Website Security Research Project

CS 467 - Summer 2021

Team Members:

Cody Medaris

Ray Franklin

Emanuel Ramierz

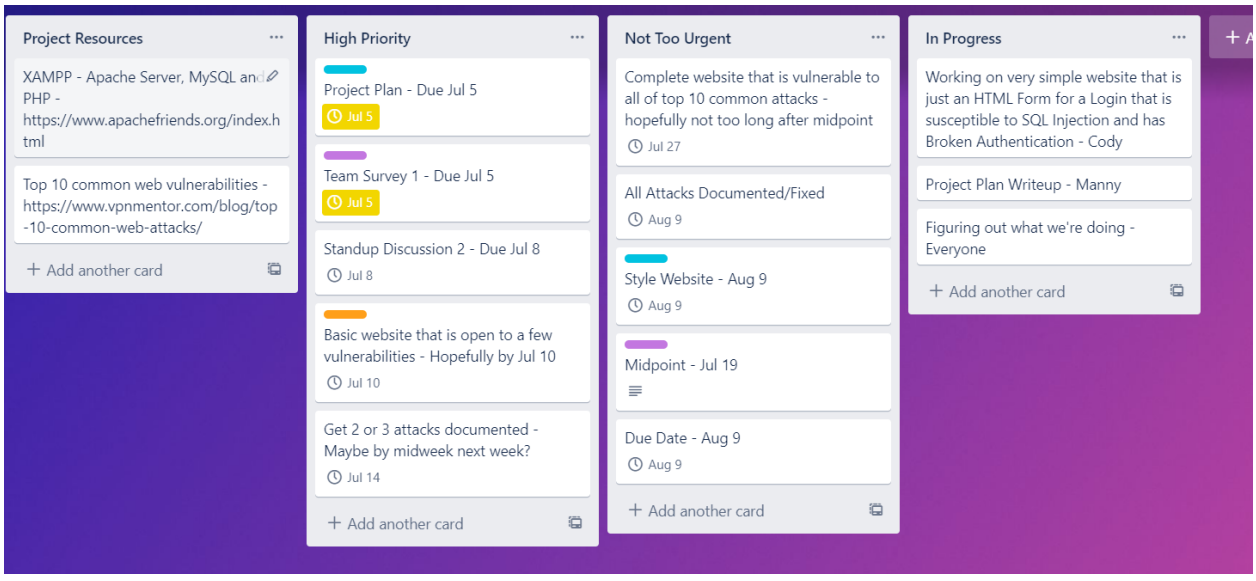
1. Introduction:

The Website Security Research Project is meant to be a sort of playground for users to try out different cyber attacks against a vulnerable website, very similar to the Damn Vulnerable Web Application (https://dvwa.co.uk) and HackMe (https://hack.me). Also the project will show different approaches on how to protect a website from some of the most common forms of attack.

From a design perspective, the project will have a hosted vulnerable website on which all the attacks will be tested against by the user. The website will have a 'Vulnerability' toggle that when activated the website will be vulnerable to all attacks. When deactivated; none of the attacks will work.

After each successful attack and patch of said attack a formal writeup of the attack will be made by the team member in charge of the attack. For now, each member has been assigned three (3) attacks/patches to work on. Those writeups will be hosted on the project GitHub repository inside the writeups/ folder with the naming scheme 'attack_name_writeup.pdf'.

Each team member plans to spend roughly 80-90 hours total on their respective part of the project, perhaps more or less depending on outside factors (work, personal life, etc). Here is our current Trello board for the project:



2. Description of what your program does from the user's perspective.

The website main functionality will be a ‘Vulnerability’ toggle that when activated will render the website completely vulnerable to the user attacks and when deactivated will be invulnerable to user attacks. There will be a simple list of attacks that the user can select to perform the attacks and if the ‘Vulnerability’ toggle was activated the attack will go through; if not nothing will happen.

3. A description of your initial thoughts on the structure of the software.

The structure of this project will be a Github repository that will contain the web application and all the detailed formal write ups describing each of the top ten attacks listed below and plausible solutions to each attack from the perspective of the host.

Ideally, the writeups/ folder will contain write ups for each of the following, but not limited to, attacks:

1. Injection
2. Broken Authentication
3. Sensitive Data Exposure
4. XML External Entities (XXE)
5. Broken Access Control
6. Security Misconfiguration
7. Cross-Site Scripting (XSS)
8. Insecure Deserialization
9. Using Components with Known Vulnerabilities
10. Insufficient Logging & Monitoring

Writeup Structure (sections):

- Attack description
    - Attacker perspective:
        - Vulnerability
        - Implementation (with details)
    - Web application perspective
        - Detection
        - Solution
        - Further prevention (i.e Defense)
    - References for sources used

4. An initial listing of which software libraries, languages, APIs, development tools, servers, and other systems will be required to create and use the software.

    i. Python and/or PHP (for scripting attacks)
    ii. Javascript, HTML, CSS (for setting up the website)
    iii. MariaDB/SQL for database handling
    iv. Apache Server
    v. Virtual Machine for testing vulnerabilities

5. A thorough and complete description of what each Team member will be accomplishing for class assignments, including who will produce each type of documentation needed (see the Project Archive - Midpoint and Project Archive - Final assignments for more). Note: We encourage each team member to contribute to every assignment!

Week	Task	Format	Due Date	Assigned
3	Project Plan	PDF submission	July 5	All
3	Team Survey #1	Canvas Form	July 5	All
3	Standup Discussion #2	Canvas discussion	July 8	All
4	Team Survey #2	Canvas Form	July 12	All
4	Standup Discussion #3	Canvas Discussion	July 15	All
5	Team Survey #3	Canvas Form	July 19	All
5	Project Archive - Midpoint	Zip file: - Source code - 1 page Installation document (pdf) - 1-2 page Instructions document (pdf)	July 19	All
5	Demonstrate Project - Midpoint	Canvas Video submission (individual)	July 22	Individual
6	Elevator Pitch - Extra Credit	TBD	July 26	
6	Standup Discussion #4	Canvas Discussion	July 29	All
7	Create Poster	PDF submission	Aug 2	All
7	Team Survey #4	Canvas Form	Aug 2	All
7	Standup Discussion	Canvas Discussion	Aug 5	All
8	Team Survey #5	Canvas Form	Aug 9	All
8	Project Archive - Final	Zip file: - Source code - 1 page installation document (pdf) - 1-2 page instruction document (pdf) - 2-3 page Final Report (pdf)	Aug 9	All
8	Demonstrate Project - Final	Canvas Video submission (individual)	Aug 9	Individual

6. A list that describes how much time each Team member will be spending on their portion of the project. The amount spent per Team member must be roughly equivalent, and should be at least 80 hours each. Expand the project accordingly to make the work amounts equivalent, and/or make sure the Project is sufficiently complex. You may use a table here.

    a. Franklin, Ray

Task	Estimated Time (hours)

Week 3:
- Help finish project plan
- Finish creating website
- Help setup database
- Start prototyping and researching vulnerabilities and possible attacks for:
  - Cross-Site Scripting (XSS)
  - Insecure Deserialization
  - Using Components with Known Vulnerabilities
  - Insufficient logging & monitoring
	14

Week 4: - Perform Cross-Site Scripting (XSS) attack - Deploy Cross-Site Scripting (XSS) defense patch - Write report for Cross-Site Scripting (XSS) attack and defense	14
Week 5: - Perform Insecure Deserialization attack - Deploy Insecure Deserialization defense patch	14

- Write report for Insecure Deserialization attack and defense	
Week 6: - Perform Using Components with Known Vulnerabilities attack - Deploy Using Components with Known Vulnerabilities defense patch - Write report for Using Components with Known Vulnerabilities attack and defense	14
Week 7: - Help group with Insufficient Monitoring & logging attack, defense and write up	14
Week 8: - Polish attack writeups for final submission. - Submit	10
Total time	**80**

b. Medaris, Cody

Task	Estimated Time (hours)
Week 3: - Create project plan - Finish creating website - Help setup database - Start prototyping and	14

researching vulnerabilities and possible attacks for: - XML External Entities (XXE) - Broken Access Control - Security Misconfiguration - Insufficient logging & monitoring	
Week 4: - Perform XXE attack - Deploy XXE defense patch - Write report for XXE attack and defense	14
Week 5: - Perform Broken Access Control attack - Deploy Broken Access Control defense patch - Write report for Broken Access Control attack and defense	14
Week 6: - Perform Security Misconfiguration attack - Deploy Security Misconfiguration defense patch - Write report for Security Misconfiguration attack and defense	14
Week 7: - Help group with Insufficient Monitoring & logging attack, defense and write up	14
Week 8: - Polish attack writeups for final submission. - Submit	10

Total time	**80**

c. Ramirez, Emanuel

Task	Estimated Time (hours)

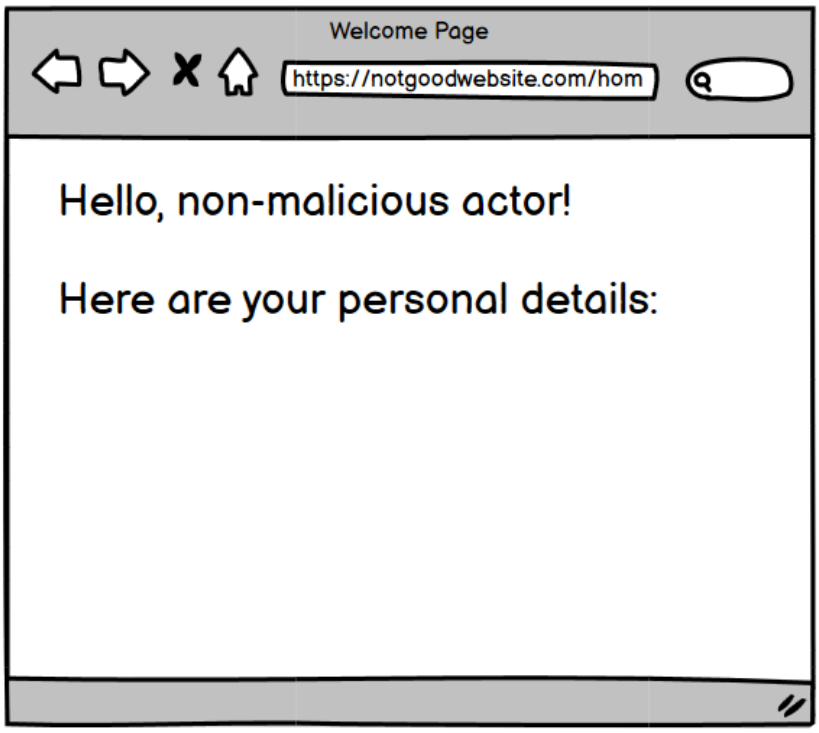
Week 3:
- Create project plan
- Finish creating website
- Help setup database
- Start prototyping and researching vulnerabilities and possible attacks for:
  - Injection
  - Broken Authentication
  - Sensitive Data exposure
  - Insufficient logging & monitoring
	14

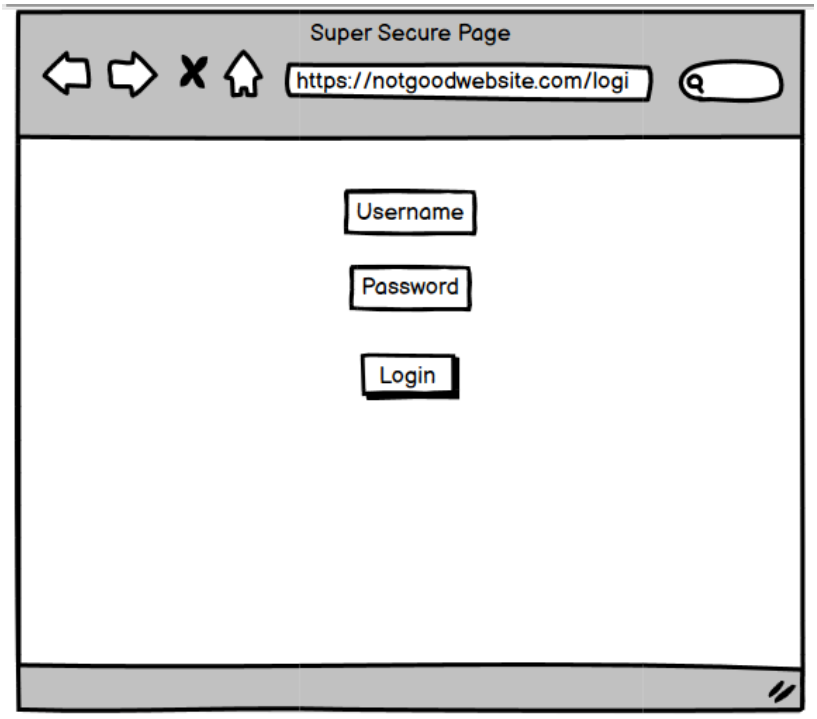
Week 4: - Perform Injection attack - Deploy injection defense patch - Write report for Injection attack and defense	14
Week 5: - Perform Broken Authentication attack - Deploy Broken Authentication defense patch - Write report for Broken Authentication attack and defense	14

Week 6: - Perform Sensitive Data Exposure attack - Deploy Sensitive Data exposure defense patch - Write report for Sensitive Data exposure attack and defense	14
Week 7: - Help group with Insufficient Monitoring & logging attack, defense and write up	14
Week 8: - Polish attack writeups for final submission. - Submit	10
Total time	**80**

7. At least two graphical examples that demonstrate a page, chart, board, level, flowchart, or layout.

A mock of what the login/homepage could be like for the website. Obviously the login page would have some fatal flaw allowing an attacker to log in with someone else's credentials:





8. A conclusion.

   The Website Security Research Project team plans to create an environment for users to learn about and experience the top ten most

critical security risks to web applications. This project will take an estimated 240 hours to complete and should be completed on schedule.

9. References (if needed).
    a. OWASP Top Ten Web Application Security Risks | OWASP - https://owasp.org/www-project-top-ten/
    b. Top 10 common web vulnerabilities - https://www.vpnmentor.com/blog/top-10-common-web-attacks/
    c. XAMPP - Apache Server, MySQL and PHP - https://www.apachefriends.org/index.html
    d. DVWA - Damn Vulnerable Web Application - https://dvwa.co.uk/