CHANDIGARH UNIVERSITY, GHARUAN

Department of Computer Science and Engineering

INTERNSHIP REPORT

Submitted in Partial Fulfilment of the
Requirements for the Degree of
Bachelor of Engineering (B.E.) in Computer Science and Engineering

Internship Program (20 Sep-25 Dec):

Cybersecurity Internship Program – Shentinelix Sphere Pvt Ltd

Task 8

OWASP Top 10(2021) and Secure SDLC

Internship Report Submitted By:
Vikash Upadhyay

Table of Contents

Executive Summary

This internship focused on practical application of web and network security fundamentals with emphasis on the OWASP Top Ten categories, image geolocation (IMINT/GEOINT), passive OSINT investigation workflows, and integrating secure engineering through a Secure Software Development Life Cycle (Secure SDLC). Through hands-on labs (TryHackMe rooms: OhSINT, SearchLight, Sakura, ACME, OWASP labs) and supervised exercises, I gained competence in identifying common web vulnerabilities (IDOR, Injection, Broken Authentication, SSRF, Crypto issues, insecure components, etc.), conducting structured reconnaissance, documenting findings, and proposing prioritized remediation grounded in secure-by-design principles. The report concludes with a prescriptive Secure SDLC tailored for small-to-medium teams and an actionable checklist for developers and security teams.

Introduction

Modern web applications power businesses yet also expose a broad attack surface. The OWASP Top Ten represents high-impact categories of weaknesses commonly found in web apps. This internship combined vulnerability theory with hands-on practice: recognizing flaws, understanding how they occur, and — most importantly — how to mitigate them in design, code, deployment, and operations.

Complementing technical vulnerability work, the internship covered OSINT and IMINT techniques (SearchLight, Sakura) used to gather contextual intelligence during investigations and penetration tests. Finally, I studied Secure SDLC practices to embed security early in development and to continuously verify security controls across CI/CD and production.

This report documents duties performed, technical accomplishments, the specific OWASP topics covered, extended technical skills gained, challenges faced, and a full Secure SDLC with practical controls and checklists.

Overview of the Company

Shentinelix Sphere Pvt. Ltd. is a cybersecurity training and solutions provider that specializes in delivering practical exposure in information security, penetration testing, and cyber defense. The company focuses on empowering students and professionals with real-world skills through structured labs, Capture-the-Flag (CTF) style challenges, and guided mentorship.

During the internship, I was assigned to the Cybersecurity Research and Training Department The organization strongly emphasizes hands-on learning by offering lab environments, curated challenges, and continuous support from experienced security practitioners, ensuring that interns develop both technical competence and a strong research methodology.

Description of Duties

During the internship I was responsible for:

1. OWASP coursework & practical challenges
   - Studied each OWASP Top Ten category in detail.
   - Completed hands-on labs that demonstrated identification and remediation strategies.
2. Reconnaissance & IMINT/OSINT
   - Performed passive and active reconnaissance using DNS tools, Shodan, Google dorking.
   - Geolocated images and extracted metadata (OhSINT / SearchLight).
   - Carried out Sakura OSINT chain-of-evidence tasks: username correlation, GitHub history recovery, and cryptocurrency transaction tracing.
3. Artifact analysis
   - Located and analyzed flat-file SQLite databases in labs, extracted password hashes and understood storage mistakes.
   - Practiced responsible handling of discovered sensitive artifacts (redaction in reports).
4. Reporting & recommendations
   - Produced technical findings with risk ratings, remediation steps, and suggested Secure SDLC improvements.
   - Documented reproducible steps (defensive verification) and created developer-friendly remediation instructions.

Accomplishments

- **OWASP Lab Completion:** Completed exercises covering each OWASP topic with defensive write-ups for each vulnerability found in the labs.

- **Image Geolocation Competency:** Successfully geolocated multiple images in SearchLight and OhSINT rooms by correlating landmarks and metadata.

- **OSINT Investigator Tasks (Sakura):** Traced attacker username across platforms, located crypto wallet artifacts, identified Ether wallet 0xa1023…and traced interactions to Ethermine/Tether (using blockchain explorer). Documented steps undertaken and corroborating evidence.

- **Artifact Analysis:** Located utech.db.sqlitein a lab scenario, enumerated users and found MD5 password hashes; produced remediation and migration plan to Argon2.

- **Secure SDLC Proposal:** Designed and proposed a 7-phase Secure SDLC tailored for small teams (detailed below) and created checklists for each lifecycle phase.

- **Report Deliverables:** Produced fully documented technical findings with reproducible defensive verification steps and remediation priorities.

1. **Broken Access Control**

   Users can access resources or functions beyond their permissions. Example: Changing a URL ID to view another user's data.
   Mitigation**:** Enforce server-side authorization checks, use least-privilege roles, opaque IDs.

2. **Cryptographic Failures**

   Sensitive data exposed due to weak or missing encryption. Example: Storing passwords in MD5 or plaintext.
   Mitigation**:** Use strong encryption, TLS 1.2+, secure password hashing, secrets management.

3. **Injection**

   User input is treated as code, allowing attackers to manipulate queries or commands. Example: SQL injection to steal DB data.
   Mitigation**:** Use parameterized queries, input validation, allowlists.

4. **Insecure Design**

   Flaws in application architecture that create vulnerabilities. Example: Disabling OTP validation in development and forgetting to re-enable.
   Mitigation**:** Threat modeling, secure design principles, security reviews during planning.

5. **Security Misconfiguration**

   Incorrect system or app settings that expose vulnerabilities. Example: Default admin accounts or verbose error messages.
   Mitigation**:** Harden configs, remove defaults, disable unnecessary features, perform regular audits.

6. **Vulnerable and Outdated Components**

   Using software with known vulnerabilities. Example: Running an outdated WordPress version vulnerable to RCE.
   Mitigation**:** Keep software up to date, monitor CVEs, use vulnerability scanners.

7. **Identification and Authentication Failures**

Weak authentication mechanisms that allow account takeover. Example: Brute-force attacks on weak passwords.
Mitigation**:** Strong passwords, MFA, rate limiting, secure session management.

8. **Software and Data Integrity Failures**

Data or code modified without integrity checks. Example: Downloading software that was tampered with.
Mitigation**:** Use hashes/digital signatures, verify integrity of downloads and updates.

9. **Security Logging & Monitoring Failures**

Lack of logging or monitoring prevents detection of attacks. Example: No logs for failed login attempts or admin access.
Mitigation**:** Implement logging, monitor critical events, secure log storage, alert on anomalies.

10. **Server-Side Request Forgery (SSRF)**

Attacker forces server to make requests to internal or external systems. Example: Using a web app to scan internal network.
Mitigation**:** Validate and whitelist URLs, restrict outbound requests, block internal network access where unnecessary.

Skills Learned

Below are the concrete technical skills and example commands/techniques I practiced. These are defensive, investigative and operational — safe for reporting and remediation.

Reconnaissance & IMINT/OSINT

- DNS & passive recon: whois, nslookup, dig usage; use dig @1.1.1.1 example.com TXT to view TXT records.
- Image intelligence: use exiftool image.jpg to read metadata (note: many images lack EXIF in modern web sharing).
- Shodan & DNSDumpster: discovery of exposed services and subdomain mapping.
- Google dorking: site:example.com filetype:pdf "confidential" to find exposed docs (ethical use only).

Artifact & DB handling

- SQLite inspection: sqlite3 utech.db.sqlite ".tables" then SELECT queries to safely read non-production test DBs.
- Hash handling: recognizing weak hashes (MD5) and recommending use of Argon2/bcrypt.

Application & Infrastructure testing (defensive)

- Service discovery: nmap -sV -p 1-65535 <target> for inventory (only in scope).
- Configuration review: check server responses for debug banners; verify TLS with tools like openssl s_client -connect host:443 -servername host.

- Container hygiene: use trivy or similar scanners (integrate into CI).

Reporting & remediation

- Produce prioritized remediation (High/Medium/Low), with risk, exploitation impact, and suggested changes.
- Prepare developer-ready snippets for secure code (parameterized queries, storage hardening).

Challenges Faced

- Ambiguity in Evidence: Some IMINT/OSINT artifacts lacked context; correlating multiple sources was necessary to avoid false positives.
- Balancing Depth vs Breadth: Numerous findings required triage; learned to prioritize by exposure and impact.
- Legacy Examples in Labs: Labs sometimes use intentionally vulnerable older software (purposeful) — differentiating these from production reality required careful communication in reports.
- Secure SDLC Adoption Friction: Presenting SDLC changes to development teams highlighted cultural and process challenges (needs leadership buy-in and automation incentives).

Conclusion

This internship provided a broad, practical foundation across web security (OWASP topics), OSINT/IMINT investigative techniques, and secure engineering through a Secure SDLC. I developed durable technical skills (defensive verification, secure coding guidance), investigative skills (IMINT, Sakura chain-of-evidence), and process understanding required to reduce security risk across an application lifecycle. The Secure SDLC I propose is lightweight enough to adopt in small teams and robust enough to prevent many common web vulnerabilities. Implementing these recommendations will materially increase the security posture of web applications and reduce future remediation overhead.

Acknowledgments

I would like to thank the team at **Shentinelix Sphere Pvt. Ltd**. for providing access to lab environments, learning resources, and a supportive environment that encouraged practical learning and skill development. Special thanks to the **TryHackMe** platform and the authors of the Research Methodology room for providing structured, hands-on lessons that significantly enhanced my research and problem-solving skills. Finally, I extend my gratitude to my peers and family for their continuous support and encouragement throughout the internship period.