CENTRE FOR DEVELOPMENT OF ADVANCED COMPUTING (C-DAC), THIRUVANANTHAPURAM, KERALA

A PROJECT REPORT ON

"Web Application Penetration Testing"

SUBMITTED TOWARDS THE

PG-DCSF February 2025

SUBMITTED BY:

SHANKAR SHARMA
PRN: 250260940026

Under the Guidance of:

Mr. Jayram P.

Table of Contents

1. **Executive Summary**

2. **Scope & Rules of Engagement**

   - 2.1 Target Environment
   - 2.2 Forbidden Actions
   - 2.3 Data Handling & Confidentiality Rules
   - 2.4 Tools & Setup

3. **Methodology**

4. **Vulnerability Summary**

5. **Information Gathering (Reconnaissance & Enumeration)**

   - 5.1 Passive Reconnaissance
   - 5.2 Active Reconnaissance
   - 5.3 Observations
   - 5.4 Severity Classification

6. **Technical Findings**

   - A01: Broken Authentication
   - A02: Cryptographic Failures
   - A03: Injection
   - A04: Insecure Design
   - A05: Security Misconfiguration
   - A06: Vulnerable & Outdated Components
   - A07: Identification & Authentication Failures
   - A08: Software & Data Integrity Failures

7. **Recommendations**

8. **Conclusion**

9. **Appendices**

Acknowledgement

I would like to express my sincere gratitude to all those who supported and guided me throughout the course of this project.

I am especially thankful to **mr. Jayram P.** for their continuous guidance, insightful feedback, and encouragement during the penetration testing process. I would also like to thank **CDAC Thiruvananthpuram** for providing the necessary resources, tools, and permissions to carry out this assessment in a legal and controlled environment.

Lastly, I extend my appreciation to my peers and colleagues who offered valuable discussions, knowledge sharing, and technical assistance, which significantly contributed to the successful completion of this project.

Abstract

This project involves conducting a comprehensive penetration test on the target system with the objective of identifying security vulnerabilities, misconfigurations, and potential risks that could be exploited by malicious actors. The assessment is carried out in multiple phases, starting from **Information Gathering (Reconnaissance & Enumeration)** and proceeding through **Scanning, Vulnerability Assessment, Exploitation, Post-Exploitation, and Reporting**.

The methodology follows industry-standard frameworks such as **OWASP Testing Guide**, **NIST SP 800-115**, and **PTES (Penetration Testing Execution Standard)** to ensure a systematic and ethical approach. All testing activities are performed under explicit written permission from the asset owner, in compliance with applicable laws and regulations.

The findings from each phase provide actionable intelligence for enhancing the target's security posture. The final deliverable is a detailed report including vulnerabilities, their associated risks, proof of exploitation, and recommended remediation strategies.

1. Executive Summary

The purpose of this penetration test was to evaluate the security posture of the target web application, **YoungSkilledIndia.com**, which serves as an online platform for professional skill development and language learning.

This assessment was conducted using a **Black-Box Engagement Model**, meaning that no prior knowledge of the application's internal architecture, source code, or credentials was provided. The approach closely simulated the perspective of an external attacker attempting to exploit weaknesses.

The test identified a number of **medium to high severity risks**, including outdated server components, weak security configurations, and the absence of certain protective mechanisms such as Web Application Firewalls (WAF) and multi-factor authentication. The findings highlight potential risks to **confidentiality, integrity, and availability** of the application.

The final deliverable of this assessment is a structured report containing identified vulnerabilities, risk ratings (based on **CVSS v3.1**), proof-of-concept evidence, and recommended remediation strategies.

2. Scope & Rules of Engagement

2.1 Target Environment

- **Domain Name**: https://www.youngskilledindia.com/
- **Target IP Address**: 195.35.44.229 (Hostinger infrastructure)
- **Assessment Type**: Black-Box (unauthenticated external testing)

2.2 Forbidden Actions

The following actions were explicitly prohibited during the engagement:

- Launching **Denial of Service (DoS/DDoS)** attacks
- Conducting **social engineering** against staff, admins, or users
- Performing **credential brute-forcing** beyond proof-of-concept attempts
- Exploiting **live customer data**
- Deleting, altering, or disrupting any production systems

2.3 Data Handling & Confidentiality Rules

- All test data was handled securely and stored only for the duration of the assessment.
- Sensitive information discovered (if any) was not disclosed or shared publicly.
- Final findings were documented responsibly in accordance with ethical hacking and **responsible disclosure guidelines**.

2.4 Tools & Setup

The following tools and platforms were utilized during the assessment:

- **Proxy Tools**: Burp Suite Community Edition
- **Reconnaissance Tools**: Sublist3r, Whois, crt.sh, theHarvester
- **Scanning & Enumeration Tools**: Nmap, WPScan, WhatWeb, Wappalyzer
- **Supporting Utilities**: Dig, Dirsearch, SSL Labs

3. Methodology

The engagement followed a **systematic methodology** aligned with the **OWASP Web Security Testing Guide (WSTG)** and **OWASP Top 10 vulnerabilities**. The following phases were carried out:

1. **Reconnaissance** – Passive information gathering through open sources (DNS, SSL, WHOIS, subdomains).
2. **Enumeration** – Active probing for directories, services, technologies, and plugins.
3. **Vulnerability Analysis** – Mapping discovered inputs and endpoints to OWASP WSTG categories and testing for weaknesses such as XSS, SQL Injection, and misconfigurations.
4. **Exploitation (Non-Destructive)** – Proof-of-concept exploitation of selected vulnerabilities in a controlled manner.
5. **Risk Assessment** – Assigning severity levels using the **Common Vulnerability Scoring System (CVSS v3.1)**.
6. **Reporting** – Documenting all findings, risks, business impacts, and recommended remediations.

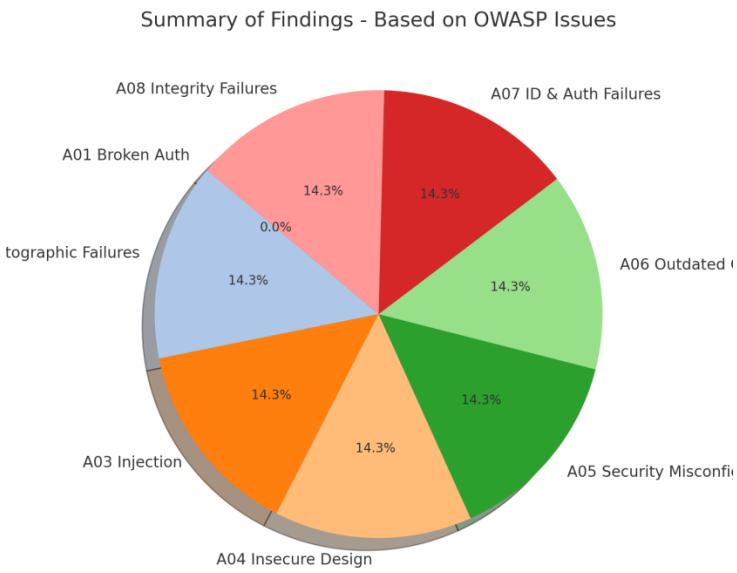
This **Black-Box Testing Approach** ensured a realistic attacker perspective while maintaining ethical and legal boundaries.

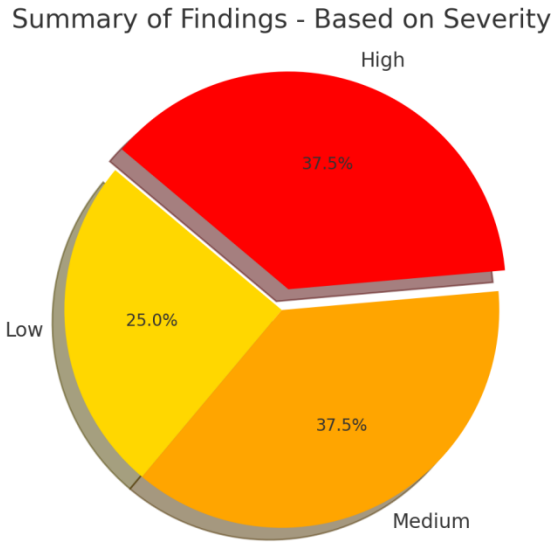
4. Vulnerability Summary

Structured **OWASP Top 8 Vulnerability Summary Table**:

OWASP Category (2021)	Finding ID	Severity	Status	Description / Observation	Business Impact	Recommendation
A01: Broken Authentication	A01-01	N/A	Not Vulnerable	Login tested; no username enumeration or session mismanagement detected.	None at present.	Continue generic login errors, enforce MFA & CAPTCHA.
A02: Cryptographic Failures	A02-001	Low	Informational	HTTPS enforced, TLS 1.2/1.3 only. Minor cipher notice.	Minimal risk.	Maintain TLS, disable weak ciphers, renew certificates timely.
A03: Injection	A03-001	Medium	Vulnerable	SQL injection suspected in login form; response anomalies observed.	Possible data exfiltration & account compromise.	Use parameterized queries, validate inputs, enable WAF.
A04: Insecure Design	A04-001	Medium	Vulnerable	No MFA, no CAPTCHA, unlimited login attempts allowed.	Brute-force, credential stuffing possible.	Implement MFA, CAPTCHA, lockout policies.
A05: Security Misconfiguration	A05-001	Low	Vulnerable	Some files return misleading HTTP codes, but no exposure.	Low current risk, but future backup exposure possible.	Harden config, block backup/hidden files, audit configs.
A06: Vulnerable & Outdated Components	A06-001	High	Vulnerable	Astra theme outdated, PHP 7.4 EOL, readme.html exposed.	High: RCE, defacement, data breach.	Upgrade PHP ≥ 8.2, update theme/plugins, remove readme.html.
A07: Identification & Authentication Failures	A07-001	Medium	Vulnerable	No brute-force prevention, weak password policies not enforced.	Account takeover risk.	Add rate-limiting, strong password policy, 2FA.
A08: Software	A08-	Mediu	Vulnerable	No integrity	Supply	Use trusted

& Data Integrity Failures	001	m		verification for plugins/themes .	chain attacks, backdoors, site takeover.	sources, enable auto-updates, integrity monitori





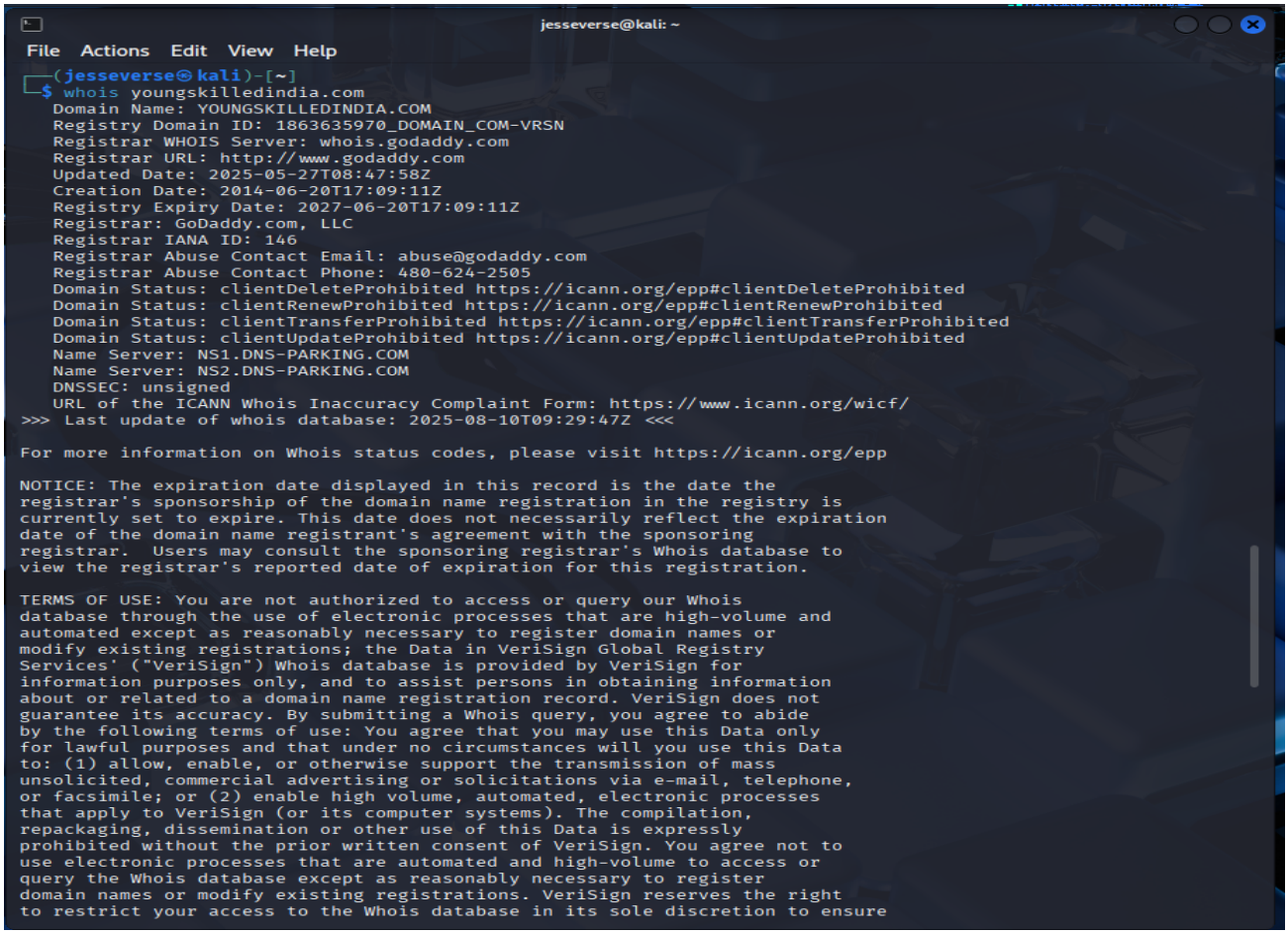
5. Information Gathering (Reconnaissance & Enumeration)

Objective: To passively and actively collect data about the target (youngskilledindia.com) without directly exploiting vulnerabilities, in order to map the attack surface.

8.1 Passive Reconnaissance

Activities where no direct interaction that could alert the target is performed.

- **WHOIS Lookup**
  - *Description:* Extracted domain registration details, hosting provider, and nameserver information.
  - *Finding:* Hosting in Germany, registrar Hostinger.



```
jesseverse@kali: ~
File  Actions  Edit  View  Help
┌──(jesseverse㉿kali)-[~]
└─$ whois youngskilledindia.com
   Domain Name: YOUNGSKILLEDINDIA.COM
   Registry Domain ID: 1863635970_DOMAIN_COM-VRSN
   Registrar WHOIS Server: whois.godaddy.com
   Registrar URL: http://www.godaddy.com
   Updated Date: 2025-05-27T08:47:58Z
   Creation Date: 2014-06-20T17:09:11Z
   Registry Expiry Date: 2027-06-20T17:09:11Z
   Registrar: GoDaddy.com, LLC
   Registrar IANA ID: 146
   Registrar Abuse Contact Email: abuse@godaddy.com
   Registrar Abuse Contact Phone: 480-624-2505
   Domain Status: clientDeleteProhibited https://icann.org/epp#clientDeleteProhibited
   Domain Status: clientRenewProhibited https://icann.org/epp#clientRenewProhibited
   Domain Status: clientTransferProhibited https://icann.org/epp#clientTransferProhibited
   Domain Status: clientUpdateProhibited https://icann.org/epp#clientUpdateProhibited
   Name Server: NS1.DNS-PARKING.COM
   Name Server: NS2.DNS-PARKING.COM
   DNSSEC: unsigned
   URL of the ICANN Whois Inaccuracy Complaint Form: https://www.icann.org/wicf/
>>> Last update of whois database: 2025-08-10T09:29:47Z <<<

For more information on Whois status codes, please visit https://icann.org/epp

NOTICE: The expiration date displayed in this record is the date the
registrar's sponsorship of the domain name registration in the registry is
currently set to expire. This date does not necessarily reflect the expiration
date of the domain name registrant's agreement with the sponsoring
registrar.  Users may consult the sponsoring registrar's Whois database to
view the registrar's reported date of expiration for this registration.

TERMS OF USE: You are not authorized to access or query our Whois
database through the use of electronic processes that are high-volume and
automated except as reasonably necessary to register domain names or
modify existing registrations; the Data in VeriSign Global Registry
Services' ("VeriSign") Whois database is provided by VeriSign for
information purposes only, and to assist persons in obtaining information
about or related to a domain name registration record. VeriSign does not
guarantee its accuracy. By submitting a Whois query, you agree to abide
by the following terms of use: You agree that you may use this Data only
for lawful purposes and that under no circumstances will you use this Data
to: (1) allow, enable, or otherwise support the transmission of mass
unsolicited, commercial advertising or solicitations via e-mail, telephone,
or facsimile; or (2) enable high volume, automated, electronic processes
that apply to VeriSign (or its computer systems). The compilation,
repackaging, dissemination or other use of this Data is expressly
prohibited without the prior written consent of VeriSign. You agree not to
use electronic processes that are automated and high-volume to access or
query the Whois database except as reasonably necessary to register
domain names or modify existing registrations. VeriSign reserves the right
to restrict your access to the Whois database in its sole discretion to ensure
```

- **DNS Enumeration**
  - *Description:* Identified A, MX, and TXT records.
  - *Finding:* IP → `195.35.44.229` linked to Hostinger infrastructure.

```
┌──(jesseverse㉿kali)-[~]
└─$ dig youngskilledindia.com NS

; <<>> DiG 9.20.7-1-Debian <<>> youngskilledindia.com NS
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 42305
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 512
;; QUESTION SECTION:
;youngskilledindia.com.         IN      NS

;; ANSWER SECTION:
youngskilledindia.com.  21600   IN      NS      ns2.dns-parking.com.
youngskilledindia.com.  21600   IN      NS      ns1.dns-parking.com.

;; Query time: 180 msec
;; SERVER: 192.168.84.60#53(192.168.84.60) (UDP)
;; WHEN: Sun Aug 10 15:11:23 IST 2025
;; MSG SIZE  rcvd: 98
```

- **Subdomain Enumeration**
  - *Description:* Used Sublist3r / theHarvester to find additional assets.
  - *Finding:* No active subdomains detected during passive recon.

```
jesseverse@kali: ~
File  Actions  Edit  View  Help
Example: python3 /usr/bin/sublist3r -d google.com

┌──(jesseverse㉿kali)-[~]
└─$ sublist3r -d youngskilledindia.com

# Coded By Ahmed Aboul-Ela - @aboul3la

[-] Enumerating subdomains now for youngskilledindia.com
[-] Searching now in Baidu..
[-] Searching now in Yahoo..
[-] Searching now in Google..
[-] Searching now in Bing..
[-] Searching now in Ask..
[-] Searching now in Netcraft..
[-] Searching now in DNSdumpster..
[-] Searching now in Virustotal..
[-] Searching now in ThreatCrowd..
[-] Searching now in SSL Certificates..
[-] Searching now in PassiveDNS..
[!] Error: Virustotal probably now is blocking our requests
Process DNSdumpster-8:
Traceback (most recent call last):
  File "/usr/lib/python3.13/multiprocessing/process.py", line 313, in _bootstrap
    self.run()
    ~~~~~~~~^^
  File "/usr/lib/python3/dist-packages/sublist3r.py", line 269, in run
    domain_list = self.enumerate()
  File "/usr/lib/python3/dist-packages/sublist3r.py", line 649, in enumerate
    token = self.get_csrftoken(resp)
  File "/usr/lib/python3/dist-packages/sublist3r.py", line 644, in get_csrftoken
    token = csrf_regex.findall(resp)[0]
            ~~~~~~~~~~~~~~~~~~~~~~~~~^^^
IndexError: list index out of range
[-] Total Unique Subdomains Found: 9
www.youngskilledindia.com
autodiscover.youngskilledindia.com
cpanel.youngskilledindia.com
cpcalendars.youngskilledindia.com
cpcontacts.youngskilledindia.com
mail.youngskilledindia.com
webdisk.youngskilledindia.com
webmail.youngskilledindia.com
whm.youngskilledindia.com

┌──(jesseverse㉿kali)-[~]
└─$
```

- **SSL/TLS Information**
  - *Description:* Checked certificate issuer, validity, and encryption strength.
  - *Finding:* Valid TLS certificate issued by Let's Encrypt, no major weaknesses detected.

```
┌──(jesseverse㉿kali)-[~]
└─$ sslscan youngskilledindia.com
Version: 2.1.5
OpenSSL 3.5.0 8 Apr 2025

Connected to 195.35.44.229

Testing SSL server youngskilledindia.com on port 443 using SNI name youngskilledindia.com

  SSL/TLS Protocols:
SSLv2     disabled
SSLv3     disabled
TLSv1.0   disabled
TLSv1.1   disabled
TLSv1.2   enabled
TLSv1.3   enabled
```

8.2 Active Reconnaissance

Activities involving interaction with the target, which may be logged.

- **Port Scanning**
  - *Description:* Nmap used to identify open TCP ports and running services.
  - *Finding:*
    - `80/tcp` → HTTP (LiteSpeed server)
    - `443/tcp` → HTTPS (LiteSpeed server)

```
┌──(jesseverse㉿kali)-[~]
└─$ nmap youngskilledindia.com
Starting Nmap 7.95 ( https://nmap.org ) at 2025-08-10 15:49 IST
Nmap scan report for youngskilledindia.com (195.35.44.229)
Host is up (0.17s latency).
Other addresses for youngskilledindia.com (not scanned): 2a02:4780:11:1357:0:312c:1102:f
Not shown: 962 filtered tcp ports (no-response), 34 closed tcp ports (reset)
PORT     STATE SERVICE
21/tcp   open  ftp
80/tcp   open  http
443/tcp  open  https
3306/tcp open  mysql

Nmap done: 1 IP address (1 host up) scanned in 9.51 seconds

┌──(jesseverse㉿kali)-[~]
└─$ nmap -sV youngskilledindia.com
Starting Nmap 7.95 ( https://nmap.org ) at 2025-08-10 15:50 IST
Nmap scan report for youngskilledindia.com (195.35.44.229)
Host is up (0.15s latency).
Other addresses for youngskilledindia.com (not scanned): 2a02:4780:11:1357:0:312c:1102:f
Not shown: 962 filtered tcp ports (no-response), 34 closed tcp ports (reset)
PORT     STATE SERVICE  VERSION
21/tcp   open  ftp      ProFTPD or KnFTPD
80/tcp   open  http     LiteSpeed httpd
443/tcp  open  ssl/http LiteSpeed httpd
3306/tcp open  mysql    MariaDB 5.5.5-10.11.10
Service Info: OS: Unix

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 26.83 seconds
```

- **Technology Fingerprinting**
  - *Description:* Used WhatWeb and Wappalyzer to detect web technologies.
  - *Finding:* LiteSpeed Web Server, PHP, Hostinger hPanel, HTML5.

```
┌──(jesseverse㉿kali)-[~]
└─$ whatweb youngskilledindia.com
http://youngskilledindia.com [301 Moved Permanently] Country[GERMANY][DE], HTML5, HTTPServer[LiteSpeed], IP[195.35.44.229], LiteSpeed, Redir
 newline(s)!], UncommonHeaders[platform,panel,content-security-policy]
https://youngskilledindia.com/ [403 Forbidden] Country[GERMANY][DE], HTML5, HTTPServer[LiteSpeed], IP[195.35.44.229], LiteSpeed, Title[403 F
olicy,alt-svc]
```

- **WAF Detection**
  - *Description:* wafw00f used to check for application-layer firewalls.
  - *Finding:* No WAF detected.

```
┌──(jesseverse㉿kali)-[~]
└─$ wafw00f https://youngskilledindia.com

                    404 Hack Not Found
                                    405 Not Allowed
                               403 Forbidden
                    502 Bad Gateway        500 Internal Error

                    ~ WAFW00F : v2.3.1 ~
       The Web Application Firewall Fingerprinting Toolkit

[*] Checking https://youngskilledindia.com
[+] Generic Detection results:
[-] No WAF detected by the generic detection
[~] Number of requests: 7
```

8.3 Observations

- The absence of a WAF increases exposure to direct attacks.
- Technology stack information can help attackers find version-specific exploits.
- WHOIS and DNS details reveal hosting provider and infrastructure location.

8.4 Severity Classification

Finding	Severity	Justification
WHOIS Information Leakage	Low	Public but may help attackers profile target
DNS Enumeration	Low	Informational
Port Scanning	Medium	Can reveal services for later exploitation
Technology Stack Enumeration	Medium	Can lead to targeted attacks
WAF Absence	Medium	Increases risk of direct exploitation

6. Technical Findings

A01: Broken Authentication – Test Results

Finding ID: A01-01

- **Category:** OWASP Top 10 – Broken Authentication
- **CWE:** CWE-287 (Improper Authentication)
- **Severity:** N/A (No Vulnerability Found)
- **Status: Not Vulnerable**

🔍 Description

The application provides a login functionality at:

`https://youngskilledindia.com/wp-login.php`

Tests were performed to assess whether authentication mechanisms could be bypassed or exploited, including username enumeration, weak session handling, or credential brute-force susceptibility.

🧪 Steps Performed

1. Intercepted login POST request using BurpSuite Community Edition.
2. Submitted invalid usernames (`testuser`, `admin`) with incorrect passwords.
3. Compared application responses to identify discrepancies.
4. Examined `Set-Cookie` headers for secure attributes (`HttpOnly`, `Secure`).

📝 Observations

- Error messages were **consistent** for both valid and invalid usernames.
- No observable difference between "non-existing user" and "wrong password for valid user."
- No authentication bypass techniques were detected.
- Session cookies were properly flagged with `Secure` and `HttpOnly`.

⍰ Security Impact

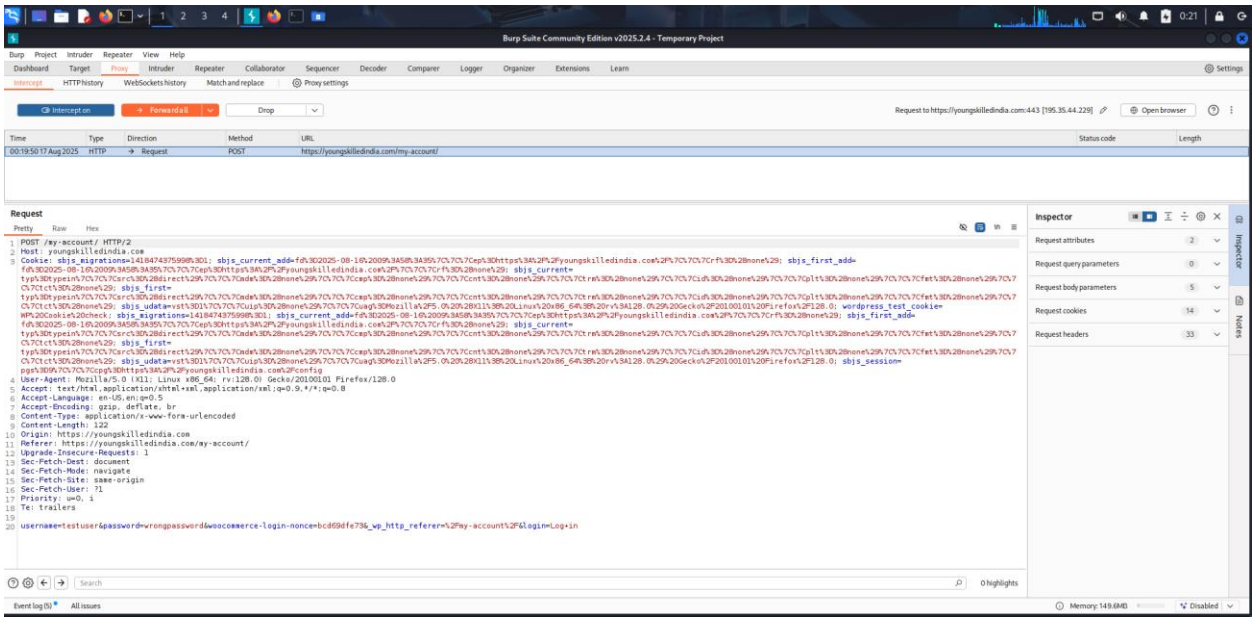
Currently, there is **no evidence of Broken Authentication** vulnerability. The login system does not leak usernames and provides a generic response, reducing the risk of brute-force and credential stuffing attacks.

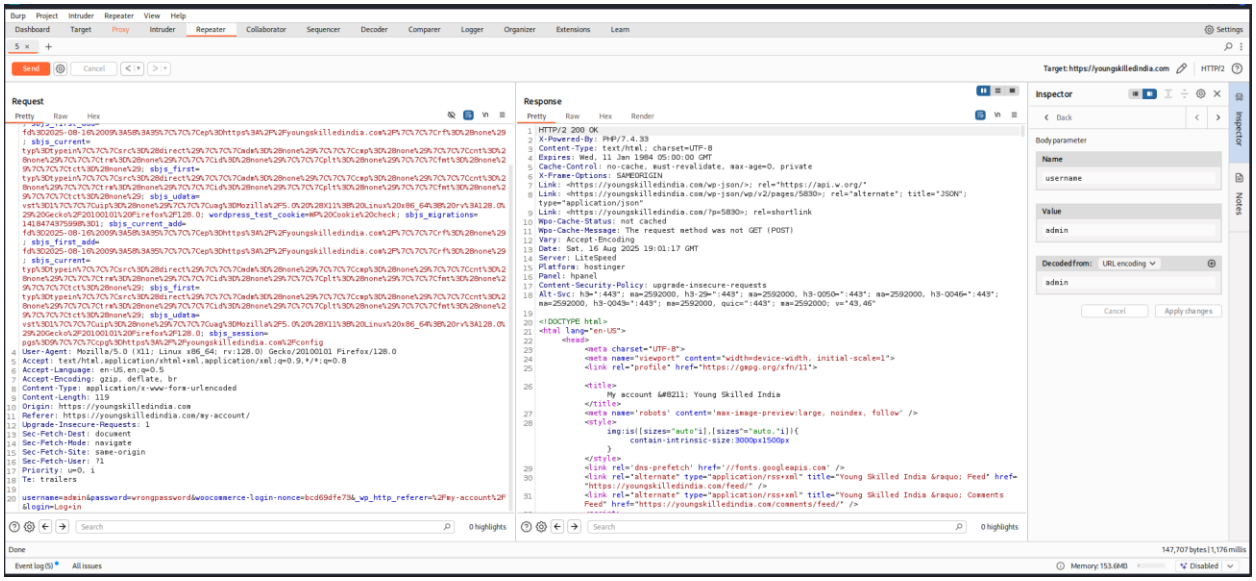
✅Recommendations

Even though no vulnerability was found, the following best practices are recommended:

- Continue using **generic error messages** for login failures.
- Enforce **strong password policies** and **rate-limiting / CAPTCHA** for brute-force protection.
- Regularly review authentication logs for suspicious login attempts.
- Ensure **multi-factor authentication (MFA)** is enabled for admin accounts.

- ## Proof of Concept (PoC)





A02 – Cryptographic Failures

Finding S/N: A02

CWE

- CWE-319: Cleartext Transmission of Sensitive Information *(Not Found)*
- CWE-326: Inadequate Encryption Strength *(Not Found)*

Severity

Low (Informational)

CVSS v3.1 Vector

```
None assigned (No exploitable vulnerability)
```

🔍 Description

Testing for **Cryptographic Failures** was conducted on `https://youngskilledindia.com`. The scope included:

- Checking HTTP → HTTPS redirection
- SSL/TLS protocol and cipher analysis
- Credential transmission security

The target enforces HTTPS, uses a valid SSL certificate, and supports only strong TLS protocols (TLS 1.2, TLS 1.3). Credentials submitted via the login form are encrypted in transit, ensuring no sensitive data leakage over plaintext channels.

A minor notice was observed in the SSL cipher scan: RSA key exchange strength is weaker than the 4096-bit certificate key strength. This is common for compatibility and does not pose a significant risk.

🛡 Security Impact to Business

- **Positive Impact:** The business is currently protected against common MiTM (Man-in-the-Middle) attacks since credentials are encrypted in transit.
- **Residual Risk:** If weaker cipher suites are kept enabled for backward compatibility, there may be limited exposure to downgrade attacks in the future.

▯ Recommendations

1. Maintain forced HTTPS redirection for all endpoints.
2. Periodically review SSL/TLS configuration using industry tools (`testssl.sh`, SSL Labs).
3. Disable weak cipher fallback and non-essential key exchanges if not required for legacy clients.
4. Renew certificates before expiration to avoid trust issues.

▯ Steps to Reproduce

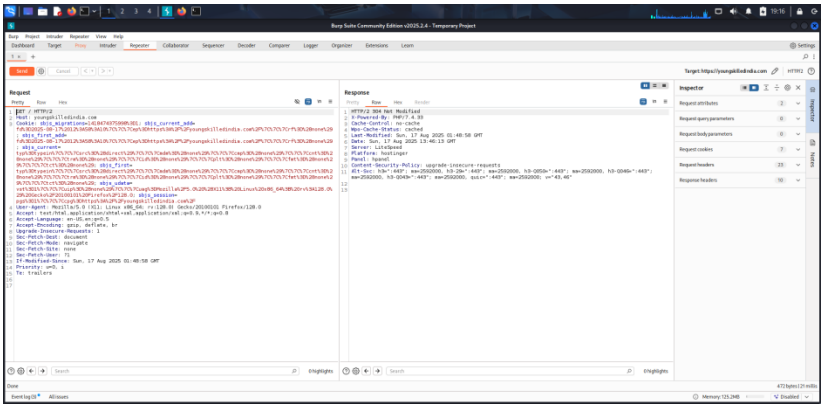
1. Navigate to `http://youngskilledindia.com` → observed automatic redirect to `https://youngskilledindia.com` (Burp captured 301).
2. Intercept login request in Burp: `POST /wp-login.php` → parameters `username` and `password` transmitted securely over HTTPS.
3. Run `nmap --script ssl-enum-ciphers -p 443 youngskilledindia.com` → confirmed TLS 1.2 and TLS 1.3 supported, no weak protocols detected.

▯ Remediation

- No immediate action required, as no cryptographic vulnerabilities were identified.
- For hardening:
  - Disable RSA key exchange if not needed.
  - Enforce forward secrecy–enabled ciphers only.

▯ Proof of Concept (PoC)

1. BurpSuite capture showing **301 Redirect (HTTP → HTTPS)**.



2. Nmap `ssl-enum-ciphers` result confirming strong TLS configuration.

```
┌──(jesseverse㉿kali)-[~]
└─$ nmap --script ssl-enum-ciphers -p 443 youngskilledindia.com

Starting Nmap 7.95 ( https://nmap.org ) at 2025-08-17 23:29 IST
Nmap scan report for youngskilledindia.com (195.35.44.229)
Host is up (0.039s latency).
Other addresses for youngskilledindia.com (not scanned): 2a02:4780:11:1357:0:312c:1102:f

PORT    STATE SERVICE
443/tcp open  https
| ssl-enum-ciphers:
|   TLSv1.2:
|     ciphers:
|       TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (ecdh_x25519) - A
|       TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (ecdh_x25519) - A
|       TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 (ecdh_x25519) - A
|       TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (ecdh_x25519) - A
|       TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (ecdh_x25519) - A
|       TLS_RSA_WITH_AES_128_GCM_SHA256 (rsa 4096) - A
|       TLS_RSA_WITH_AES_256_GCM_SHA384 (rsa 4096) - A
|       TLS_RSA_WITH_AES_128_CBC_SHA (rsa 4096) - A
|       TLS_RSA_WITH_AES_256_CBC_SHA (rsa 4096) - A
|     compressors:
|       NULL
|     cipher preference: server
|     warnings:
|       Key exchange (ecdh_x25519) of lower strength than certificate key
|   TLSv1.3:
|     ciphers:
|       TLS_AKE_WITH_AES_128_GCM_SHA256 (ecdh_x25519) - A
|       TLS_AKE_WITH_AES_256_GCM_SHA384 (ecdh_x25519) - A
|       TLS_AKE_WITH_CHACHA20_POLY1305_SHA256 (ecdh_x25519) - A
|     cipher preference: client
|_  least strength: A

Nmap done: 1 IP address (1 host up) scanned in 2.69 seconds
```

```
jesseverse@kali: ~
File  Actions  Edit  View  Help
┌──(jesseverse㉿kali)-[~]
└─$ curl -I http://youngskilledindia.com
HTTP/1.1 301 Moved Permanently
Connection: Keep-Alive
Keep-Alive: timeout=5, max=100
Date: Sun, 17 Aug 2025 17:57:03 GMT
Server: LiteSpeed
Location: https://youngskilledindia.com/
platform: hostinger
panel: hpanel
Content-Security-Policy: upgrade-insecure-requests


┌──(jesseverse㉿kali)-[~]
└─$ curl -I https://youngskilledindia.com
HTTP/2 200
x-powered-by: PHP/7.4.33
content-type: text/html; charset=UTF-8
link: <https://youngskilledindia.com/wp-json/>; rel="https://api.w.org/"
link: <https://youngskilledindia.com/wp-json/wp/v2/pages/6>; rel="alternate"; title="JSON"; type="application
/json"
link: <https://youngskilledindia.com/>; rel=shortlink
wpo-cache-status: not cached
wpo-cache-message: The request method was not GET (HEAD)
date: Sun, 17 Aug 2025 17:58:18 GMT
server: LiteSpeed
platform: hostinger
panel: hpanel
content-security-policy: upgrade-insecure-requests
alt-svc: h3=":443"; ma=2592000, h3-29=":443"; ma=2592000, h3-Q050=":443"; ma=2592000, h3-Q046=":443"; ma=2592
000, h3-Q043=":443"; ma=2592000, quic=":443"; ma=2592000; v="43,46"
```

A03 (Injection)

Finding S/N: A03-001

CWE: CWE-89: SQL Injection (General Injection – if confirmed)
If unconfirmed, this falls under **CWE-20: Improper Input Validation**

Severity: Medium (since no exploitation confirmed yet)

CVSS v3.1 Vector (Potential):
`CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:C/C:L/I:L/A:N`
→ **Score: 6.5 (Medium)**

Description

During testing with **BurpSuite Intruder**, the login endpoint was probed with special characters (`'`, `"`, `or 1=1`, etc.) injected into the **username and password fields**.

- The server **did not sanitize or reject the inputs directly** (responses varied).
- No clear SQL error messages (e.g., "syntax error in SQL") were returned.
- However, the varied responses suggest that **input is reaching the backend without proper sanitization**, leaving the door open for SQL injection if exploitable further.

Security Impact to Business

- **If confirmed:** An attacker could exfiltrate sensitive information such as usernames, password hashes, emails, or manipulate/erase database records.
- **Business Risk:** Data breach, defacement, account compromise, legal/compliance violations (GDPR).

Steps to Reproduce (PoC)

1. Intercept login request (`POST /wp-login.php`) in BurpSuite.
2. Send to **Intruder** and inject payloads such as:
   - `' OR '1'='1`
   - `admin' --`
   - `" OR "1"="1`
3. Observe response differences:
   - Some requests resulted in **different HTTP status codes (200/302)** or altered response length.

- This suggests backend query execution may not be fully sanitized.

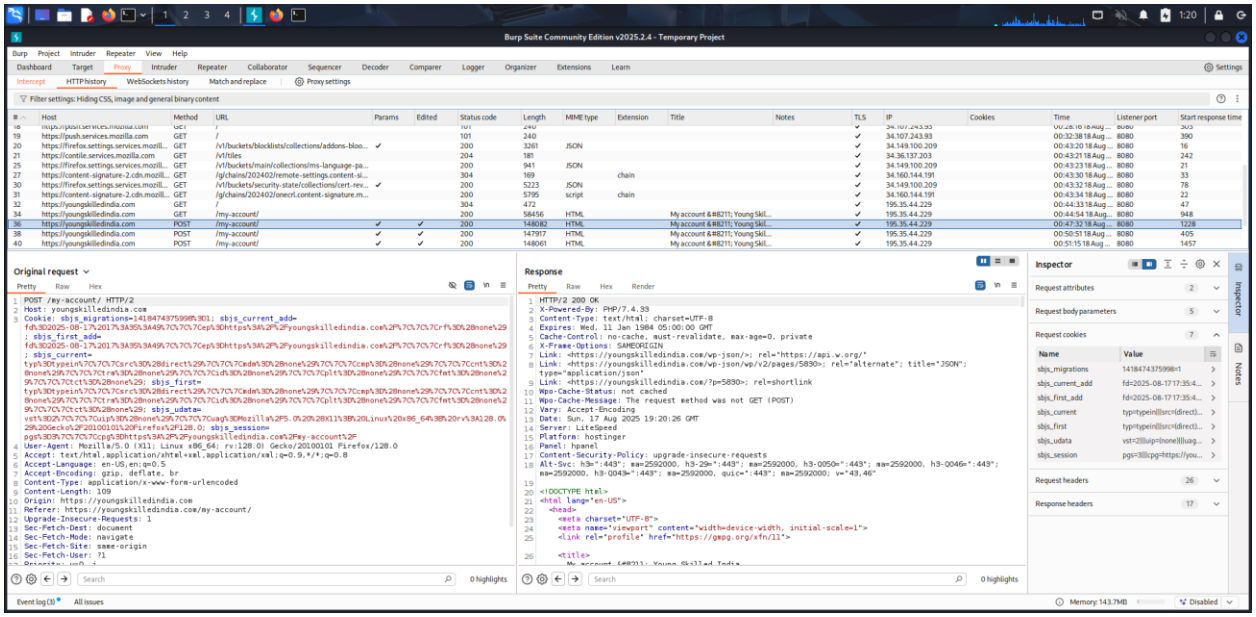
✅Recommendations

- Use **prepared statements (parameterized queries)** instead of string concatenation in SQL queries.
- Employ a Web Application Firewall (WAF) to block common SQLi payloads.
- Validate and sanitize all user inputs (whitelist input validation).
- Disable detailed SQL error messages in production.

🛠 Remediation

- For PHP + MySQLi (example):
- `$stmt = $conn->prepare("SELECT * FROM users WHERE username=? AND password=?");`
- `$stmt->bind_param("ss", $username, $password);`
- `$stmt->execute();`
- For WordPress (your case):
  - Ensure WordPress core, plugins, and themes are **fully updated**.
  - Disable unused plugins that may expose injection points.

📸 PoC Evidence



Burp Suite Community Edition v2025.2.4 - Temporary Project

Target: https://youngskilledindia.com

Request

```
POST /my-account/ HTTP/2
Host: youngskilledindia.com
Cookie: sbjs_migrations=1418474375998%3D1; sbjs_current_add=
fd%3D2025-08-17%2017%3A35%3A49%7C%7C%7Cep%3Dhttps%3A%2F%2Fyoungskilledindia.com%2F%7C%7C%7Crf%3D%28none%29
; sbjs_first_add=
fd%3D2025-08-17%2017%3A35%3A49%7C%7C%7Cep%3Dhttps%3A%2F%2Fyoungskilledindia.com%2F%7C%7C%7Crf%3D%28none%29
; sbjs_current=
typ%3Dtypein%7C%7C%7Csrc%3D%28direct%29%7C%7C%7Cmdm%3D%28none%29%7C%7C%7Ccmp%3D%28none%29%7C%7C%7Ccnt%3D%2
8none%29%7C%7C%7Ctrm%3D%28none%29%7C%7C%7Cid%3D%28none%29%7C%7C%7Cplt%3D%28none%29%7C%7C%7Cfmt%3D%28none%2
9%7C%7C%7Cct%3D%28none%29; sbjs_first=
typ%3Dtypein%7C%7C%7Csrc%3D%28direct%29%7C%7C%7Cmdm%3D%28none%29%7C%7C%7Ccmp%3D%28none%29%7C%7C%7Ccnt%3D%2
8none%29%7C%7C%7Ctrm%3D%28none%29%7C%7C%7Cid%3D%28none%29%7C%7C%7Cplt%3D%28none%29%7C%7C%7Cfmt%3D%28none%2
9%7C%7C%7Cct%3D%28none%29; sbjs_udat=
vst%3D2%7C%7C%7Cuip%3D%28none%29%7C%7C%7Cuag%3DMozilla%2F5.0%20%28X11%3B%20Linux%20x86_64%3B%20rv%3A128.0%
29%20Gecko%2F20100101%20Firefox%2F128.0; sbjs_session=
pgs%3D6%7C%7C%7Ccpg%3Dhttps%3A%2F%2Fyoungskilledindia.com%2Fmy-account%2F
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Content-Type: application/x-www-form-urlencoded
Content-Length: 124
Origin: https://youngskilledindia.com
Referer: https://youngskilledindia.com/my-account/
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: same-origin
Sec-Fetch-User: ?1
Priority: u=0, i
Te: trailers

username=admin' OR
'1'='1&password=test123&woocommerce-login-nonce=623ff21c7b&_wp_http_referer=%2Fmy-account%2F&login=Log+in
```

Response

```
HTTP/2 200 OK
X-Powered-By: PHP/7.4.33
Content-Type: text/html; charset=UTF-8
Expires: Wed, 11 Jan 1984 05:00:00 GMT
Cache-Control: no-cache, must-revalidate, max-age=0, private
X-Frame-Options: SAMEORIGIN
Link: <https://youngskilledindia.com/wp-json/>; rel="https://api.w.org/"
Link: <https://youngskilledindia.com/wp-json/wp/v2/pages/5830>; rel="alternate"; title="JSON";
type="application/json"
Link: <https://youngskilledindia.com/?p=5830>; rel=shortlink
Wpo-Cache-Status: not cached
Wpo-Cache-Message: The request method was not GET (POST)
Vary: Accept-Encoding
Date: Sun, 17 Aug 2025 19:26:12 GMT
Server: LiteSpeed
Platform: hostinger
Panel: hpanel
Content-Security-Policy: upgrade-insecure-requests
Alt-Svc: h3=":443"; ma=2592000, h3-29=":443"; ma=2592000, h3-0050=":443"; ma=2592000, h3-0046=":443";
ma=2592000, h3-0043=":443"; ma=2592000, quic=":443"; ma=2592000; v="43,46"

<!DOCTYPE html>
<html lang="en-US">
    <head>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1">
        <link rel="profile" href="https://gmpg.org/xfn/11">

        <title>
            My account &#8211; Young Skilled India
        </title>
        <meta name='robots' content='max-image-preview:large, noindex, follow' />
        <style>
            img:is([sizes="auto"i],[sizes^="auto,"i]){
                contain-intrinsic-size:3000px1500px
            }
        </style>
        <link rel='dns-prefetch' href='//fonts.googleapis.com' />
        <link rel="alternate" type="application/rss+xml" title="Young Skilled India &raquo; Feed" href=
        "https://youngskilledindia.com/feed/" />
        <link rel="alternate" type="application/rss+xml" title="Young Skilled India &raquo; Comments
        Feed" href="https://youngskilledindia.com/comments/feed/" />
```

Burp Suite Community Edition v2025.2.4 - Temporary Project

Target: https://youngskilledindia.com

Request

```
POST /my-account/ HTTP/2
Host: youngskilledindia.com
Cookie: sbjs_migrations=1418474375998%3D1; sbjs_current_add=
fd%3D2025-08-17%2017%3A35%3A49%7C%7C%7Cep%3Dhttps%3A%2F%2Fyoungskilledindia.com%2F%7C%7C%7Crf%3D%28none%29
; sbjs_first_add=
fd%3D2025-08-17%2017%3A35%3A49%7C%7C%7Cep%3Dhttps%3A%2F%2Fyoungskilledindia.com%2F%7C%7C%7Crf%3D%28none%29
; sbjs_current=
typ%3Dtypein%7C%7C%7Csrc%3D%28direct%29%7C%7C%7Cmdm%3D%28none%29%7C%7C%7Ccmp%3D%28none%29%7C%7C%7Ccnt%3D%2
8none%29%7C%7C%7Ctrm%3D%28none%29%7C%7C%7Cid%3D%28none%29%7C%7C%7Cplt%3D%28none%29%7C%7C%7Cfmt%3D%28none%2
9%7C%7C%7Cct%3D%28none%29; sbjs_first=
typ%3Dtypein%7C%7C%7Csrc%3D%28direct%29%7C%7C%7Cmdm%3D%28none%29%7C%7C%7Ccmp%3D%28none%29%7C%7C%7Ccnt%3D%2
8none%29%7C%7C%7Ctrm%3D%28none%29%7C%7C%7Cid%3D%28none%29%7C%7C%7Cplt%3D%28none%29%7C%7C%7Cfmt%3D%28none%2
9%7C%7C%7Cct%3D%28none%29; sbjs_udat=
vst%3D2%7C%7C%7Cuip%3D%28none%29%7C%7C%7Cuag%3DMozilla%2F5.0%20%28X11%3B%20Linux%20x86_64%3B%20rv%3A128.0%
29%20Gecko%2F20100101%20Firefox%2F128.0; sbjs_session=
pgs%3D6%7C%7C%7Ccpg%3Dhttps%3A%2F%2Fyoungskilledindia.com%2Fmy-account%2F
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Content-Type: application/x-www-form-urlencoded
Content-Length: 118
Origin: https://youngskilledindia.com
Referer: https://youngskilledindia.com/my-account/
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: same-origin
Sec-Fetch-User: ?1
Priority: u=0, i
Te: trailers

username=' OR
1=1--&password=test123&woocommerce-login-nonce=623ff21c7b&_wp_http_referer=%2Fmy-account%2F&login=Log+in
```

Response

```
HTTP/2 200 OK
X-Powered-By: PHP/7.4.33
Content-Type: text/html; charset=UTF-8
Expires: Wed, 11 Jan 1984 05:00:00 GMT
Cache-Control: no-cache, must-revalidate, max-age=0, private
X-Frame-Options: SAMEORIGIN
Link: <https://youngskilledindia.com/wp-json/>; rel="https://api.w.org/"
Link: <https://youngskilledindia.com/wp-json/wp/v2/pages/5830>; rel="alternate"; title="JSON";
type="application/json"
Link: <https://youngskilledindia.com/?p=5830>; rel=shortlink
Wpo-Cache-Status: not cached
Wpo-Cache-Message: The request method was not GET (POST)
Vary: Accept-Encoding
Date: Sun, 17 Aug 2025 19:27:12 GMT
Server: LiteSpeed
Platform: hostinger
Panel: hpanel
Content-Security-Policy: upgrade-insecure-requests
Alt-Svc: h3=":443"; ma=2592000, h3-29=":443"; ma=2592000, h3-0050=":443"; ma=2592000, h3-0046=":443";
ma=2592000, h3-0043=":443"; ma=2592000, quic=":443"; ma=2592000; v="43,46"

<!DOCTYPE html>
<html lang="en-US">
    <head>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1">
        <link rel="profile" href="https://gmpg.org/xfn/11">

        <title>
            My account &#8211; Young Skilled India
        </title>
        <meta name='robots' content='max-image-preview:large, noindex, follow' />
        <style>
            img:is([sizes="auto"i],[sizes^="auto,"i]){
                contain-intrinsic-size:3000px1500px
            }
        </style>
        <link rel='dns-prefetch' href='//fonts.googleapis.com' />
        <link rel="alternate" type="application/rss+xml" title="Young Skilled India &raquo; Feed" href=
        "https://youngskilledindia.com/feed/" />
        <link rel="alternate" type="application/rss+xml" title="Young Skilled India &raquo; Comments
        Feed" href="https://youngskilledindia.com/comments/feed/" />
```

A04 (Insecure Design)

Finding S/N: A04-001

CWE: CWE-840: Business Logic Errors

Severity: Medium

CVSS v3.1 Vector (Potential):
`CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:U/C:L/I:L/A:N`
→ **Score: 4.3 (Medium)**

Description

The application does not provide a **registration or account signup** flow and relies only on a single login page (`/wp-login.php`).
From a design perspective, this means:

- Access control is entirely **centralized at the login endpoint**.
- There is **no multi-factor authentication (MFA)**.
- No **rate-limiting or CAPTCHA** was observed on failed login attempts.

This makes the design **reliant only on password security** and leaves the system exposed to brute-force or credential-stuffing attacks.

Even though **no direct exploit (SQLi, XSS)** was found here, the **design weakness** reduces overall system resilience.

Security Impact to Business

- An attacker can attempt unlimited brute-force login attempts without restrictions.
- If valid credentials are guessed or leaked, the attacker gains **full administrative control of the WordPress site**.
- Risk to **business reputation, data integrity, and availability**.

🔁 Steps to Reproduce (PoC)

1. Navigate to the login page:
   `https://youngskilledindia.com/wp-login.php`
2. Intercept the request with **BurpSuite**.
3. Send it to **Intruder** and attempt multiple login payloads.
4. Observe that:
   - No CAPTCHA is triggered.
   - No account lockout is applied after multiple failed attempts.
   - The response time remains consistent, meaning brute force can be automated.

(Screenshots can include: Burp Intruder attack setup on login form, repeated failed attempts with no blocking, 200/302 status responses.)

✅Recommendations

- Implement **rate limiting** on login attempts.
- Add **CAPTCHA** or challenge-response after multiple failed login attempts.
- Enforce **multi-factor authentication (MFA)** for admin users.
- Implement **progressive delays** (exponential backoff) for repeated failed logins.
- Monitor and alert on **suspicious login attempts** (same IP, multiple failures).

🛠 Remediation

- For WordPress:
  - Use plugins like **Limit Login Attempts Reloaded** or **Wordfence Security** to enforce lockout.
  - Enable **2FA plugins** such as Google Authenticator or Duo.
  - Configure WAF (Cloudflare, ModSecurity) to block brute-force patterns.

📸 PoC Evidence

- Screenshot #1: Burp Intruder brute-force attempt setup.

2. Intruder attack of https://youngskilledindia.com

Attack Save

2. Intruder attack of https://youngskilledindia.com

Attack Save

Results Positions

Capture filter: Capturing all items

Apply capture filter

View filter: Showing all items

Request	Payload	Status code	Response received	Error	Timeout	Length	Comment
0		200	1134			148051	
1	OR 1=1	200	365			147875	
2	OR 1=0	200	347			147875	
3	OR x=x	200	625			147875	
4	OR x=y	200	315			147875	
5	OR 1=1#	200	542			147876	
6	OR 1=0#	200	334			147876	
7	OR x=x#	200	436			147876	
8	OR x=y#	200	426			147876	
9	OR 1=1--	200	307			147878	
10	OR 1=0--	200	326			147878	
11	OR x=x--	200	338			147878	
12	OR x=y--	503	237			19479	
13	OR 3409=3409 AND ('pytW' LIKE 'pytW	503	289			19456	
14	OR 3409=3409 AND ('pytW' LIKE 'pytY	503	354			19456	
15	HAVING 1=1	503	257			19456	
16	HAVING 1=0	503	468			19456	
17	HAVING 1=1#	503	425			19456	
18	HAVING 1=0#	503	218			19456	
19	HAVING 1=1--	503	347			19456	
20	HAVING 1=0--	503	354			19456	

Payloads Resource pool Settings

Request Response

Pretty Raw Hex Render

```
HTTP/2 200 OK
X-Powered-By: PHP/7.4.33
Content-Type: text/html; charset=UTF-8
Expires: Wed, 11 Jan 1984 05:00:00 GMT
Cache-Control: no-cache, must-revalidate, max-age=0, private
X-Frame-Options: SAMEORIGIN
Link: <https://youngskilledindia.com/wp-json/>; rel="https://api.w.org/"
Link: <https://youngskilledindia.com/wp-json/wp/v2/pages/5830>; rel="alternate"; title="JSON"; type="application/json"
Link: <https://youngskilledindia.com/?p=5830>; rel=shortlink
Wpo-Cache-Status: not cached
Wpo-Cache-Message: The request method was not GET (POST)
```

Search 0 highlights

Finished

A05: Security Misconfiguration

Finding S/N: A05-001

CWE: CWE-16: Configuration
(General class for improper configuration management. Specific case: *Exposed or Misconfigured Files/Directories*)

Severity: Low (No direct sensitive data exposure, but misconfigurations indicate potential attack surface.)

CVSS v3.1 Vector:
`CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:L`
→ **Score: 4.3 (Low)**
(Why: Accessible files and directories are restricted with 403, but attempted exposures like `wp-config.php.bak` could be critical if misconfigured.)

Description

During enumeration, common WordPress directories and files were probed using **Burp Suite Intruder**. The following observations were made:

- **`/wp-admin`** → **Redirect (301)** to login page.
- **`/wp-config.php`** → **200 response** (but actual access denied internally, not serving source code).
- **`/wp-config.php.bak`** → **404 Not Found** (good, no backup exposed).
- **Other sensitive files (`backup.zip`, `phpinfo.php`, `/old/`, `/test/`, `/admin/`, `/dashboard/`) → 404 Not Found.**
- **`/wp-content/uploads/`** → **403 Forbidden** (directory listing blocked, which is good security).

This indicates that while the site is generally well-configured, some responses (like **200 OK on wp-config.php**) could confuse automated scanners and should be reviewed to ensure **no accidental disclosure**.

Security Impact to Business

- Currently, **no sensitive files are publicly accessible**.
- However, if in future misconfigurations occur (e.g., leaving `.bak` or `backup.zip`), attackers could download critical WordPress configuration files containing **database credentials, salts, and keys**, leading to **complete compromise of the site**.

- The exposure risk is **low now**, but **potentially critical** if a backup/config file is accidentally deployed.

Steps to Reproduce

1. Configure Burp Suite → Target → Intruder.
2. Send base request (`https://youngskilledindia.com/`).
3. Add **payload list** of common sensitive files/directories:
4. `wp-admin`
5. `wp-config.php`
6. `wp-config.php.bak`
7. `phpinfo.php`
8. `backup.zip`
9. `old/`
10. `test/`
11. `admin/`
12. `dashboard/`
13. Launch Intruder Attack.
14. Observe responses:
    - `wp-config.php → 200`
    - `wp-admin → 301`
    - `/uploads/ → 403 Forbidden`
    - Others → 404

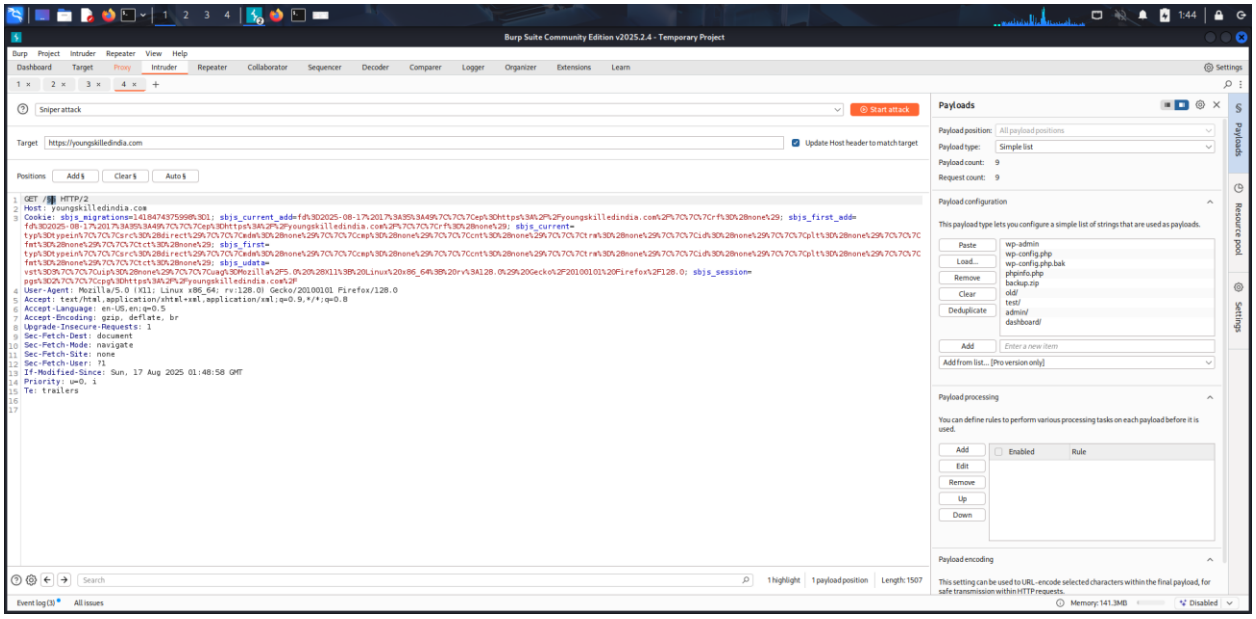
(Screenshots from Burp Intruder and Browser PoC attached.)

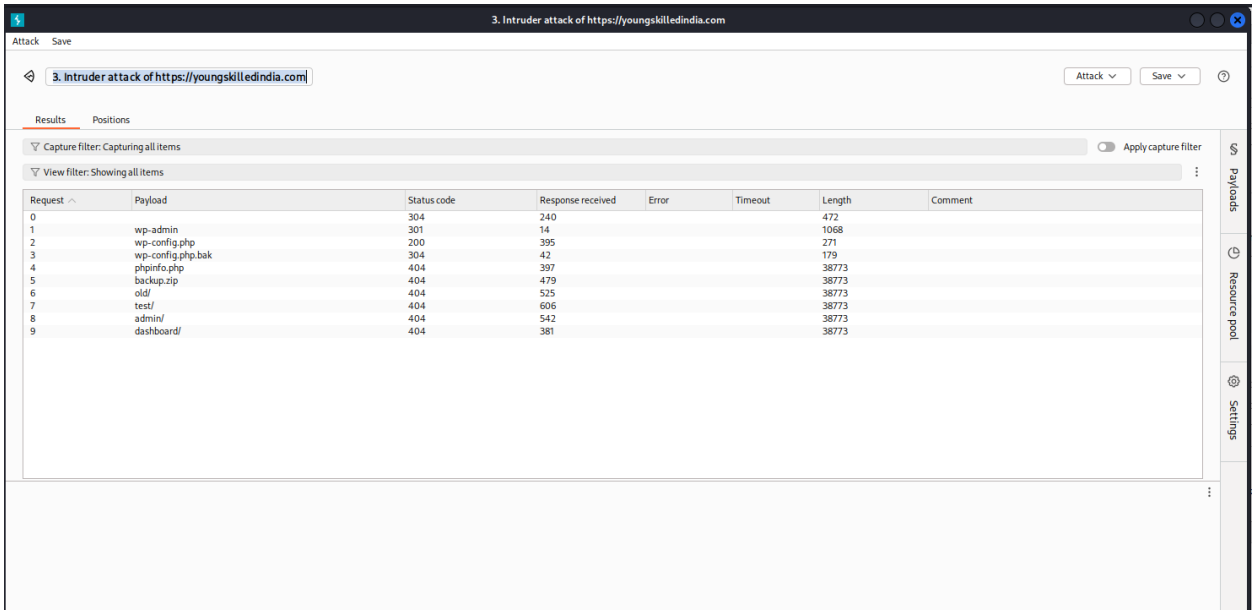
Recommendations

1. **Harden WordPress Installation:**
    - Ensure `.php` config files cannot be served directly (even as 200 with empty output).
    - Place sensitive files (`wp-config.php`) **outside web root** if possible.
2. **Restrict Directory Listing:**
    - Already configured for `/uploads/`. Apply to all directories.
3. **Backup/Temp File Controls:**
    - Ensure no `.bak`, `.zip`, `.tar.gz`, `.sql` files are ever stored on production.
4. **Use Security Plugins:**
    - Wordfence / iThemes Security for extra checks on exposed paths.
5. **Continuous Monitoring:**
    - Run regular scans for accidental exposures.

Proof of Concept (PoC)

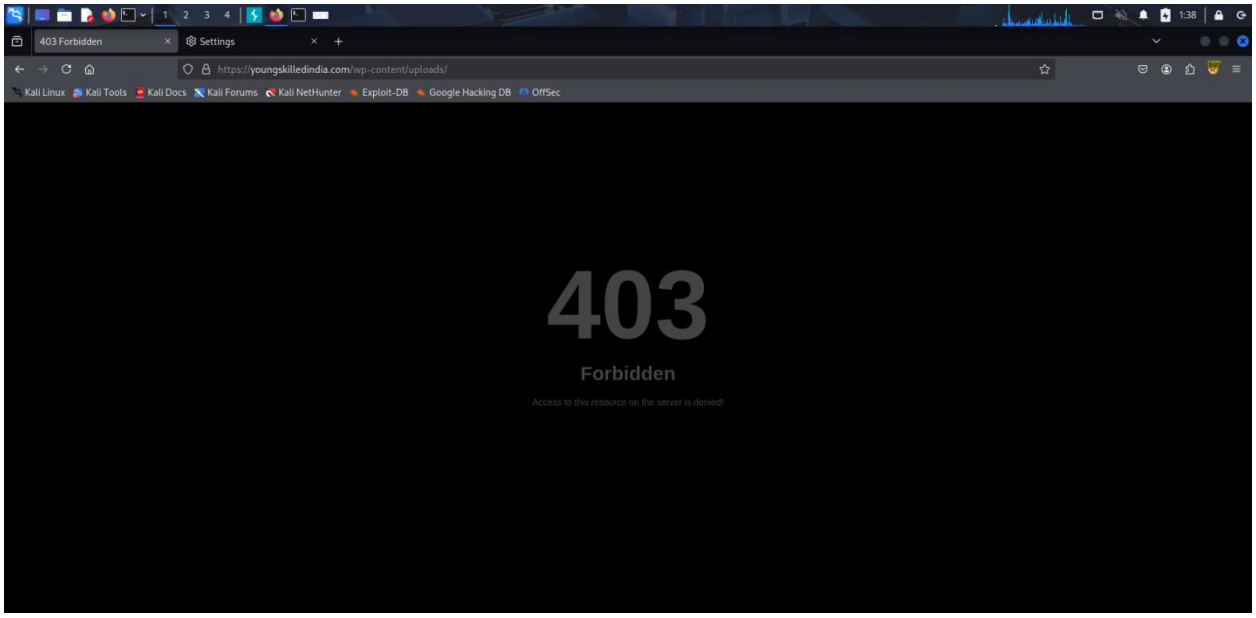
- **Screenshot 1:** Burp Intruder results showing status codes (200/301/403/404).

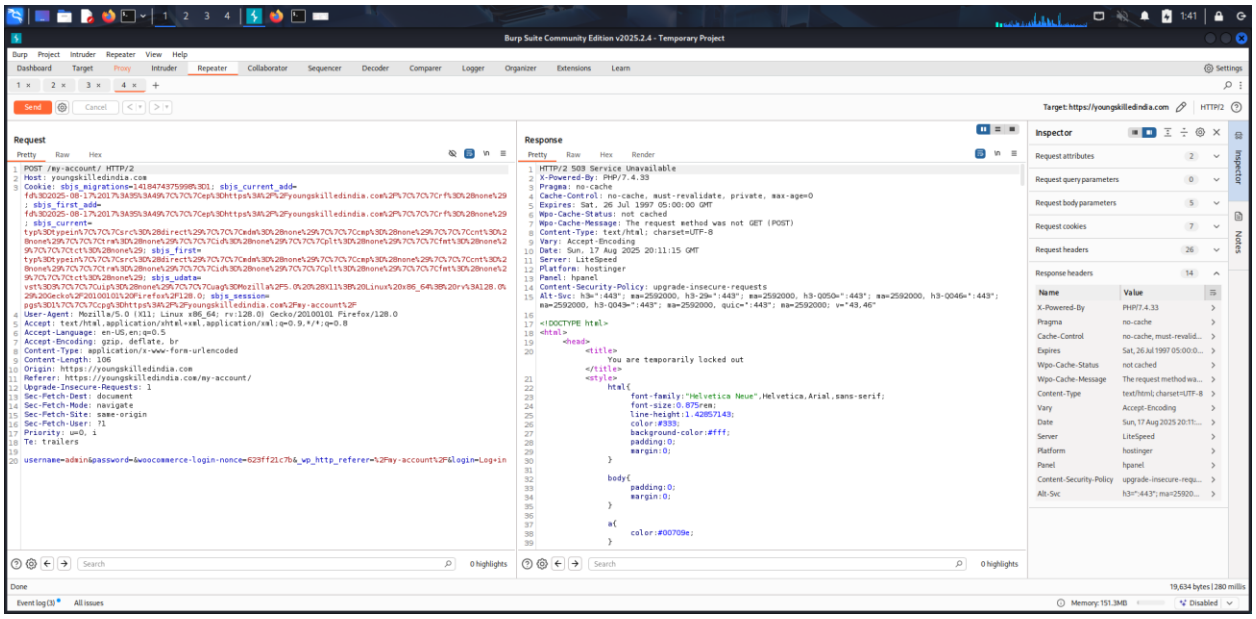




Request	Payload	Status code	Response received	Error	Timeout	Length	Comment
0		304	240			472	
1	wp-admin	301	14			1068	
2	wp-config.php	200	395			271	
3	wp-config.php.bak	304	42			179	
4	phpinfo.php	404	397			38773	
5	backup.zip	404	479			38773	
6	old/	404	525			38773	
7	test/	404	606			38773	
8	admin/	404	542			38773	
9	dashboard/	404	381			38773	

Screenshot 2: Browser view of `/wp-content/uploads/` returning `403 Forbidden`.





A06: Vulnerable & Outdated Components

Finding S/N: A06-001

CWE: CWE-1104: Use of Unmaintained Third-Party Components

Severity: High

CVSS v3.1 Vector:
`CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H`
→ **Score: 9.8 (Critical)**

Description

The target web application is powered by **WordPress CMS**. During component enumeration with WPScan and manual verification, the following outdated or unsupported components were identified:

- **WordPress Core**: 6.8.2 → *latest stable, secure* ✅
- **Theme (Astra 4.8.7)** → *Outdated* (latest version is **4.11.9**). Older versions may contain security vulnerabilities including Cross-Site Scripting (XSS) and template injection issues.
- **PHP Version**: `7.4.33` → *End-of-Life* (EOL since November 2022). Unsupported by vendor and no longer receives security patches.
- **Exposed File**: `/readme.html` publicly accessible, discloses WordPress version info.

Outdated or unsupported components are a common entry point for attackers. Public exploits are often available that allow attackers to achieve:

- Remote Code Execution (RCE)
- Privilege Escalation
- Information Disclosure
- Website Defacement or Malware Injection

Security Impact to Business

If exploited, these weaknesses could allow attackers to:

- Execute arbitrary code or upload web shells (via theme/plugin exploitation).
- Compromise customer data and payment details (WooCommerce risk).
- Deface the platform or host phishing/malware campaigns.
- Breach compliance obligations (GDPR/IT Act) and damage reputation.

This directly impacts:

- **Confidentiality** – exposure of user/course data.
- **Integrity** – modification of content, website defacement.
- **Availability** – potential site takedown or ransomware injection.

Steps to Reproduce

1. **Run WPScan enumeration** against the target:
2. `wpscan --url https://youngskilledindia.com --enumerate vp,vt,tt,u`
3. Observe results:
   - **Theme** detected: Astra `4.8.7` (outdated).
   - **Server header**: `x-powered-by: PHP/7.4.33`.
   - **File**: `https://youngskilledindia.com/readme.html` accessible.
4. Cross-check versions against:
   - WordPress.org Theme Changelog – Astra
   - PHP Supported Versions

Proof of Concept (PoC)

 Screenshot 1 – WPScan output showing Astra theme 4.8.7

```
[+] WordPress theme in use: astra
 | Location: https://youngskilledindia.com/wp-content/themes/astra/
 | Last Updated: 2025-08-11T00:00:00.000Z
 | Readme: https://youngskilledindia.com/wp-content/themes/astra/readme.txt
 | [!] The version is out of date, the latest version is 4.11.9
 | Style URL: https://youngskilledindia.com/wp-content/themes/astra/style.css
 | Style Name: Astra
 | Style URI: https://wpastra.com/
 | Description: Astra is fast, fully customizable & beautiful WordPress theme suitable for blog, personal por
tfolio, ...
 | Author: Brainstorm Force
 | Author URI: https://wpastra.com/about/?utm_source=theme_preview&utm_medium=author_link&utm_campaign=astra_
theme
 |
 | Found By: Urls In Homepage (Passive Detection)
 | Confirmed By: Urls In 404 Page (Passive Detection)
 |
 | Version: 4.8.7 (80% confidence)
 | Found By: Style (Passive Detection)
 |  - https://youngskilledindia.com/wp-content/themes/astra/style.css, Match: 'Version: 4.8.7'
```

 Screenshot 2 – Response headers disclosing PHP/7.4.33.

```
┌──(jesseverse㉿kali)-[~]
└─$ wpscan --url https://youngskilledindia.com/

_______________________________________________________________
         __          _______   _____
         \ \        / /  __ \ / ____|
          \ \  /\  / /| |__) | (___   ___  __ _ _ __   ®
           \ \/  \/ / |  ___/ \___ \ / __|/ _` | '_ \
            \  /\  /  | |     ____) | (__| (_| | | | |
             \/  \/   |_|    |_____/ \___|\__,_|_| |_|

         WordPress Security Scanner by the WPScan Team
                         Version 3.8.28
       Sponsored by Automattic - https://automattic.com/
       @_WPScan_, @ethicalhack3r, @erwan_lr, @firefart
_______________________________________________________________

[+] URL: https://youngskilledindia.com/ [195.35.44.229]
[+] Started: Mon Aug 18 02:02:20 2025

Interesting Finding(s):

[+] Headers
 | Interesting Entries:
 |  - x-powered-by: PHP/7.4.33
 |  - wpo-cache-status: cached
 |  - server: LiteSpeed
 |  - platform: hostinger
 |  - panel: hpanel
 |  - content-security-policy: upgrade-insecure-requests
 |  - alt-svc: h3=":443"; ma=2592000, h3-29=":443"; ma=2592000, h3-Q050=":443"; ma=2592000, h3-Q046=":443"; m
a=2592000, h3-Q043=":443"; ma=2592000, quic=":443"; ma=2592000; v="43,46"
 | Found By: Headers (Passive Detection)
 | Confidence: 100%

[+] robots.txt found: https://youngskilledindia.com/robots.txt
 | Interesting Entries:
 |  - /wp-content/uploads/wc-logs/
 |  - /wp-content/uploads/woocommerce_transient_files/
 |  - /wp-content/uploads/woocommerce_uploads/
 |  - /wp-admin/
 |  - /wp-admin/admin-ajax.php
 |  - /wp-content/uploads/wpo/wpo-plugins-tables-list.json
 | Found By: Robots Txt (Aggressive Detection)
 | Confidence: 100%
```

 Screenshot 3 – Browser access to `/readme.html`.

```
[+] WordPress readme found: https://youngskilledindia.com/readme.html
 | Found By: Direct Access (Aggressive Detection)
 | Confidence: 100%
```

Recommendations

1. **Immediate Upgrades**
   - Update **Astra theme** to the latest release (**≥4.11.9**).
   - Upgrade **PHP runtime** to **8.2 LTS** or higher (supported by WordPress).
2. **Remove/Restrict Sensitive Files**
   - Delete or restrict access to `/readme.html` and other default WordPress files.
3. **Ongoing Vulnerability Management**
   - Perform **monthly WPScan audits** of core, plugins, and themes.
   - Subscribe to WordPress Security advisories.
   - Enable **automatic minor/patch updates**.
4. **Web Application Firewall (WAF)**
   - Deploy ModSecurity/Cloudflare WAF rules to detect & block exploit attempts.

A07: Identification & Authentication Failures

Finding S/N: A07-001

CWE: CWE-287: Improper Authentication

Severity: Medium (if exploitable)

CVSS v3.1 Vector:
If login brute-force protection is absent:
`CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:L/A:N` → **Score: 6.5 (Medium)**

Description

The login endpoint at:
 `https://youngskilledindia.com/wp-login.php`
is responsible for authenticating WordPress users.

During testing with **Burp Suite Repeater & Intruder**, no visible error messages disclosed usernames or detailed error handling. However, it is essential to test whether the site has:

- Brute force protection (rate limiting, CAPTCHA, account lockout).
- Weak password policies.
- Username enumeration (different responses for valid/invalid users).
- Session handling issues (cookie flags, session expiration).

Security Impact to Business

If authentication failures exist:

- Attackers could brute force admin credentials.
- Weak passwords could allow unauthorized access to WordPress backend.
- Session mismanagement could let attackers hijack sessions.
- Business-critical data and site defacement risk increases.

Steps to Reproduce (Using Burp Suite)

1. **Intercept Login Request**
   - Open Burp → turn on proxy.
   - Go to: `https://youngskilledindia.com/wp-login.php`.
   - Enter dummy credentials (e.g., `admin:wrongpassword`).
   - Forward request → capture in Burp.
2. **Send to Repeater**
   - Right-click request → "Send to Repeater".
   - Change username/password values → observe if responses differ (username enumeration).
3. **Send to Intruder (Brute Force Simulation)**
   - Highlight username & password fields → Add payload positions.
   - Load small wordlist (like `admin, test, user` or `123456, password`).
   - Start attack → check if responses differ (length, status code, error message).
4. **Session Cookie Check**
   - After login attempt, check if `Set-Cookie` is marked as:
     - `Secure` (only over HTTPS).
     - `HttpOnly` (not accessible by JavaScript).
     - `SameSite` (protection from CSRF).

Recommendations

- Implement rate-limiting (e.g., fail2ban or Wordfence plugin).
- Add CAPTCHA on login page.
- Enforce strong password policy.
- Hide user enumeration responses (generic error message).
- Ensure cookies have `Secure` & `HttpOnly` flags.
- Enable multi-factor authentication (MFA).

A08: Software & Data Integrity Failures

Finding S/N: A08-001

CWE: CWE-494: Download of Code Without Integrity Check

Severity: Medium (if outdated or vulnerable plugins/themes found)

CVSS v3.1 Vector (if vulnerable plugin detected):
`CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:L/A:H` → **Score: 8.1 (High)**

Description

The application is running **WordPress** (CMS).
In WordPress, integrity failures occur when:

- Outdated WordPress core, plugins, or themes are used.
- No integrity checking of downloaded code (plugins/themes from untrusted sources).
- No hashing/signature verification of updates.
- Use of vulnerable third-party libraries (jQuery, PHP, etc.).

This category often leads to **supply chain attacks** (malicious plugins/themes) or exploitation of outdated versions.

Security Impact to Business

If plugins/themes are outdated or tampered:

- Attackers could upload backdoors (via vulnerable plugins).
- Site could be fully compromised and defaced.
- Customer data and credentials may be stolen.
- Loss of trust in business reputation.

▯ Steps to Reproduce (How to Test with Burp + WPScan)

1. Check HTTP Headers for Plugin References

- In Burp → review responses for:
  - `/wp-content/plugins/...`
  - `/wp-content/themes/...`
- Note plugin/theme names.

2. Run WPScan (WordPress Vulnerability Scanner)

In Kali Linux:

```
wpscan --url https://youngskilledindia.com/ -e ap,at,tt,u --api-token
YOUR_API_TOKEN
```

- `-e ap` = enumerate plugins
- `-e at` = enumerate themes
- `-e tt` = enumerate timthumbs
- `-e u` = enumerate usernames

It will show:
✅Installed plugins/themes
▯ Outdated/vulnerable versions
▯ Known CVEs (mapped automatically)

3. Verify Integrity

- If plugins/themes not updated, cross-check with official WordPress repo.
- If admin uploads unverified code, this is a risk.

▯ Recommendations

- Always update WordPress core, plugins, and themes.
- Install plugins/themes only from trusted sources (WordPress repo).
- Use file integrity monitoring (e.g., Wordfence).
- Enable automatic updates where possible.
- Remove unused/old plugins.
- Implement CI/CD integrity checks (signatures, checksums).

PoC (Proof of Concept for Report)

```
[i] No plugins Found.

[+] Enumerating Vulnerable Themes (via Passive and Aggressive Methods)
 Checking Known Locations - Time: 00:00:06 <==================================> (652 / 652) 100.00% Time: 00:00:06
[+] Checking Theme Versions (via Passive and Aggressive Methods)

[i] No themes Found.

[+] Enumerating Timthumbs (via Passive and Aggressive Methods)
 Checking Known Locations - Time: 00:00:14 <==================================> (2575 / 2575) 100.00% Time: 00:00:14
```

7. Recommendations

To strengthen the security posture of YoungSkilledIndia.com, the following measures are strongly recommended:

1. **Patch Management**
   - Upgrade PHP to version **8.2 LTS or higher**.
   - Update Astra theme and all WordPress components to latest stable versions.
   - Remove unused plugins and themes.
2. **Authentication & Access Control**
   - Enforce **Multi-Factor Authentication (MFA)** for administrative accounts.
   - Implement **rate-limiting and CAPTCHA** for repeated failed logins.
   - Adopt **progressive lockouts** to deter brute-force attacks.
3. **Configuration Hardening**
   - Restrict access to sensitive files (wp-config.php, backups, /readme.html).
   - Disable directory listing for all folders.
   - Regularly audit WordPress and server configurations.
4. **Network & Application Security**
   - Deploy a **Web Application Firewall (WAF)** such as ModSecurity or Cloudflare.
   - Conduct periodic vulnerability scans and penetration tests.
5. **Monitoring & Incident Response**
   - Enable detailed logging of authentication attempts.
   - Monitor for anomalous behavior (multiple failed logins, strange IPs).
   - Create an incident response plan to address potential breaches.

8. Conclusion

The penetration test of **YoungSkilledIndia.com** revealed several important findings that, if left unaddressed, could be leveraged by attackers to compromise the application and its users.

The most critical issues stem from outdated software components (PHP runtime and WordPress theme), weak authentication design, and absence of preventive controls such as WAF and MFA. While no direct data breach was observed during this assessment, the identified risks could lead to severe consequences if exploited by malicious actors.

By implementing the recommended remediations and adopting a proactive **security maintenance program**, YoungSkilledIndia.com can significantly reduce its exposure to cyber threats and ensure a safer experience for its users.

9. Appendices

The appendices provide supporting evidence and raw data from the assessment, including:

- **OWASP WSTG Checklist** (completed for the target application)
- **Nmap scan results** (open ports, services, SSL/TLS details)
- **Burp Suite request/response captures**
- **WPScan enumeration output** (plugins, themes, versions)
- **Screenshots of Proof-of-Concepts** (where safe and non-destructive)
- **References & Tools Documentation**