy → APIs like camera, geolocation exposed
Downgrade/No HTTPS	No HSTS → An attacker could easily downgrade to HTTP

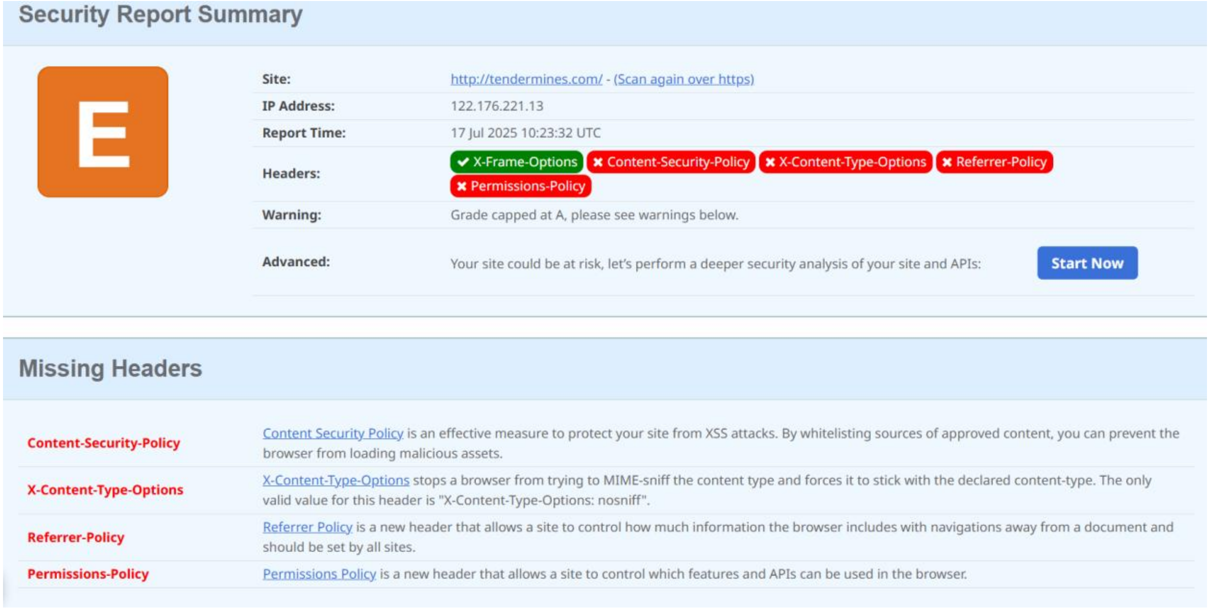
Recommendations

Header	Recommended Configuration Example
Content-Security-Policy	script-src'self' 'unsafe-inline'; object-src 'none'; default-src'self';
Strict-Transport-Security	Strict-Transport-Security: max-age=63072000; includeSubDomains; preload
X-Content-Type-Options	X-Content-Type-Options: nosniff
Referrer-Policy	Referrer-Policy: no-referrer-when-downgrade
Permissions-Policy	Permissions-Policy: camera=(), microphone=(), fullscreen=(self)

Screenshot



```
PowerShell 7.5.2
PS C:\Users\quadr> curl -I http://tendermines.com
HTTP/1.1 200 OK
Date: Thu, 17 Jul 2025 10:22:56 GMT
Server: Apache
x-frame-options: SAMEORIGIN
Set-Cookie: ci_session=542nic4q9tlg6gguvhm69g45n7p9moip; expires=Thu, 17-Jul-2025 12:22:56 GMT; Max-Age=7200; path=/; HttpOnly
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Content-Type: text/html; charset=UTF-8
```



Security Report Summary

Site:	http://tendermines.com/ - (Scan again over https)
IP Address:	122.176.221.13
Report Time:	17 Jul 2025 10:23:32 UTC
Headers:	✓ X-Frame-Options ✗ Content-Security-Policy ✗ X-Content-Type-Options ✗ Referrer-Policy ✗ Permissions-Policy
Warning:	Grade capped at A, please see warnings below.
Advanced:	Your site could be at risk, let's perform a deeper security analysis of your site and APIs: Start Now

Missing Headers

Content-Security-Policy Content Security Policy is an effective measure to protect your site from XSS attacks. By whitelisting sources of approved content, you can prevent the browser from loading malicious assets.

X-Content-Type-Options X-Content-Type-Options stops a browser from trying to MIME-sniff the content type and forces it to stick with the declared content-type. The only valid value for this header is "X-Content-Type-Options: nosniff".

Referrer-Policy Referrer Policy is a new header that allows a site to control how much information the browser includes with navigations away from a document and should be set by all sites.

Permissions-Policy Permissions Policy is a new header that allows a site to control which features and APIs can be used in the browser.

Leaked SQL Database on Dark Web

Dark web intelligence platforms like intelx.io and dark web archives were found to index a SQL database purportedly belonging to tendermines.com.
The SQL dump that was made public seems to include:

- Email addresses and user names
- Hashes of passwords (or plaintext, if stored insecurely)
- Potentially admin data or internal systems

Attribute	Details
Vulnerability Type	Data Breach (Leaked SQL Database)
Category	Information Disclosure / Dark Web Intelligence
Source	Dark Web Repository (intelx.io / OSINT tools)
Severity	Critical
Exposed Asset	SQL dump (potentially containing usernames, passwords, emails)
Detection Date	[Insert actual date from recon scan or report]

Risk Impact

Threat Scenario	Risk Level	Potential Impact
Credential stuffing or account takeover	**High**	Users may reuse credentials elsewhere
Brand/reputation damage	**High**	Public leak harms trust
Insider data exploitation	**High**	Admin/system emails exposed
Phishing/social engineering	**Medium**	Targeted attacks using leaked emails

Recommendations

Control Measure	Description
Confirm the Leak Internally	Determine whether the dump is valid, recent, and whether it came from your systems
Notify Affected Users	Initiate a forced password reset and notify via incident response
Remove Public Exposure	Send DMCA requests to the dump's hosting websites.
Enhance Data Security	Encrypt PII data and apply strict access controls in database management
Enable MFA Across User Base	Reduce impact of reused credentials
Monitor for Future Leaks	Set up breach alerts via HaveIBeenPwned, Dark Web Monitors, etc.

Screenshot

For screenshots refer to reconnaissance_sect.pdf under the Dark Web Filtering section

Risk Impact Assessment

Vulnerability Title	CVSS (Est.)	Risk Level	Affected Component	Impact
Reflected Cross-Site Scripting (XSS)	5.4	Medium	Search and input fields	Medium
SQL Injection	8.6	High	Form for Login (username field)	High
Missing HTTPS Implementation	7.5	High	Entire website	High
Missing SPF/DKIM/DMARC	8.3	High	Email Domain (@tendermines.com)	High
Sensitive Information Disclosure	7.0	High	Login Error Handling	Medium-High
Admin/Login Page Indexed	6.5	Medium	/login URL	Medium
Leaked SQL Database (on Dark Web)	9.0	Critical	External Breach + OSINT	Critical
Missing Security Headers (CSP, HSTS, etc.)	6.3	Medium	Web Server Response	Medium

Recommendations & Mitigation Strategies

Vulnerability Title	Recommended Mitigation Action(s)
Reflected Cross-Site Scripting (XSS)	Sanitize user inputs, implement Content Security Policy (CSP), and use encoding
SQL Injection (Error-Based)	Use parameterized queries / ORM, suppress SQL errors, input validation
Missing the HTTPS Implementation	Install the TLS certificate, change HTTP to HTTPS, and implement the HSTS policy.
Missing SPF/DKIM/DMARC	Configure DNS TXT records for SPF, DKIM, and DMARC
Sensitive Information Disclosure	Replace debug errors with user-friendly messages; remove file path leaks
Admin/Login Page Indexed	Use robots.txt, rename admin path, add IP whitelisting, enable CAPTCHA
Leaked SQL Database (Dark Web)	Rotate credentials, investigate breach scope, alert users, implement WAF
Missing Security Headers	Add CSP, X-Content-Type-Options, HSTS, Referrer-Policy, Permissions-Policy

Conclusion

The evaluation successfully found several security flaws in the target application (http://tendermines.com), ranging from configuration errors and a lack of best practices to high-risk defects like SQL Injection and exposed sensitive data.

The findings highlight the urgent need for:

- Improved input validation process
- A Secure server configuration
- Encryption implementation (HTTPS)
- Email domain hardening
- Vulnerability monitoring

Addressing these issues will significantly improve the overall security posture, reduce attack vulnerability, and ensure adherence to modern web application security standards (OWASP Top 10, CIS Controls, etc.).