Section: PUBP

Web Application Penetration Testing Course For Developers

Zach Steele

Problem Statement:

There is a large difference in priorities and skillset between penetration testers and code developers, leading to a troubling amount of insecure web applications and difficulty communicating vulnerabilities to the application teams. While developers are constantly told to follow secure coding practices, oftentimes even utilizing security focused development frameworks, they rarely prioritize secure coding practices. (Secure Code Warrior, 2022) Most developers do not display a passion for cyber security. In one study, 68% of participants stated that they only did secure code training because of a compliance need or recent exploit. (Security Journey, 2024)

Personally, I believe a large factor contributing to this problem is the gap in cyber security training programs. These courses and programs fall into two categories based on my experiences as a penetration tester responsible for teaching application teams how to test and remediate vulnerabilities. First, developer focused resources are too high level to be helpful. These courses may teach the very basics of specific misconfigurations or exploits, but they do not provide many opportunities for developers to walk away with knowledge that can be applied to their current apps. Second, resources targeted towards penetration testers, meaning the material is extremely in depth, requires specialized tools, and a massive time investment from the developer. I believe that this pushes away most developers that were willing to commit a small amount of time to upskilling their cyber security knowledge. Ultimately, I believe that this is the reason we continue to see an increase in vulnerabilities every year, even though most developers receive annual penetration tests on their applications and secure coding training. (Tribbey & Winterfeld, 2023)

Solution Statement:

This project will help developers test their own applications for vulnerabilities in order to reduce the likelihood of exploitation from an attacker or annual penetration test by utilizing a training course dedicated to quickly learning how to test applications for common vulnerabilities.

Specifically, I plan on teaching developers how to test their applications from a similar set of material that penetration testers use to conduct assessments. Most web application penetration testers utilize a checklist with a set of vulnerabilities to test, payload lists, and an intercepting proxy tool to bypass client side controls. My goal is to use a similar framework that excludes complex attack types and unnecessary overhead. My target audience is developers who are looking for a way to reproduce findings or test new code for vulnerabilities before pushing to production. However, I will be providing third party links and suggestions on how to learn more as we go through the checklist for the audience members that would like to take penetration testing to the next level.

Lastly, I would like to explain my methodology for evaluating the effectiveness of this solution. It is easy to express the need for this course when reviewing the widespread security flaws across the internet. As an example, only 2% of the top 1000 visited sites have a properly implemented Content-Security-Policy (CSP). (Pacheco, 2020) In a few minutes, we could teach developers how to build a strong CSP to prevent cross-site scripting and framing attacks. However, measuring the impact and results of this course would

require a set of developers who plan to complete the whole course and a much larger time window to analyze actions taken based on the material. Since we do not have enough time in this course to build the course and test its effectiveness, I will ask developers with varying ranges of security knowledge to review the material and provide feedback. Hopefully, this will result in a clear picture of the potential impact this course will have on a true audience.

Completed Tasks (Last 2 Week):

- First, I installed a Kali virtual machine, all required developer tools and multiple testing applications. This included Postman, Firefox, Chrome, Chromium, Burp Suite, Juice Shop, and bWAPP. Then, I practiced using the tools and changed some configurations to make the content more clear for the audience.
- Next, I created the outline for the video series. While most videos have a description and key notes, other videos have not been completely noted yet.
- After, the GitHub repo and payloads page outline were created. Please visit the repo to view all work papers: https://github.gatech.edu/zsteele3/ZS6727Summer2025
- Completed the checklist template and first set of items. This includes a list of other checks, but the description and resources have not been completed after the first section.
- Created a YouTube channel and tested recording apps, then uploaded a placeholder video to practice adding a description, creating a playlist, and using timestamps. https://www.youtube.com/watch?v=0eWlFAxKzFU

Tasks for the Next Project Report:

Over the next two weeks, I will continue to build out the payloads in GitHub and expand on the checklist. Additionally, I will start filming youtube videos for the course. My goal is to complete most of the videos leading up to the first payload section(XSS). Some videos, like the introduction, will be skipped so that I can film them at a later time when the course is completely built out.

Questions I have or Issues I'm running into:

During the first week, I struggled to get my environment setup for testing due to multiple server issues. This issue was remediated by switching to a Mac computer, causing some limitations in tooling. Otherwise, I am not running into any issues with my project.

However, I would like clarification on my evaluation plan. Would it be appropriate to survey my coworkers and colleagues for opinions and acceptance of my project? I would like to ask them what they think of the course and how effective they think it could be for achieving my overall goal.

Methodology Paragraph Summary:

During the first few weeks, I will be building the structure of all required documents and creating the video order to ensure I have a pathway to follow with a clear understanding of the final outcome. Afterwards, I will spend a few hours preparing and creating the documentation required for each video in order. For example, when it is time to film the SQL injection video, I will first build out the payload list, checklist, and find the exact labs and tools that will best demonstrate how to test for SQL injection. This methodology will provide a clear path forward after every action item and will make it easy to see the work completed each week along with what is coming in the following weeks. It is important to note that some videos and deliverables may need to be completed out of order due to required resources or time to

complete. For example, the introductory video cannot be completed before building out the course materials because that video will display the completed checklist and GitHub to give the viewer an idea of what we will be doing for the remainder of the course.

Timeline:

Week #	Description of Task	Status
W1	Installed VM, required tools and test apps.	Completed
W1	Created checklist template and first section.	Completed
W1/W2	Created GitHub and payloads page outline. https://github.gatech.edu/zsteele3/ZS6727Summer2025	In progress
W2	Created the outline for the video series.	In Progress
W2	Created YouTube channel and uploaded placeholder video.	Completed
W3/W4	Build out the response headers and cookies checklist and GitHub pages.	Not Started
W3/W4	Record video for continuous review, response headers, and cookie testing.	Not Started
W3/W4	Conduct research on secure coding practices for Video 3, resulting in a summary of actionable items for the dev team.	Not Started
W4	Film Video 3 - Best coding practices/remediation.	Not Started
W5/W6	Complete GitHub pages and checklist for all payload based exploits covered in the checklist.	Not Started
W5/W6	Film videos 7(XSS) through 10(OS command Inj.)	Not Started
W5/W6	Reach out to developers for evaluation of material up to this point.	Not Started
W5/W6	Re-evaluate checklist items based on studies of most common vulnerabilities.	Not Started
W7/W8	Complete videos 11(login pages) to Video 16(Outro).	Not Started
W7/W8	Make changes on past material based on developer feedback.	Not Started
W7/W8	Film optional videos if time permits	Not Started
W9	Review entire project for places that may need improvement to make the series more cohesive.	Not Started

Evaluation:

I plan to evaluate the success and effectiveness of this course by surveying multiple developers with varying levels of development and cyber security knowledge.

Report Outline:

Report outline has not been created yet.

References:

Specific penetration testing payloads and remediation guidance will be linked throughout the GitHub repo and checklist resources section. The references below are all utilized in this document.

Secure code warrior survey finds 86% of developers do not view application security as a top priority. Secure Code Warrior. (Apr 05, 2022). https://www.securecodewarrior.com/press-releases/secure-code-warrior-survey-finds-86-of-developers-do-not-view-application-security-as-a-top-priority

A Study on Secure Coding Training. Security Journey. (2024, January). https://www.securityjourney.com/hubfs/SJ_StudyonSecureCodingTraining24_011624.pdf

Tribbey, B., & Winterfeld, S. (2023, April 18). *Slipping through the security gaps: The rise of application and API attacks* | Akamai. Akamai. https://www.akamai.com/blog/security-research/the-rise-of-application-and-api-attacks

Pacheco, P. (2020, September 3). *Content security policy limits dangerous activity... so why isn't everyone doing it?* Bitsight. https://www.bitsight.com/blog/content-security-policy-limits-dangerous-activity-so-why-isnt-everyone-doing-it

Appendix

My youtube channel link: https://github.gatech.edu/zsteele3/ZS6727Summer2025

My GitHub page link: https://www.youtube.com/watch?v=0eWlFAxKzFU