IBM Innovation Center for Education

PHEMESOFT Software That Matters

TRUSTED DELIVERY PARTNER for IBM ICE

Yenepoya Institute of Arts, Science, Management & commerce

Final Project Report

On

Vulnerability Assessment of a Web Application

Team members:

Abhiraj A -22BCACDA04

Daivik Rajesh -22BCACDC16

Muhammed Hanin Alfas M -22BCACDC43

Robin Puthuparampil Roy-22BCACDC59

Vaishnavi-22BSCFDC45

GUIDED BY

SASHANK

Table of Contents

Executive Summary

1. Background

Cybersecurity is crucial for all organizations, including small businesses that often lack advanced security infrastructure. This project focuses on identifying and analyzing web vulnerabilities on real-world websites to help improve their security posture.

1.1 Aim

To conduct ethical web vulnerability testing on approved websites to identify and report potential security flaws using industry-standard tools and techniques.

1.2 Technologies

1. Burp Suite – A powerful web vulnerability scanner and proxy tool used for intercepting and testing HTTP/HTTPS requests.
2. OWASP ZAP – An open-source tool for finding security vulnerabilities in web applications through automated scanning.
3. Nmap – A network scanning tool used to discover hosts and services on a computer network.
4. Whois Lookup – A command-line tool to retrieve domain registration and ownership information.
5. SSL Labs – An online service that analyzes the SSL configuration of a website and grades its security.
6. SecurityHeaders.com – A free tool that scans a website for HTTP response headers related to security best practices.

1.3 Hardware Architecture

1. Laptop with at least Intel Core i5 Processor – Required to run vulnerability scanning tools smoothly without performance lag.
2. Minimum 8GB RAM – Ensures stable operation of resource-intensive applications like Burp Suite and OWASP ZAP.
3. Stable Internet Connection – Needed for live scanning of web applications and accessing online tools such as SSL Labs or Whois.

.4 Software Architecture

1. Kali Linux/Windows OS – Operating systems used to host and run security tools; Kali is tailored for penetration testing, while Windows ensures broader compatibility.
2. Burp Suite – A web security testing platform used for scanning vulnerabilities, intercepting requests, and testing input fields.
3. ZAP Proxy (OWASP ZAP) – A proxy-based tool for automating vulnerability scans and analyzing application traffic.
4. Web Browsers – Used to manually navigate and test target websites, observe frontend behaviors, and replicate vulnerabilities.
5. Command-line Utilities – Tools like whois, nslookup, curl, and dig used for gathering domain information, testing endpoints, and analyzing configurations.

2.System Requirements

2.1 **Requirements**

2.1.1 **Functional Requirements**

1. Perform scanning using Burp Suite and OWASP ZAP
2. Identify OWASP Top 10 vulnerabilities
3. Document findings with screenshots and evidence
4. Recommend mitigation steps

2.1.2 User Requirements

1. Easy-to-understand vulnerability report
2. Recommendations for non-technical users
3. Legal and authorized scope of testing

2.1.3 Environmental Requirements

1. Stable Internet – Essential for performing live vulnerability scans, accessing online analysis tools, and downloading necessary updates or modules.
2. Access to Authorized Websites – Testing must only be performed on websites for which explicit permission has been granted to ensure legality and avoid misuse.
3. Ethical Boundaries and Legal Permissions – All testing activities must follow ethical hacking principles and operate within the scope defined by legal agreements and permissions.

2.2 Design and Architecture

System Design

The design of the testing process followed a structured and ethical methodology:

1. Reconnaissance: Identifying the target scope and collecting domain and hosting information.
2. Application Mapping: Exploring and documenting all available pages, forms, and input vectors.
3. Request Interception: Using proxy tools to observe how data is transmitted and processed.

4. Fuzzing and Manual Testing: Supplying crafted inputs to input fields to check for improper validation and vulnerabilities.
5. Scanning Strategy: Combining both passive and active scanning techniques to uncover weaknesses with minimal disruption.

System Architecture

The architecture refers to the setup of the tools and systems used during testing:

1. Client System: Laptop with Kali Linux or Windows OS acting as the main testing environment.
2. Proxy Tools: Burp Suite and OWASP ZAP configured to intercept and modify web traffic.
3. Online Resources: External tools like SSL Labs and SecurityHeaders.com accessed through the browser.
4. Command-line Interface: Used for running DNS, WHOIS, and port scan commands for deeper inspection.

2.3 Implementation

Implementation Steps

The implementation phase involved the practical execution of scanning and analysis using selected tools and methods. Key activities included:

1. Passive Scanning using OWASP ZAP:
   Conducted automated scanning without altering server state to identify issues like missing headers, insecure cookies, and outdated libraries.

2. Manual Testing with Burp Suite:
   Intercepted and modified requests to test for vulnerabilities such as XSS (Cross-Site Scripting), SQL Injection, and CSRF (Cross-Site Request Forgery).

3. SSL and Header Analysis using Online Tools:
   Used platforms like SSL Labs and SecurityHeaders.com to evaluate HTTPS configuration, certificate validity, and presence of security headers (e.g., CSP, HSTS).

4. WHOIS and DNS Information Collection:
   Gathered domain registration details, name server records, and IP mappings to understand domain ownership and potential exposure.

5. Report Generation:
   Documented all findings with screenshots, risk ratings, and remediation suggestions to assist the website owner in improving security.

3. Testing

3.1 Test Plan Objectives

The primary objective of the testing phase was to evaluate the security posture of the target websites through structured vulnerability assessment. Key goals included:

1. Identify Critical Vulnerabilities
   Detect high-impact issues such as XSS, SQL Injection, and insecure configurations.
2. Assess Severity
   Categorize each finding based on risk level using CVSS standards.
3. Provide Actionable Recommendations
   Suggest clear, practical fixes for each issue to help improve website security.
4. Maintain Ethical Scope
   Ensure testing stays within legal and authorized boundaries.
5. Document Findings
   Record results with screenshots and logs for clarity and reporting.

3.2 Test Strategy

Testing Approach and Prioritization:

1. Use Both Automated and Manual Testing
   Automated tools like OWASP ZAP were used for broad scanning, while manual techniques with Burp Suite helped verify and exploit specific vulnerabilities.
2. OWASP Top 10 as Benchmark
   The testing focused on identifying common and critical web application flaws based on the OWASP Top 10 list (e.g., Injection, Broken Access Control, Security Misconfigurations).
3. Prioritize Based on CVSS Score
   Each vulnerability was evaluated and ranked using the Common Vulnerability Scoring System (CVSS) to assess its impact and urgency.

3.3 System Test

System testing validated the SIEM solution's functionality, including:

- Log collection and forwarding.
- Threat detection and alerting.
- Data visualization and reporting.

Results:

- The system successfully collected and forwarded logs.
- Threats were detected and alerted in a timely manner.
- Data visualization provided valuable insights.

3.4 Security Test

Vulnerability Detection and Analysis:

1. Targeted URLs:
   https://christhujyothi.com/
   https://demotestfire.net/
2. Tools Used:
   OWASP ZAP, Burp Suite, SecurityHeaders.com, SSL Labs
3. Key Results:
   ▪ Identified reflected XSS vulnerability in input fields
   ▪ Detected outdated JavaScript libraries and exposed server information
   ▪ Missing critical security headers like Content-Security-Policy (CSP) and X-Frame-options
   ▪ Insecure cookies and lack of proper SSL configuration observed on one of the targets

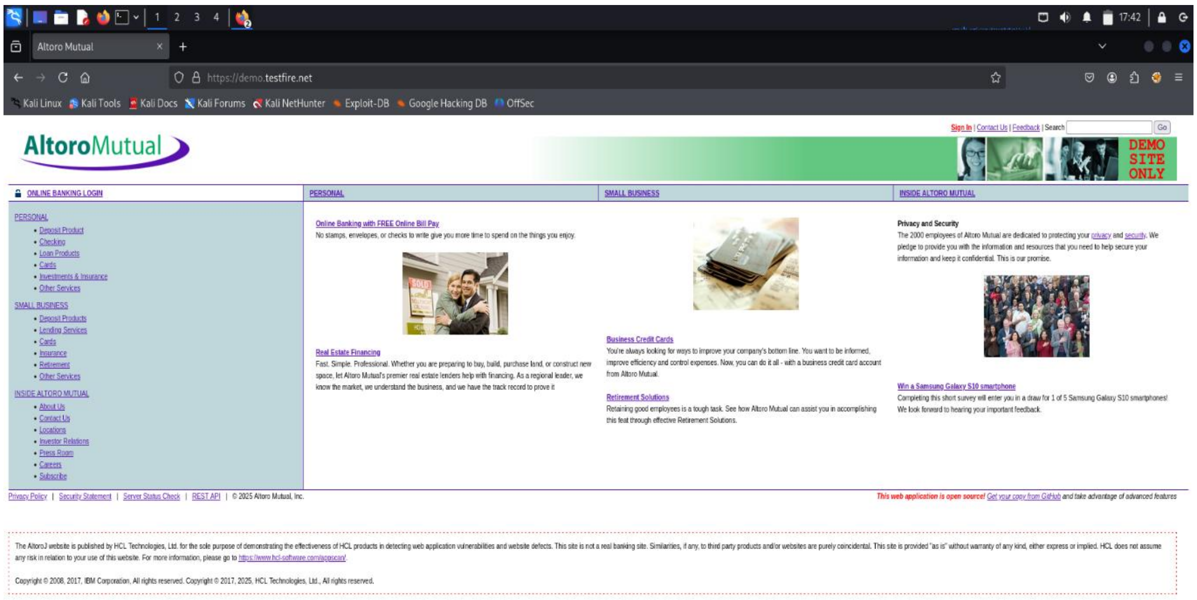
3.5 User Acceptance Test

Report Delivery, Validation, and Stakeholder Response:

1. Comprehensive reports containing identified vulnerabilities, impact summaries, evidence (screenshots/logs), and recommended fixes were submitted to the respective site owners.
2. The reports were reviewed and accepted by the site administrators, confirming the authenticity and relevance of the findings.

3. Critical issues such as missing security headers, exposed server information, and outdated components were acknowledged and prioritized for resolution.
4. Communication channels were maintained via email and meetings to clarify technical details and provide guidance on remediation steps.
5. Follow-up interactions confirmed that some security improvements had already been applied (e.g., CSP headers added, SSL settings improved).
6. The feedback process helped validate the effectiveness of the testing approach and contributed to real-world improvements in the target sites' security posture.

4. Snapshots of the Project



Altoro Mutual

https://demo.testfire.net

Kali Linux | Kali Tools | Kali Docs | Kali Forums | Kali NetHunter | Exploit-DB | Google Hacking DB | OffSec

AltoroMutual

Sign In | Contact Us | Feedback | Search Go

DEMO SITE ONLY

ONLINE BANKING LOGIN | PERSONAL | SMALL BUSINESS | INSIDE ALTORO MUTUAL

PERSONAL
- Deposit Product
- Checking
- Loan Products
- Cards
- Investments & Insurance
- Other Services

SMALL BUSINESS
- Deposit Products
- Lending Services
- Cards
- Insurance
- Retirement
- Other Services

INSIDE ALTORO MUTUAL
- About Us
- Contact Us
- Locations
- Investor Relations
- Press Room
- Careers
- Subscribe

Online Banking with FREE Online Bill Pay
No stamps, envelopes, or checks to write give you more time to spend on the things you enjoy.

Real Estate Financing
Fast. Simple. Professional. Whether you are preparing to buy, build, purchase land, or construct new space, let Altoro Mutual's premier real estate lenders help with financing. As a regional leader, we know the market, we understand the business, and we have the track record to prove it

Business Credit Cards
You're always looking for ways to improve your company's bottom line. You want to be informed, improve efficiency and control expenses. Now, you can do it all - with a business credit card account from Altoro Mutual.

Retirement Solutions
Retaining good employees is a tough task. See how Altoro Mutual can assist you in accomplishing this feat through effective Retirement Solutions.

Privacy and Security
The 2000 employees of Altoro Mutual are dedicated to protecting your privacy and security. We pledge to provide you with the information and resources that you need to help secure your information and keep it confidential. This is our promise.

Win a Samsung Galaxy S10 smartphone
Completing this short survey will enter you in a draw for 1 of 5 Samsung Galaxy S10 smartphones! We look forward to hearing your important feedback.

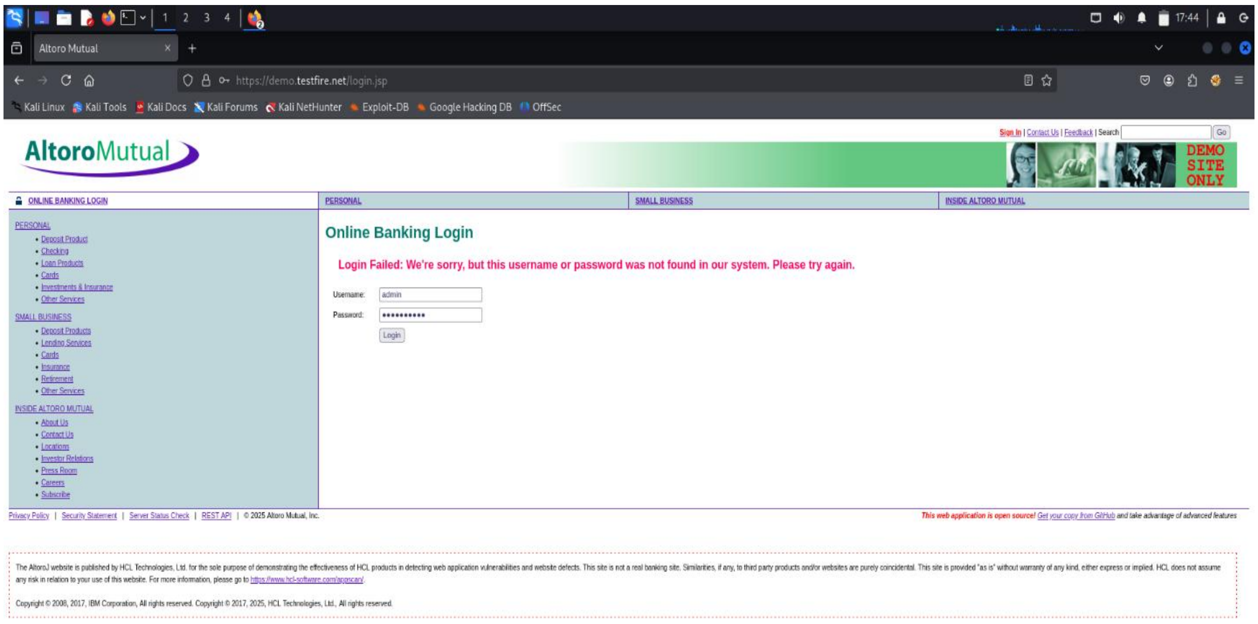
Privacy Policy | Security Statement | Server Status Check | REST API | © 2025 Altoro Mutual, Inc.

This web application is open source! Get your copy from GitHub and take advantage of advanced features

The AltoroJ website is published by HCL Technologies, Ltd. for the sole purpose of demonstrating the effectiveness of HCL products in detecting web application vulnerabilities and website defects. This site is not a real banking site. Similarities, if any, to third party products and/or websites are purely coincidental. This site is provided "as is" without warranty of any kind, either express or implied. HCL does not assume any risk in relation to your use of this website. For more information, please go to https://www.hcl-software.com/appscan/.

Copyright © 2008, 2017, IBM Corporation, All rights reserved. Copyright © 2017, 2025, HCL Technologies, Ltd., All rights reserved.

LOGIN PAGE OF THE TESTING SITE:



Altoro Mutual

https://demo.testfire.net/login.jsp

Kali Linux | Kali Tools | Kali Docs | Kali Forums | Kali NetHunter | Exploit-DB | Google Hacking DB | OffSec

AltoroMutual

Sign In | Contact Us | Feedback | Search Go

DEMO SITE ONLY

ONLINE BANKING LOGIN | PERSONAL | SMALL BUSINESS | INSIDE ALTORO MUTUAL

PERSONAL
- Deposit Product
- Checking
- Loan Products
- Cards
- Investments & Insurance
- Other Services

SMALL BUSINESS
- Deposit Products
- Lending Services
- Cards
- Insurance
- Retirement
- Other Services

INSIDE ALTORO MUTUAL
- About Us
- Contact Us
- Locations
- Investor Relations
- Press Room
- Careers
- Subscribe

Online Banking Login

Login Failed: We're sorry, but this username or password was not found in our system. Please try again.

Username: admin
Password: ••••••••••
Login

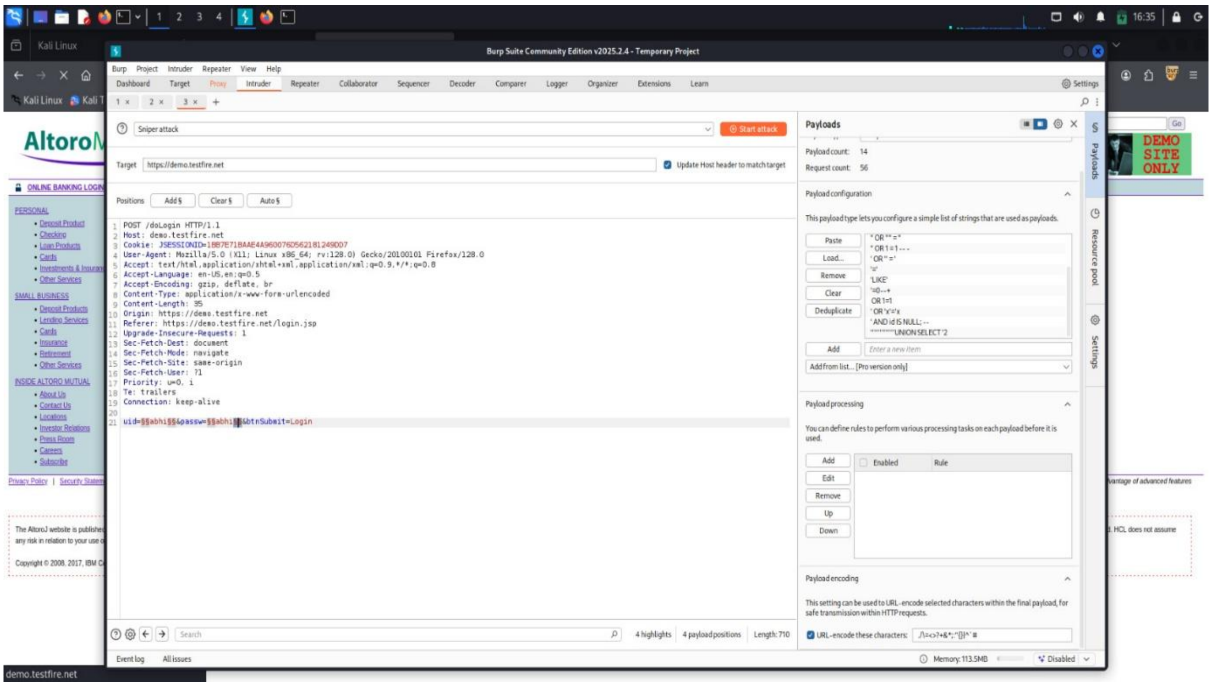
Privacy Policy | Security Statement | Server Status Check | REST API | © 2025 Altoro Mutual, Inc.

This web application is open source! Get your copy from GitHub and take advantage of advanced features

The AltoroJ website is published by HCL Technologies, Ltd. for the sole purpose of demonstrating the effectiveness of HCL products in detecting web application vulnerabilities and website defects. This site is not a real banking site. Similarities, if any, to third party products and/or websites are purely coincidental. This site is provided "as is" without warranty of any kind, either express or implied. HCL does not assume any risk in relation to your use of this website. For more information, please go to https://www.hcl-software.com/appscan/.

Copyright © 2008, 2017, IBM Corporation, All rights reserved. Copyright © 2017, 2025, HCL Technologies, Ltd., All rights reserved.

SQL INJECTION ATTCK USING BURP SUITE:


Burp Suite Community Edition v2025.2.4 - Temporary Project
Burp Project Intruder Repeater View Help
Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Extensions Learn
Start attack
Positions Add § Clear § Auto §
Payloads
Paste
Load...
Remove
Clear
Deduplicate
Add
Payload processing
Payload encoding
Event log All issues
demo.testfire.net

Sniper attack

Target https://demo.testfire.net

Update Host header to match target

```
POST /doLogin HTTP/1.1
Host: demo.testfire.net
Cookie: JSESSIONID=1BB7E71BAAE4A960076D562181249DD7
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: 35
Origin: https://demo.testfire.net
Referer: https://demo.testfire.net/login.jsp
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
Connection: keep-alive

uid=§§abhi§§&passw=§§abhi§§&btnSubmit=Login
```

Payload count: 14
Request count: 56

Payload configuration

This payload type lets you configure a simple list of strings that are used as payloads.

```
" OR "" = "
" OR 1 = 1 - - -
' OR '' = '
'='
'LIKE'
'=0--+
 OR 1=1
' OR 'x'='x
' AND id IS NULL; --
'''''''''''''UNION SELECT '2
```

Add from list... [Pro version only]

You can define rules to perform various processing tasks on each payload before it is used.

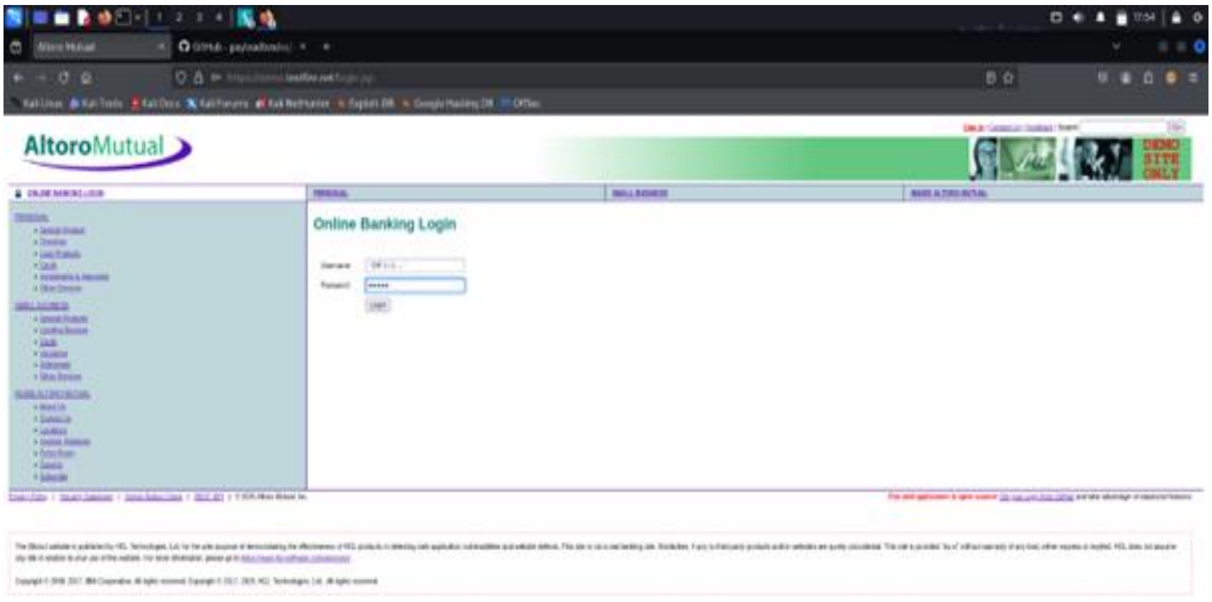
This setting can be used to URL-encode selected characters within the final payload, for safe transmission within HTTP requests.

URL-encode these characters: ./\=<>?+&*;:"{}|^`#

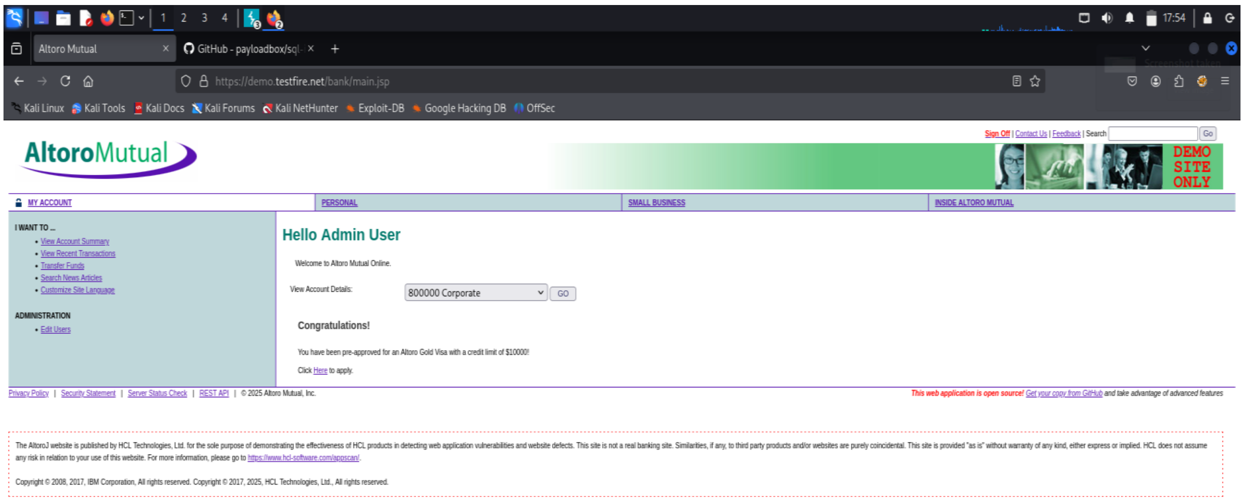
4 highlights 4 payload positions Length: 710

Memory: 113.5MB Disabled

SQL INJECTION PERFORMING ON INPUT FIELD:



LOGGED INTO USER ACCOUNT:



Altoro Mutual | GitHub - payloadbox/sql-

https://demo.testfire.net/bank/main.jsp

Sign Off | Contact Us | Feedback | Search

MY ACCOUNT | PERSONAL | SMALL BUSINESS | INSIDE ALTORO MUTUAL

I WANT TO ...
- View Account Summary
- View Recent Transactions
- Transfer Funds
- Search News Articles
- Customize Site Language

ADMINISTRATION
- Edit Users

Hello Admin User

Welcome to Altoro Mutual Online.

View Account Details: 800000 Corporate GO

Congratulations!

You have been pre-approved for an Altoro Gold Visa with a credit limit of $10000!

Click Here to apply.

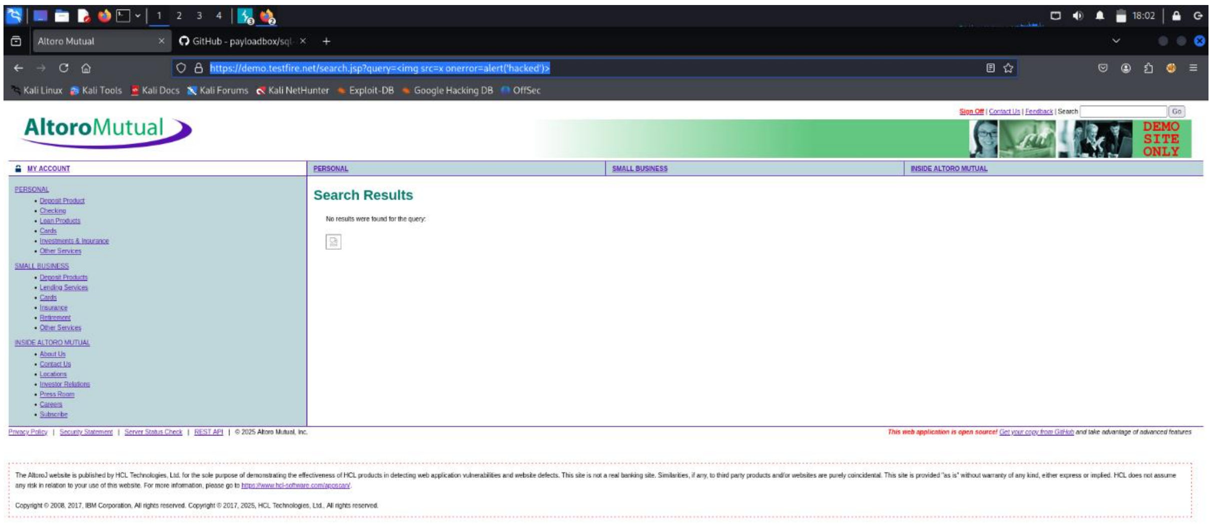
Privacy Policy | Security Statement | Server Status Check | REST API | © 2025 Altoro Mutual, Inc.

This web application is open source! Get your copy from GitHub and take advantage of advanced features

The AltoroJ website is published by HCL Technologies, Ltd. for the sole purpose of demonstrating the effectiveness of HCL products in detecting web application vulnerabilities and website defects. This site is not a real banking site. Similarities, if any, to third party products and/or websites are purely coincidental. This site is provided "as is" without warranty of any kind, either express or implied. HCL does not assume any risk in relation to your use of this website. For more information, please go to https://www.hcl-software.com/appscan/.

Copyright © 2008, 2017, IBM Corporation, All rights reserved. Copyright © 2017, 2025, HCL Technologies, Ltd., All rights reserved.

XSS ATTACK PERFORMING ON URL INPUT FIELD:



Altoro Mutual | GitHub - payloadbox/sql-

```
https://demo.testfire.net/search.jsp?query=<img src=x onerror=alert('hacked')>
```

Sign Off | Contact Us | Feedback | Search

MY ACCOUNT | PERSONAL | SMALL BUSINESS | INSIDE ALTORO MUTUAL

PERSONAL
- Deposit Product
- Checking
- Loan Products
- Cards
- Investments & Insurance
- Other Services

SMALL BUSINESS
- Deposit Products
- Lending Services
- Cards
- Insurance
- Retirement
- Other Services

INSIDE ALTORO MUTUAL
- About Us
- Contact Us
- Locations
- Investor Relations
- Press Room
- Careers
- Subscribe

Search Results

No results were found for the query:

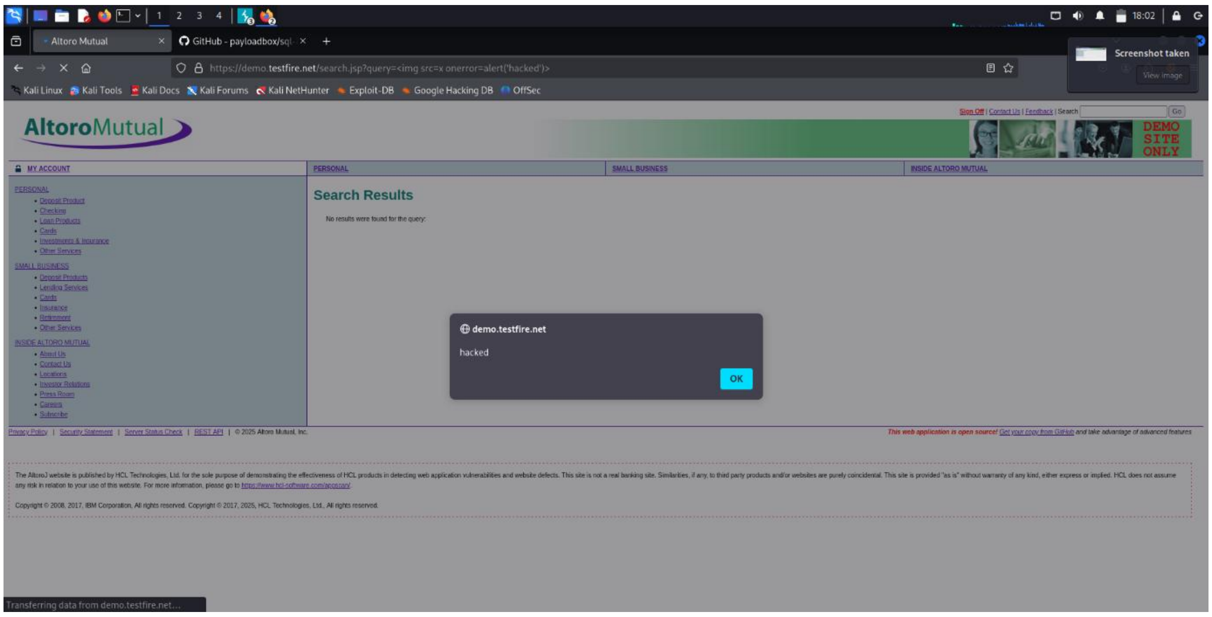
Privacy Policy | Security Statement | Server Status Check | REST API | © 2025 Altoro Mutual, Inc.

This web application is open source! Get your copy from GitHub and take advantage of advanced features

The AltoroJ website is published by HCL Technologies, Ltd. for the sole purpose of demonstrating the effectiveness of HCL products in detecting web application vulnerabilities and website defects. This site is not a real banking site. Similarities, if any, to third party products and/or websites are purely coincidental. This site is provided "as is" without warranty of any kind, either express or implied. HCL does not assume any risk in relation to your use of this website. For more information, please go to https://www.hcl-software.com/appscan/.

Copyright © 2008, 2017, IBM Corporation. All rights reserved. Copyright © 2017, 2025, HCL Technologies, Ltd.. All rights reserved.

REFELCTED XSS ATTACK POSSIBLE:



https://demo.testfire.net/search.jsp?query=<img src=x onerror=alert('hacked')>

Search Results

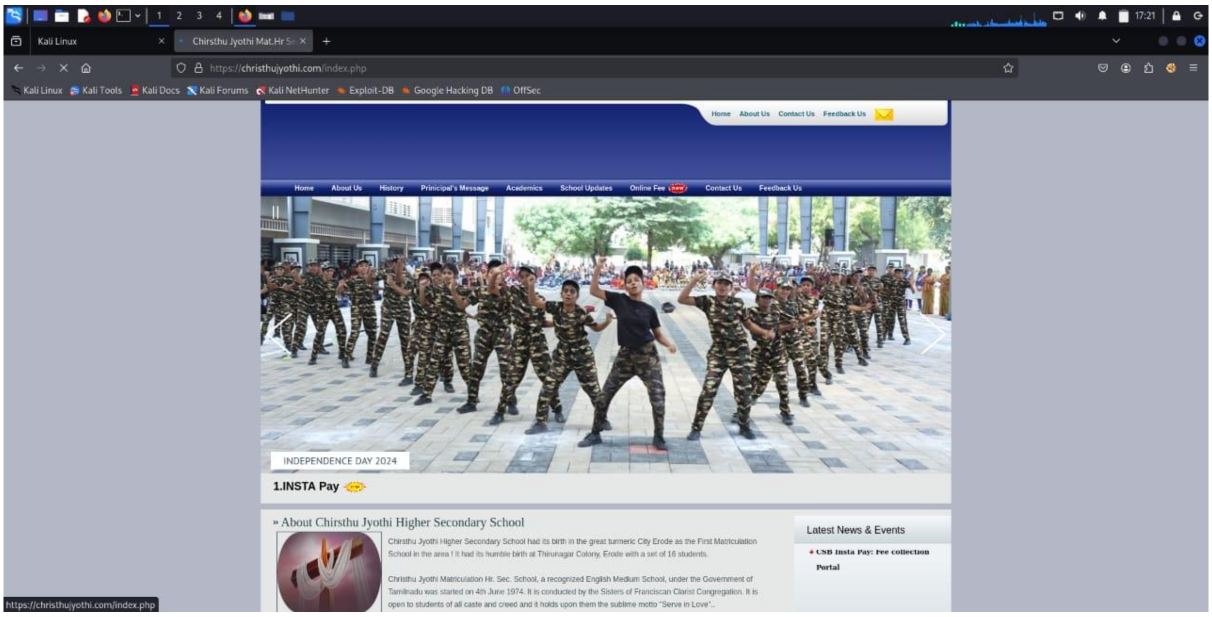
No results were found for the query:

demo.testfire.net

hacked

OK

HOME PAGE OF SECOND WEB SITE:



https://christhujyothi.com/index.php

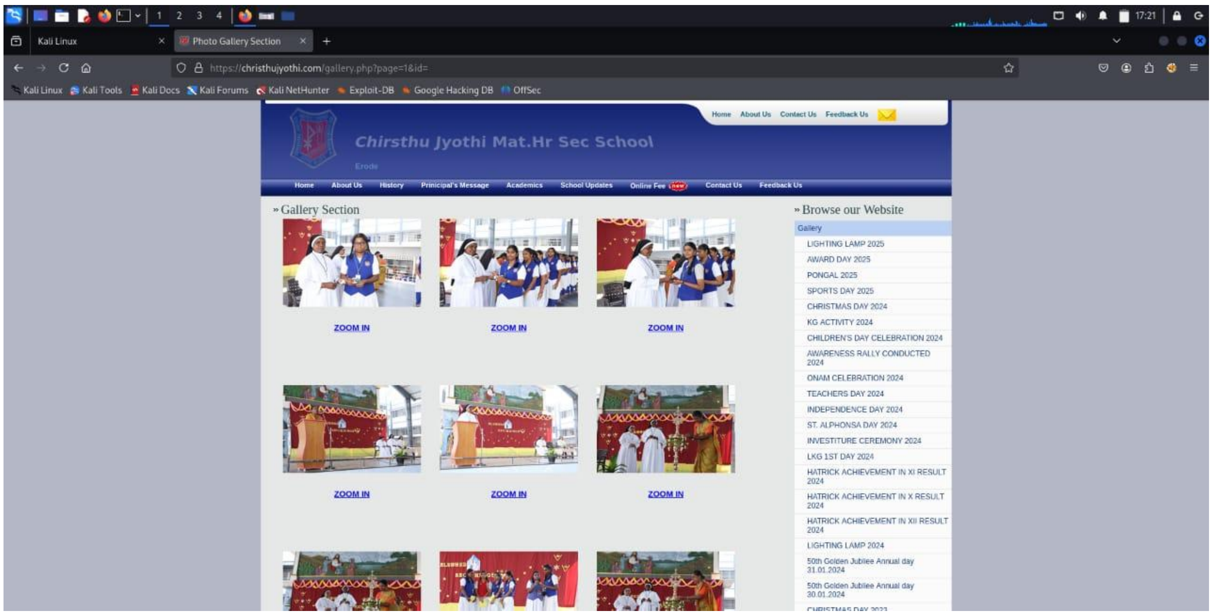
INDEPENDENCE DAY 2024

1.INSTA Pay

» About Chirsthu Jyothi Higher Secondary School

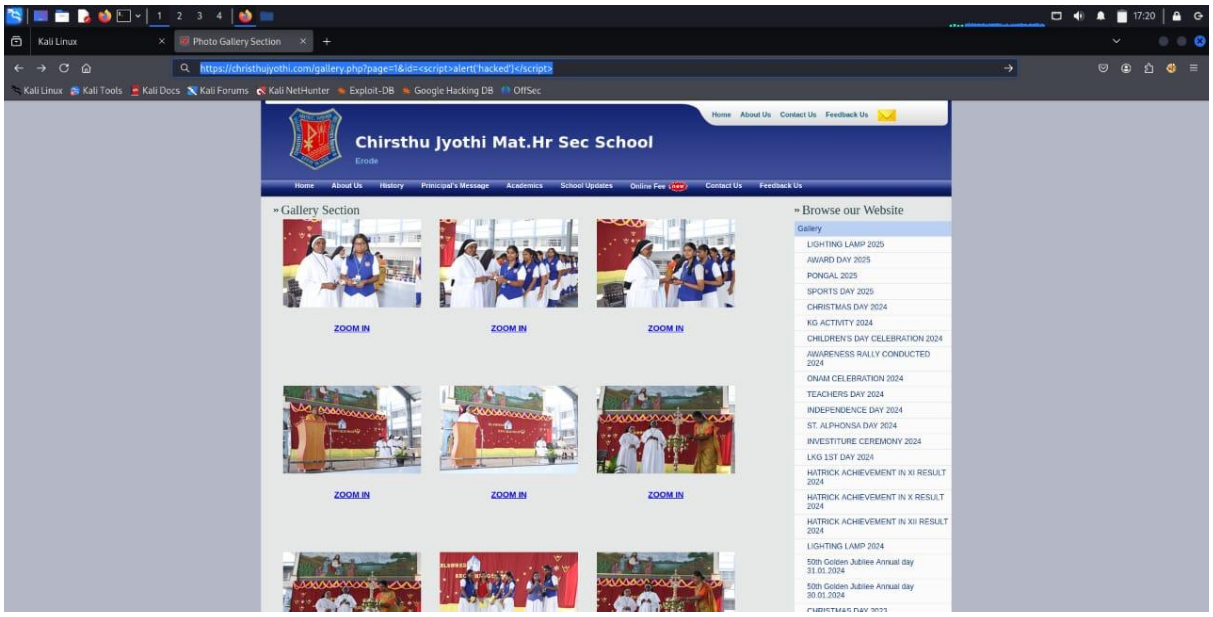
Latest News & Events

GALLERY SECTION IS THE VULNERABLE AREA OF THIS SITE:



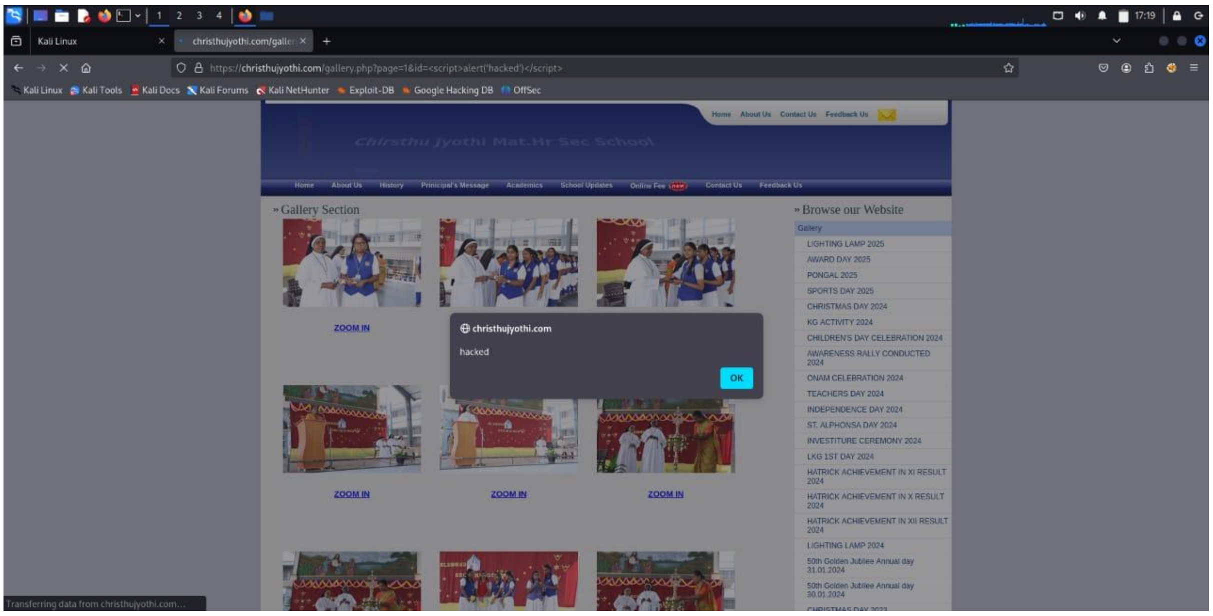
```
https://christhujyothi.com/gallery.php?page=1&id=
```

PERFORMING XSS ATTACK ON URL INPUT FIELD:



```
https://christhujyothi.com/gallery.php?page=1&id=<script>alert('hacked')</script>
```

REFELCTED XSS ATTACK POSSIBLE IN THIS WEB SITE:



```
https://christhujyothi.com/gallery.php?page=1&id=<script>alert('hacked')</script>
```

```
christhujyothi.com
hacked
OK
```

5. Conclusion

The project successfully identified and reported web vulnerabilities in real-world websites within an ethical and legally permitted framework. Through the use of tools such as Burp Suite, OWASP ZAP, and other reconnaissance utilities, several security flaws were discovered, including missing security headers, outdated components, and potential cross-site scripting (XSS) vulnerabilities.

This hands-on experience has significantly enhanced both theoretical and practical knowledge in the field of cybersecurity, particularly in web application security testing. It provided an opportunity to apply classroom concepts in a real-world environment, helping to understand how attackers exploit weaknesses and how such risks can be mitigated.

Moreover, the project strengthened skills in vulnerability assessment, ethical hacking practices, and professional reporting. It also emphasized the importance of responsible disclosure and communication with stakeholders to improve the overall security posture of web applications.

In conclusion, this project serves as a strong foundation for a future career in cybersecurity, especially in roles such as SOC Analyst, Penetration Tester, or Vulnerability Assessor.

6. Further Development or Research

To enhance the effectiveness and scalability of web vulnerability testing, the following areas can be explored for further development or research:

1. Automated CI/CD Vulnerability Scanning
Integrate security testing tools within the Continuous Integration/Continuous Deployment (CI/CD) pipeline to automatically scan code and applications for vulnerabilities before deployment.

2. Integration of DAST Tools in Development
Incorporate Dynamic Application Security Testing (DAST) tools during the development phase to identify runtime vulnerabilities in real-time, thereby reducing the cost and effort of late-stage fixes.

3. Custom Vulnerability Reporting Dashboard
Develop a centralized dashboard that consolidates vulnerability reports, severity levels, timestamps, and remediation status, making it easier for development and security teams to collaborate.

4. Threat Intelligence Integration
Leverage threat intelligence feeds to correlate discovered vulnerabilities with active exploits in the wild, helping prioritize remediation based on real-world threat levels.

5. Machine Learning for Anomaly Detection
Explore the use of machine learning models to detect unusual behavior patterns in web applications that may indicate zero-day vulnerabilities or unknown threats.

6. Automated Report Generation
Create automated scripts or tools to generate detailed vulnerability reports with screenshots, descriptions, and suggested mitigation steps to improve efficiency and consistency.

7. Security Awareness for Developers
Recommend regular security training and awareness programs for development teams to reduce the introduction of common vulnerabilities at the coding stage.

8. Integration with Bug Bounty Platforms
Study the feasibility of integrating internal testing efforts with external bug bounty platforms to encourage broader vulnerability discovery and community collaboration.

9. Performance Impact Analysis of Security Tools
Investigate how security tools and practices affect the performance and usability of web applications, ensuring a balance between security and user experience.

7. References

1. OWASP Foundation.
OWASP Top 10 – The Ten Most Critical Web Application Security Risks.
Available at: https://owasp.org/www-project-top-ten/

2. PortSwigger.
Burp Suite Documentation.
Available at: https://portswigger.net/burp/documentation

3. Mozilla Observatory.
Web Security Scanning Tool by Mozilla.
Available at: https://observatory.mozilla.org/

4. Qualys SSL Labs.
SSL Server Test – Analyze SSL/TLS configurations.
Available at: https://www.ssllabs.com/ssltest/

5. Web Security Academy by PortSwigger.
Interactive Learning Platform for Web Security.
Available at: https://portswigger.net/web-security

6. MITRE Corporation.
Common Vulnerabilities and Exposures (CVE) Database.
Available at: https://cve.mitre.org/

7. National Institute of Standards and Technology (NIST).
NVD – National Vulnerability Database.

Available at: https://nvd.nist.gov/

8. Cybersecurity & Infrastructure Security Agency (CISA).
Alerts and Guidance on Emerging Threats.
Available at: https://www.cisa.gov/

9. Acunetix Blog.
Web Application Security Tutorials and Vulnerability Insights.
Available at: https://www.acunetix.com/blog/

10. Hack The Box & TryHackMe Labs.
Practical Platforms for Hands-on Cybersecurity Training.
Available at:
https://www.hackthebox.com/

https://tryhackme.com/

8. Appendix

A. Glossary of Terms

XSS: Cross-Site Scripting – allows attackers to inject scripts into web pages.

SQLi: SQL Injection – malicious SQL statements to access or modify databases.

CSRF: Cross-Site Request Forgery – tricks users into executing unwanted actions.

CSP: Content Security Policy – protects against XSS by controlling content sources.

B. Sample Vulnerability Report (Summary)

Vulnerability: Reflected XSS

URL: https://example.com/search

Severity: Medium

Mitigation: Encode user input, use CSP headers

Proof: Screenshot and intercepted HTTP request

C. Tool Configurations (Summary)

Burp Suite: Scope set, active scan used, Repeater for testing payloads.

OWASP ZAP: Passive + active scan enabled, spidering for link discovery.

Nmap: Used with --script vuln for service and vulnerability detection.

SSL Labs: Checked SSL/TLS strength.

D. Authorization

Permission granted for testing https://christhujyothi.com

https://demotestfire.net is an open test environment