COMP 490: SSL/TLS Certificate Vulnerability Testing Project Proposal

Aimen Arif and Simran Dhillon

1. Abstract/Summary

SSL/TLS (Secure Sockets Layer/Transport Layer Security) protocols are a major part of secure internet communication (Rescorla). These protocols basically keep sensitive data safe when it is sent between web browsers or servers and they make sure that millions of online transactions every day are private, accurate, and authenticated (Dierks and Rescorla 1). However, many businesses don't set up and maintain their SSL/TLS implementations correctly. This in turn makes their systems vulnerable to several attack types, such as man-in-the-middle attacks, protocol downgrade attacks, and cryptographic exploitation (Georgiev et al. 83).

SSL/TLS misconfigurations are still a common and serious issue despite the progress that has been made in network security. These weaknesses allow for exploitation through attacks such as man-in-the-middle, protocol downgrade, and cryptographic abuse (Georgiev et al. 83; Vratonjic et al. 541). Some of the most frequent problems include the use of outdated protocol versions such as SSLv3 or TLS 1.0, weak cipher suites that can be broken with brute-force attacks, certificate errors involving expiration, improper issuance, or hostname mismatches, insecure certificate chain validation, and the absence of modern protections like HTTP Strict Transport Security, Certificate Transparency, or Perfect Forward Secrecy (Holz et al. 247; Durumeric et al. 477). These issues place organizations at risk of data breaches, credential theft, and reputational damage, which is why correct configuration of SSL/TLS remains essential for a strong cybersecurity posture (National Institute of Standards and Technology 12).

This project will address these concerns by carrying out a thorough security evaluation of SSL/TLS implementations on publicly accessible websites. Using standard testing tools such as SSLyze and OpenSSL, and following established ethical hacking practices, the project will scan and analyze websites for protocol, cipher, and certificate flaws, document the technical details of each vulnerability, assess the risks, and provide remediation strategies based on industry standards (Sivakorn et al. 319; Buchanan and McMahon 5; Hoffman-Andrews 2).

Each identified vulnerability will be documented with technical details, risk assessment, and potential exploitation situations drawn out carefully. Our project will produce a comprehensive report that summarizes discovered vulnerabilities and provides actionable recommendations based on current industry standards from organizations such as NIST (National Institute of Standards and Technology), OWASP (Open Web Application Security Project), and the Internet Engineering Task Force (IETF) (Open Web Application Security Project 8). The recommendations will be prioritized based on severity and ease of implementation so that the most critical vulnerabilities can be effectively addressed first (Mell et al. 1).

The deliverables include thorough technical documentation detailing the testing methodologies employed, as well as vulnerability assessment reports for each domain examined. Additionally, practical recommendations focusing on current security best practices will be provided. A comparative analysis will also be included which identifies the common and recurring misconfiguration patterns observed (Kumar et al. 102). Altogether, this project seeks to improve the field of cybersecurity by highlighting prevalent SSL/TLS security challenges and providing insights to enhance security frameworks, thereby protecting sensitive information from potential threats (Felt et al. 291).

2. Aim of This Project

The aim of this project is to assess the current state of SSL/TLS security implementations across publicly accessible websites. We will identify critical vulnerabilities and misconfigurations and then provide evidence-based solutions that align with current industry security standards. By conducting ethical security testing on websites, we seek to contribute to improved cybersecurity awareness and practice within the academic community while developing practical skills in security assessment and vulnerability analysis.

3. Objectives

(*Reference*: *Bloom's Taxonomy*: https://www.bloomstaxonomy.net/)

Objective 1: Analyze and Evaluate SSL/TLS Protocol Implementations	• Examine target websites to identify which SSL/TLS protocol versions are supported • Evaluate the security stance by determining if vulnerable or faulty protocols are enabled on the websites analyzed • Compare findings against current recommendations for protocol version support • *Bloom's Taxonomy Level: Analyzing, Evaluating*
Objective 2: Assess Cipher Suite Strength and Configuration	• Identify all cipher suites supported by target servers, including their key exchange algorithms, encryption methods, and MAC functions • Evaluate the cryptographic strength of implemented cipher suites against current standards • *Bloom's Taxonomy Level: Analyzing, Evaluating*
Objective 3: Validate Certificate Management and PKI Implementation	• Examine digital certificates for validity period, ensuring they are neither expired nor deployed too early • Verify certificate chain completeness and proper intermediate CA certificate installation • Check for certificate transparency compliance and proper hostname validation • Identify self-signed certificates, usage, and certificate key strength • *Bloom's Taxonomy Level: Analyzing, Evaluating*
Objective 4: Investigate Modern Security Feature Implementation	• Test for HTTP Strict Transport Security (HSTS) header presence and proper configuration • Assess implementation of Perfect Forward Secrecy (PFS) through ephemeral key exchange support • Evaluate OCSP stapling implementation for certificate revocation checking • Check for proper implementation of TLS session renegotiation and compression settings • *Bloom's Taxonomy Level: Analyzing, Evaluating*

Objective 5: Synthesize Findings into Comprehensive Vulnerability Reports	• Create detailed technical documentation of all identified vulnerabilities with CVSS scores • Categorize vulnerabilities by severity (Critical, High, Medium, Low) and exploitation difficulty • *Bloom's Taxonomy Level: Analyzing, Synthesizing*
Objective 6: Create Evidence-Based Remediation Recommendations	• Develop specific solution steps for each identified vulnerability category • Align recommendations with industry frameworks • Provide configuration examples and implementation guidance for common web server platforms • Prioritize recommendations based on risk level and implementation complexity • *Bloom's Taxonomy Level: Creating, Evaluating*
Objective 7: Apply Ethical Security Testing Methodologies	• Demonstrate adherence to responsible disclosure principles and ethical hacking guidelines • Implement proper scope limitation and authorization procedures for security testing • Document the complete testing methodology for reproducibility and academic integrity • Establish communication protocols for reporting findings to affected organizations • *Bloom's Taxonomy Level: Applying, Evaluating*

4. Project Schedule

Week 1-2: Planning & Setup Phase	• Literature review on SSL/TLS vulnerabilities • Tool installation and configuration (SSLyze, OpenSSL, nmap) • Development of testing methodology and ethical guidelines • Target website selection and scope definition • Deliverable: Project Proposal document
Week 3-4: Initial Testing	• Passive information gathering on target domains • Initial SSL/TLS scanning with automated tools • Protocol version list • Basic cipher suite analysis
Week 5-6: Vulnerability Assessment	• Detailed cipher suite strength analysis • Certificate validation and chain verification • Testing for known vulnerabilities • Manual verification of automated findings
Week 7: Analysis & Documentation	• Vulnerability classification and severity assessment • False positive elimination and result validation • Comparative analysis across tested domains • Pattern identification in misconfigurations
Week 8: Research & Development	• Research current industry best practices • Develop specific remediation recommendations

	• Create configuration examples and implementation guides • Prioritize recommendations by risk and effort
Week 9: Report Compilation	• Technical report writing • Summary development • Review and quality assurance • Draft final report
Week 10: Finalization & Presentation	• Final report polishing • Presentation preparation • Deliverable: Final project report and presentation
Key Milestones	End of Week 2: Testing environment ready End of Week 4: 50% of scans completed End of Week 6: All testing completed End of Week 8: Solution/Report draft completed End of Week 10: Final submission

5. Project Scope

In Scope:

Target Systems	• Publicly accessible websites (for example: .edu domains) • Government portals with public access • Open educational resource platforms • Approximately 15-20 distinct domains to ensure depth of analysis
Technical Assessment Areas	• SSL/TLS protocol version support and configuration • Cipher suite strength and ordering • Digital certificate validity, chain, and configuration • Implementation of modern security features (HSTS, PFS, OCSP stapling) • Vulnerability to known SSL/TLS attacks • Certificate transparency compliance • TLS session management and renegotiation security
Deliverables	• Comprehensive project methodology document • Technical vulnerability assessment for each tested domain • Consolidated findings report with comparative analysis • Solutions recommendation guide with configuration examples • Final Project Report • Project presentation with visual data representations • Complete scan logs and tool outputs (can be part of appendix)
Tools and Technologies	• SSLyze for automated SSL/TLS scanning • OpenSSL for manual certificate and protocol testing • nmap with ssl-enum-ciphers script for cipher suite enumeration • SSL Labs' ssltest for baseline comparisons

	• Python for data processing and report generation

Out of Scope:

Excluded Activities	• Testing beyond SSL/TLS configuration assessment • Testing of internal networks or systems requiring authentication • Social engineering or phishing assessments • Source code review or application-layer vulnerability testing • Data exfiltration or access to protected resources
Technical Exclusions	• Wireless security assessments • VPN or other encrypted tunnel protocols beyond HTTPS • Email security (SMTP/TLS, STARTTLS) - separate protocol considerations • IoT device security testing • Mobile application security beyond web interfaces • Database encryption or at-rest encryption analysis
Key Limitations	• Legal and Ethical Constraints: Testing limited to publicly accessible systems where security research is legally permitted under Computer Fraud and Abuse Act (CFAA) considerations • Time Constraints: 10-week project timeline may limit depth of analysis for complex configurations • Tool Limitations: Automated tools may produce false positives requiring manual verification • Scope Creep Prevention: Strict adherence to SSL/TLS security only; no expansion into adjacent security domains • Resource Constraints: Testing performed from single network location; may not detect geo-specific configurations

6. Budget

This project is designed as an academic research project with minimal financial requirements, by using open-source tools and thus this project demonstrates that comprehensive security research can be conducted effectively using freely available tools and resources.

Total Project Budget: $0.00

Specific Cost Breakdown:

Software and Tools: $0.00	• SSLyze (Open Source) - $0.00 • OpenSSL (Open Source) - $0.00 • nmap (Open Source) - $0.00 • Python and libraries (Open Source) - $0.00 • SSL Labs API (Free tier) - $0.00

	- Documentation tools (Google Docs/Markdown editors) - $0.00
Hardware and Infrastructure: $0.00	- Personal laptop or university computer lab - $0.00 (existing resource)
Personnel: $0.00	- Student researcher time - $0.00 (academic project) - Help from Professor/Peers - $0.00 (part of course instruction)
Documentation and Reporting: $0.00	- Report writing software - $0.00 (Microsoft Word/LaTeX/Google Docs) - Presentation software - $0.00 (PowerPoint/Google Slides) - Data visualization tools - $0.00 (Python matplotlib, Excel)

7. Risk Management

Risk	**Description**	**Probability**	**Impact**	**Mitigation Strategies**
Risk 1: Legal and Ethical Concerns	Potential misinterpretation of security scanning as malicious activity or unauthorized access	Low	High	- Strictly limit testing to publicly accessible HTTPS endpoints - Maintain detailed logs of all testing activities with timestamps - Prepare documentation explaining the educational nature of the project - Include disclaimer in final report about ethical testing boundaries
Risk 2: Technical Tool Failures or Inaccuracies	Automated scanning tools may produce false positives, false negatives, or encounter unexpected errors	Medium	Medium	- Use multiple tools (SSLyze, OpenSSL etc.) for cross-validation - Manually verify all critical findings before including in final report - Maintain updated versions of all testing tools to incorporate latest vulnerability checks - Document tool versions and configurations for reproducibility - Test scanning methodology on known-vulnerable test servers first - If significant tool issues arise, use alternative tools, adjust to accommodate additional verification time
Risk 3: Target Systems Blocking or Rate-Limiting Scans	Web application firewalls (WAFs) or intrusion detection systems	Medium	Medium	- Implement conservative scanning parameters with delays between requests - Use respectful user-agent strings identifying the scanner

	may block repeated requests			- Distribute scanning across multiple days to reduce detection likelihood - Monitor for HTTP 429 (Too Many Requests) or 403 (Forbidden) responses - Maintain backup list of alternative target domains if primary targets become inaccessible - Consider testing during off-peak hours to reduce service impact
Risk 4: Insufficient Vulnerabilities Found	Target sites may be well-configured, resulting in limited findings for analysis	Low-Medium	Medium	- Select diverse sites including smaller institutions more likely to have misconfigurations - Expand analysis to include comparative best practices even for well-configured sites - Include analysis of security feature adoption rates across the tested sites - Document positive findings (good security practices) as equally valuable - Focus on detailed technical analysis and solutions rather than just vulnerability analysis

8. Evaluation and Success Criteria

Progress Monitoring Mechanisms:

Weekly Progress Reviews and Documentation	**Milestone-Based Assessment**	**Quantitative Metrics Tracking**
- Self-assessment checklist documented (using Planner on teams) - Tasks completed vs. planned noted - Number of domains successfully scanned - Vulnerabilities identified and categorized - Note down any challenges encountered	- Major evaluations done at end of weeks to see if the defined milestones were met or not and what the next steps should be - Deliverable submission with rubric-based evaluation (following deadlines given in class) - Keep track of any key decision points made between group members to ensure	- Number of domains successfully scanned noted down - Total vulnerabilities identified by category and severity - Tool execution success rates noted down - Time spent per project phase vs. estimated time by weekly plan noted down - Documentation completion discussed weekly by

- Upcoming week's priorities noted	project stays on track and everyone on same page	communicating with group members and next steps discussed

Success Criteria:

	Measurable Goal	Success Threshold	Success Indicator	Evaluation Method
Criterion 1: Comprehensive Technical Coverage	Successfully scan and analyze minimum 10 domains (target: 15-20)	Complete SSL/TLS assessment across all key areas: protocol versions, cipher suites, certificates, security features	Cover all the planned technical assessment areas and cover all sites outlined	Review of scan logs, technical report completeness checklist, tool output verification
Criterion 2: Vulnerability Identification and Classification	Identify and document about 10-15 distinct vulnerability instances across tested domains	Properly classify all findings with severity ratings (Critical, High, Medium, or Low) and add scores where applicable	Identify at least one critical or high-severity vulnerability with supporting evidence and exploitation context	Review of vulnerability documentation quality, accuracy of severity assessments, manual verification of key findings
Criterion 3: Quality of Solutions Recommendations	Provide specific, solutions or steps for each identified vulnerability category	Recommendations align with current industry standards (ex. NIST, OWASP, IETF) with cited references	Include configuration examples for at least 3 common web server platforms (ex. Apache, Nginx, IIS) with implementation difficulty ratings	Review of solutions guide, assessment of actions protocol, alignment check against industry frameworks already set up
Criterion 4: Documentation Quality and Completeness	Deliver all specified project deliverables with professional quality	Technical report should be completed, all scan outputs and raw data properly organized and included in report or zip file to hand in	The final report should be clear, well-organized, formatted well, with effective use of charts/graphs, and no grammatical errors	Rubric-based assessment should be provided in class so that we can check of what we have done correctly
Criterion 5: Technical Skill Demonstration	Effectively utilize at least 3 different	Proper configuration and execution of SSLyze, OpenSSL, and one additional tool	Custom scrip or automation to enhance testing efficiency and add	Review of tool configurations, command-line examples, any

	SSL/TLS testing tools	(explore different tools and pick best ones)	in a correlation of results across multiple tools	custom scripts developed

References

Buchanan, William J., and Adrian McMahon. "Ethical Hacking and Penetration Testing: Establishing Professional Standards." *Computer Fraud & Security*, vol. 2016, no. 9, 2016, pp. 5-10.

Dierks, Tim, and Eric Rescorla. "The Transport Layer Security (TLS) Protocol Version 1.2." *RFC 5246*, Internet Engineering Task Force, Aug. 2008, pp. 1-104.

Durumeric, Zakir, et al. "The Matter of Heartbleed." *Proceedings of the 2014 Conference on Internet Measurement Conference*, ACM, 2014, pp. 475-488.

Felt, Adrienne Porter, et al. "Measuring HTTPS Adoption on the Web." *Proceedings of the 26th USENIX Security Symposium*, USENIX Association, 2017, pp. 1323-1338.

Georgiev, Martin, et al. "The Most Dangerous Code in the World: Validating SSL Certificates in Non-Browser Software." *Proceedings of the 2012 ACM Conference on Computer and Communications Security*, ACM, 2012, pp. 38-49.

Hoffman-Andrews, Jacob. "Certificate Transparency: Public, Verifiable, Append-Only Logs." *Communications of the ACM*, vol. 64, no. 6, 2021, pp. 76-83.

Holz, Ralph, et al. "The SSL Landscape: A Thorough Analysis of the X.509 PKI Using Active and Passive Measurements." *Proceedings of the 2011 ACM SIGCOMM Internet Measurement Conference*, ACM, 2011, pp. 427-444.

Kumar, Sandeep, et al. "A Comprehensive Survey on SSL/TLS Vulnerabilities and Their Mitigation Strategies." *Computer Networks*, vol. 181, 2020, pp. 107-124.

Mell, Peter, et al. "Common Vulnerability Scoring System (CVSS) Version 3.1: Specification Document." *NIST Interagency Report 8246*, National Institute of Standards and Technology, 2019, pp. 1-23.

National Institute of Standards and Technology. "Guidelines for the Selection, Configuration, and Use of Transport Layer Security (TLS) Implementations." *NIST Special Publication 800-52 Revision 2*, U.S. Department of Commerce, 2019, pp. 1-38.

Open Web Application Security Project. "OWASP Top Ten 2021: A Standard Awareness Document for Developers and Web Application Security." *OWASP Foundation*, 2021, pp. 1-25.

Rescorla, Eric. "The Transport Layer Security (TLS) Protocol Version 1.3." *RFC 8446*, Internet Engineering Task Force, Aug. 2018, pp. 1-160.

Sivakorn, Suphannee, et al. "The Cracked Cookie Jar: HTTP Cookie Hijacking and the Exposure of Private Information." *Proceedings of the 2016 IEEE Symposium on Security and Privacy*, IEEE, 2016, pp. 724-742.

Vratonjic, Nevena, et al. "The Inconvenient Truth about Web Certificates." *Proceedings of the 2013 Workshop on Economics of Information Security*, 2013, pp. 541-567.