Comp 4820 Project ~ Ideas and Considerations
Oct. 30, 2023
Written By: TM

MY IDEAS

After doing some additional research, I have found that our web application must have at least some dynamically generated content, because that is the most common point of entry for an XSS attack. I don't think that it is much more difficult than making a static html page. Think of it as creating a html template with empty spaces and using values pulled from the database to fill them.

I think that we should make at least two attacks. One Reflected and the other Persistent. This means that our web application will need to have at least two vulnerabilities. One vulnerability, for the reflected attack, will involve taking unvalidated user input and using it to fill part of a dynamically generated HTML page.

The other vulnerability, for the persistent attack will involve taking unvalidated user input and storing it in the database, which will be pulled from the database and used to fill dynamic content at a later date.

I am very much an amateur hacker and I do not want to be left to make the attacks by myself. The attack and the web app are very much intertwined, so I would like to contribute to the web application as well and feel that no one person should be left to implement it on their own.

I have started some preliminary design on the website, making it a FB clone catered to gardeners. I haven't started coding and am still in the design and analysis phase. I would like to get some input from the more webdev-savvy group members about what technologies that we should use. I am open to using React and ExpressJS for content generation and back-end management, but I do not know how to make a React app.

WEB APP DESIGN

1. Deciding on a tech stack
    a. Use a framework - MERN, etc
        i. Database system - Relational vs Non-relational (SQL vs Mongo)
            1. Design simple DB schema
            2. Identify points of access for malicious code

        ii. Backend framework - PHP vs. Python (Django, Flask) vs. JS native (Express-React)

1. The choice doesn’t really matter here. There is a bare minimum functionality that our application will need to fulfill.
   There are lots of libraries and frameworks to help us achieve this. I am not a web development expert, but I have some understanding.

b. Use vanilla html/css/js

i. Undesirable in my opinion because we would need to write more content generating and back-end managing/communicating code that already exists in the frameworks

2. One potential direction: social network for hikers/outdoor enthusiasts.

a. Multiple web pages

i. Landing Page → Login/Signup

ii. User’s Home page / news feed → Dynamically generated feed

iii. User’s Profile → Dynamically generated friends list, history, stats, etc

iv. Groups & Events → Dynamically generated group/event page

XSS ATTACK DESIGN

1. Determining types of attacks and how they will be performed.

b. Inject data into the web application through malicious web requests

c. Data is included in dynamically generated content that is sent to a web user without being validated

3. Reflected vs Persistent XSS

a. The injected script is reflected off of the web server to the target. Social engineering techniques may be used to trick a target into interacting with malicious content. The browser is tricked into executing the malicious script by hijacking the trust of the vulnerable web application

b. The injected script is stored on target servers such as in the database or in back-end code. The victim will retrieve the malicious content from the server when it requests the stored information that the malicious code is attached to.

4. Identify and analyze some common exploitable design flaws

a. Find places where a malicious script can be injected into dynamically generated HTML output.

b. Find places where a malicious script can be injected into DB to store persistently, and can be fetched and put into dynamically generated content

c. Use these points of entry to guide design and development of insecure web application.