

CS 526 Lab 2

Web Application Exploits

Ziqing Mao

Overview

- Exploit web application vulnerabilities
 - Cross site scripting
 - Cross site request forgery
- A vulnerable forum
 - <http://forest.cs.purdue.edu/phpBB2>
- You job
 - Exploit the vulnerabilities by posting malicious messages
 - When those messages are clicked by a victim, something bad happens
 - Messages are posted on the victim's behalf
 - Victim's profile is changed

Setup

- Register two accounts
 - Naming rules: lastname_1, lastname_2
 - Be patient when registering
- Send an email to TA once registered
 - A private forum is created for you
 - Only your two accounts can access the forum
- Why you need two accounts?
 - One account is enough
 - Two accounts may ease the testing and debugging
 - Use one account as the victim, another as the adversary

Submission

- All your submissions are messages posted in your private forum
 - It does not matter which account is used to post the submissions
- **Strictly follow the naming rules for your submission**
 - **final_attack_k**, where k is the task no
- Do NOT make the attack specific to your own victim account
 - The final submission should expect an arbitrary victim user, unless explicitly specified in the task description
- Send a detailed report to TA by email

Task 1

- You need to post a malicious message
- When the message is clicked, it popup an alert windows
 - The alert window should include the text “Hello Task 1”
- JavaScript function: `alert()`

Task 2

- You need to post a malicious message
- When the message is clicked, it prints out the cookies set by the web application for the victim user
- You can use an alert window to display the cookies
- To access the cookies in JavaScript: `document.cookie`

Task 3

- You need to post a malicious message
- When the message is clicked, it posts a new message on the victim user's behalf
- The new message is titled "Hello Task 3"
- Send a HTTP request to post the message
 - Use Fiddler to get a sample HTTP request for posting a message
 - Use XMLHttpRequest (AJAX) for to send the request

Task 4

- Users may disable JavaScript in their browsers
- Repeat task 3 without using JavaScript
- Use a tag with the “src” attribute

Task 5

- You need to post a malicious message
- When the message is clicked, it modifies the victim user's profile
- The victim user's signature is changed to "Hello Task 5"
- You target a specific victim user in this task
 - The victim user is named "victim_user"
- You may need to the user_id and the email of the victim user
 - Click the "Memberlist" link in the forum
- Use XMLHttpRequest (AJAX) to send a POST request

Task 6

- Repeat task 5 without using JavaScript
- Change the victim's signature to "Hello Task 6"
- You also target a specific victim user named "victim_user"
- Use a hidden form
 - You can assume the victim user is willing to click a button in the malicious message

Task 7

- Repeat task 5 for any victim user
- Change the victim's signature to "Hello Task 7"
- Use JavaScript to get the user's name, id, and email
 - Use AJAX programming to retrieve the standard page for changing profile
 - Parse the page to get the necessary information

Task 8

- You need to post a malicious message (P1) that contains a XSS worm
- When P1 is clicked, a new message P2 is posted
- P2 is another instance of the XSS worm
 - P2 contains the same content as P1
- When P2 is clicked, another new message P3 is posted
- When P3 is clicked, P4 is posted
- ...
- It propagates as the users click on the messages

Hints

- Understand basic concepts
 - How a web application works in general
 - CSS and CSRF vulnerabilities
 - Basic HTTP protocol
 - Basic JavaScript (AJAX programming)
- Useful tools
 - Fiddler
 - Firebug
 - LiveHttpHeaders

More Hints

- Free to use Google and any reference you can find
- The attack code are short
- Many ways to exploit
- Have fun 😊

Important Notes

- Please strictly follow the account naming rules and submission name rules
- Do NOT try to attack the web application in ways other than those specified in this lab

The End

- Any Questions?