CIS 4930 - Software Security - Spring 2020

Mini-Project 1 --- Salt/Pepper Hash and Web Exploit Prevention

(required to work in teams of up to 4 for this assignment - register teams using the Canvas assignment)

Objective

Build a simple web interface to gain experience hashing passwords and preventing common web security issues. Create registration and login pages for a web server hosted on cloud infrastructure. Create versions of a 'dashboard' page representing primary functionality (of your choosing) with one version vulnerable to 4 classes of attack: 1. SQL Injection, IDOR/URL manipulation, CSRF/Session attacks, and XSS and one version where preventative measures have been incorporated. Record a screencast showing login functionality and user password storage, 4 classes of attacks and mitigations preventing attacks in hardened version.

Assignment

Demonstrate your knowledge of the covered attacks by implementing a vulnerable web service that you can then exploit. The exploits should violate stated security policy for your web site. You can choose any domain and use case for your sample build. The goal is not to build an extensive site, rather to build one or more simple user facing pages that include several input fields used to perform a typical task / use case (e.g., bank customer that logs into bank site to transfer balance or pay a bill). The site should be vulnerable to the 4 classes of attacks listed above. Then, your team must build another version of the site (could be a separate page or hosted separately) that duplicates the functionality without being susceptible to the exploits. The team will create a screencast with audio walkthrough of each exploit and the the result / impact to the user / organization. You will then discuss mitigations for each type of attack while attempting the exploit on the hardened version. The attacks should represent exploits that would inflict harm to the user and / or organization.

Task 1: Create registration and login page workflow that includes password hash, salt, and pepper and persist user info to cloud database. Registration and login workflow should be done manually without the aid of 3rd party services (e.g., Firebase) with the exception of standard libraries to generate the hash etc (e.g., bcrypt)**.**

Task 2: Create vulnerable "dashboard" page(s) that enables primary functionality for your chosen use case(s) - dashboard should only be accessible via a registered user login. Should be vulnerable to 4 types of exploits.

Task 3: Create hardened site with registration, login, and dashboard page(s) that mitigates each type of attack.

Task 4: Create Use / Abuse Case diagram showing functionality and potential threats. Provide a description of the usage scenarios, potential exploits, risks, and potential for security policy violations with the abuse cases. Your objective is to focus on security policy that targets to the use/abuse cases specified and not the overall domain of the application if the context is not needed to understand the impact of the exploit.

Task 5: Demonstrate login attacks and mitigations by recording a screencast. Walkthrough each attack, resources compromised and potential to harm to the user / organization. Demonstrate the hardened site mitigations put in place and attempt to exploit the

Task 6: Report (max 1.5 pages): Discuss how to initiate exploits, type of vulnerabilities, how the mitigations were applied, difference between password hashing & encrypting password and what salting and salt + peppering passwords enable (compare / contrast).

Required: Work in groups (i.e., minimum 2) up to 4. Use Google Cloud Platform (credits provided) to setup a server, host the web service (should be publicly accessible via IP) & open source SQL database. Not permitted to use Microsoft products or services on GCP.

SUBMIT : PDF report, source and config files, link to web services, screencasts (links ok) and any other data / files should be added to Git Repo. Provide public IP link to running versions of site for grading - Must remain active through March 29th.

Eval: feasibility, depth / realism, report analysis and discussion, screencast detail

Due Sunday, March 15 at 11:59pm - 200 Quiz & Mini-Project Points