Experiment 2

Student Name: Ruchika Raj	**UID: 20BCS9285**
Branch: CSE	**Section/Group: 615/B**
Semester: 5th	**Date of Performance:04/09/2022**
Subject Name: Web and mobile Security Lab	**Subject Code: 20CSP-338**

1. **Aim/Overview of the practical:**

   Design a method to simulate the html injection and cross site scripting to exploit the attackers.

2. **Task to be done/ Which logistics used:**

   Objective: To analyse Http traffic.

   Software/Hardware Requirements:
   Windows 7 & above version
   Wireshark Packet Sniffer and Packet Capture Library Microsoft Word. Win Zip as necessary

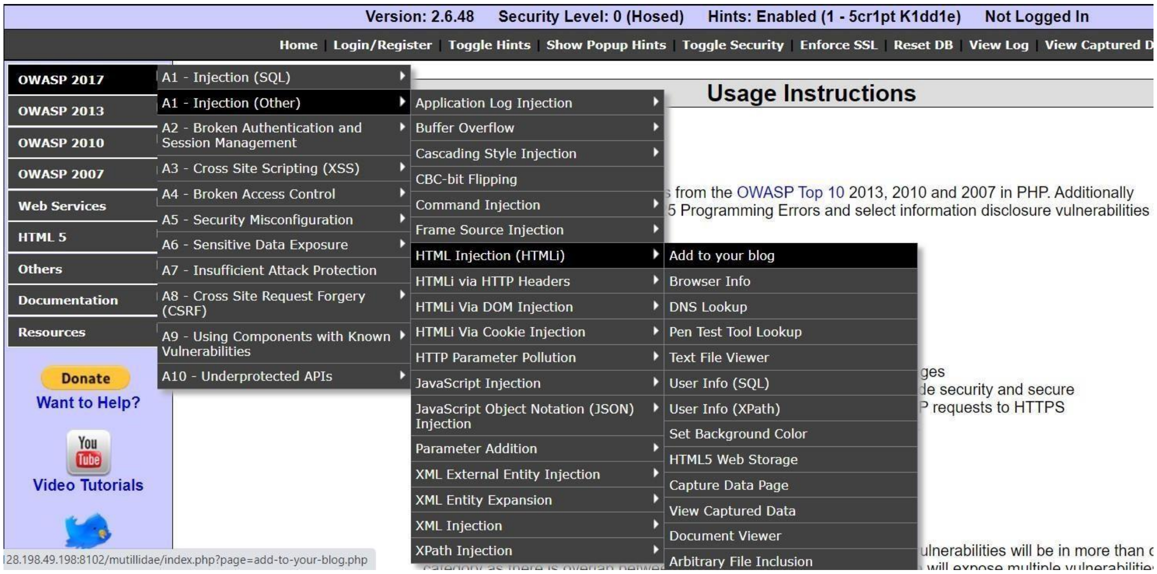
3. **Result/Output/Writing Summary:**

HTML Injection

1. Open website : **OWASP Mutillidae II: Web Pwn in Mass Production**

(URL: http://128.198.49.198:8102/mutillidae/index.php?page=documentation/usageinstructions.php)

2. Now, we’ll be redirected to the web page which is suffering from an **HTML Injection vulnerability** which allows the user to submit his entry in the blog.
3. On the left hand side, click on OWASP 2017□A1-injection(others)□HTML injection□Add to your blog screenshot)


Version: 2.6.48 Security Level: 0 (Hosed) Hints: Enabled (1 - 5cr1pt K1dd1e) Not Logged In
Home | Login/Register | Toggle Hints | Show Popup Hints | Toggle Security | Enforce SSL | Reset DB | View Log | View Captured D
OWASP 2017
OWASP 2013
OWASP 2010
OWASP 2007
Web Services
HTML 5
Others
Documentation
Resources
Donate
Want to Help?
Video Tutorials
A1 - Injection (SQL)
A1 - Injection (Other)
A2 - Broken Authentication and Session Management
A3 - Cross Site Scripting (XSS)
A4 - Broken Access Control
A5 - Security Misconfiguration
A6 - Sensitive Data Exposure
A7 - Insufficient Attack Protection
A8 - Cross Site Request Forgery (CSRF)
A9 - Using Components with Known Vulnerabilities
A10 - Underprotected APIs
Application Log Injection
Buffer Overflow
Cascading Style Injection
CBC-bit Flipping
Command Injection
Frame Source Injection
HTML Injection (HTMLi)
HTMLi via HTTP Headers
HTMLi Via DOM Injection
HTMLi Via Cookie Injection
HTTP Parameter Pollution
JavaScript Injection
JavaScript Object Notation (JSON) Injection
Parameter Addition
XML External Entity Injection
XML Entity Expansion
XML Injection
XPath Injection
Add to your blog
Browser Info
DNS Lookup
Pen Test Tool Lookup
Text File Viewer
User Info (SQL)
User Info (XPath)
Set Background Color
HTML5 Web Storage
Capture Data Page
View Captured Data
Document Viewer
Arbitrary File Inclusion
Usage Instructions

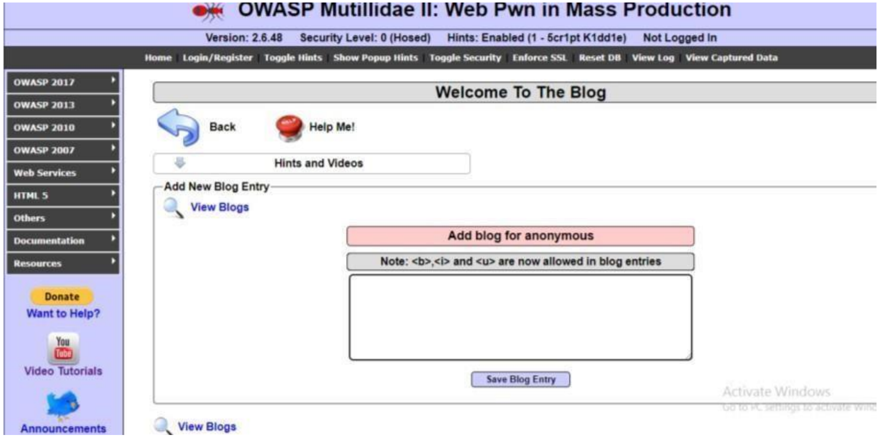
s from the OWASP Top 10 2013, 2010 and 2007 in PHP. Additionally
5 Programming Errors and select information disclosure vulnerabilities

ges
de security and secure
P requests to HTTPS

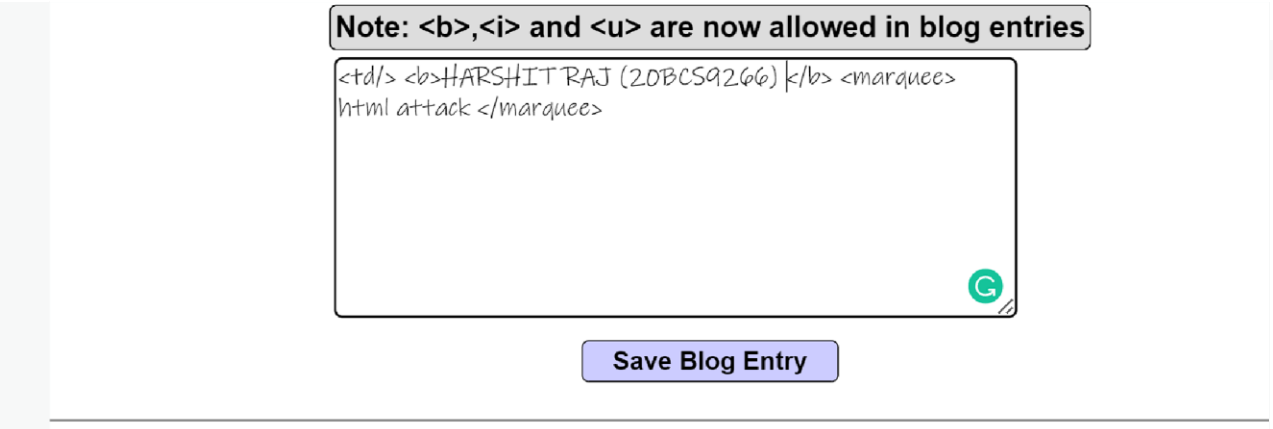
ulnerabilities will be in more than c
will expose multiple vulnerabilitie

128.198.49.198:8102/mutillidae/index.php?page=add-to-your-blog.php

4. Welcome to blog window will appear on the screen. Now, let's try to inject malicious code. Enter the HTML code inside the given text area in order to set up the HTML attack.


OWASP Mutillidae II: Web Pwn in Mass Production
Version: 2.6.48 Security Level: 0 (Hosed) Hints: Enabled (1 - 5cr1pt K1dd1e) Not Logged In
Home | Login/Register | Toggle Hints | Show Popup Hints | Toggle Security | Enforce SSL | Reset DB | View Log | View Captured Data
OWASP 2017
OWASP 2013
OWASP 2010
OWASP 2007
Web Services
HTML 5
Others
Documentation
Resources
Donate
Want to Help?
Video Tutorials
Announcements
Welcome To The Blog
Back
Help Me!
Hints and Videos
Add New Blog Entry
View Blogs
Add blog for anonymous
Note: <b>,<i> and <u> are now allowed in blog entries
Save Blog Entry
Activate Windows
View Blogs

5. For example injected code is : **<td/> CU blog <marquee> html attack </marquee>** then save blog entry

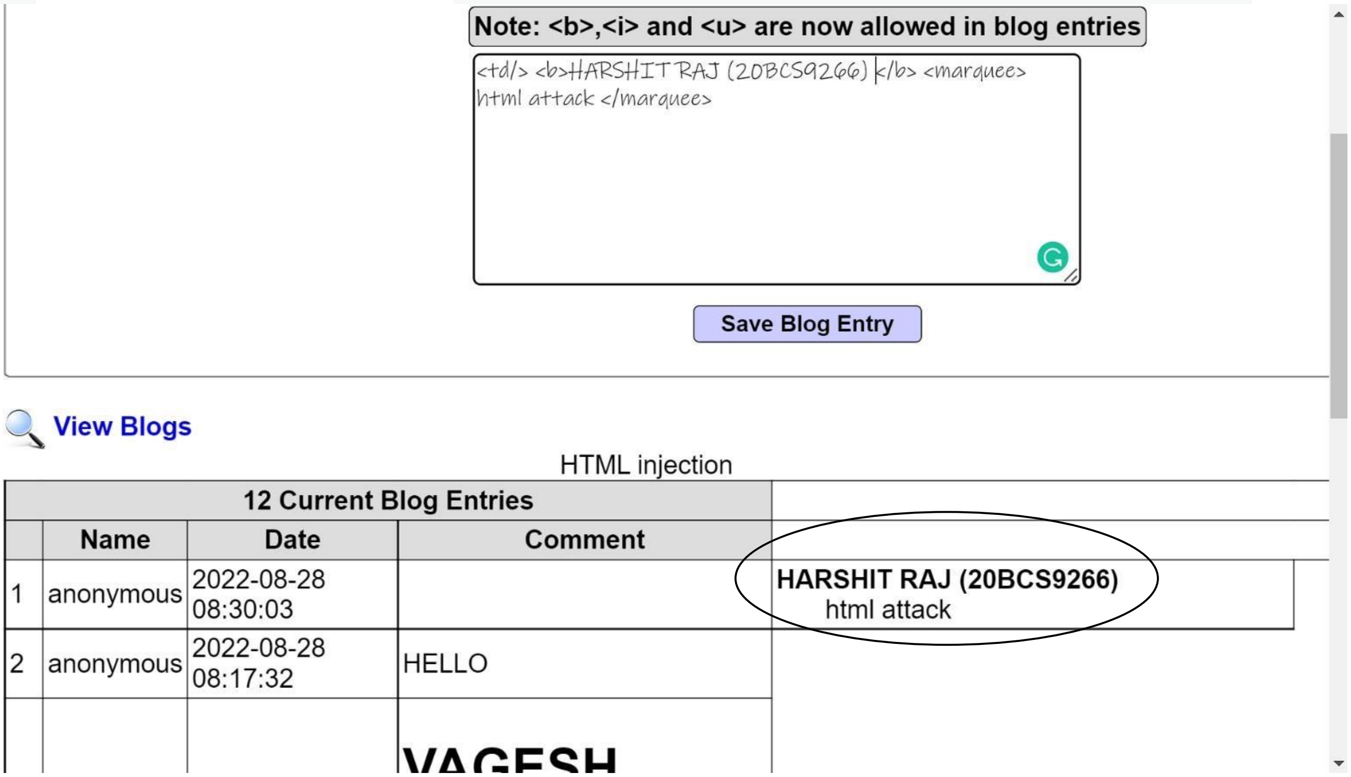


Note: <b>,<i> and <u> are now allowed in blog entries

```
<td/> <b>HARSHIT RAJ (20BCS9266) </b> <marquee>
html attack </marquee>
```

XSS

6. That html code is thus now into the application’s web server, which gets rendered every time whenever the victim visits this malicious page, he’ll always have this code which looks official to him.



Note: <b>,<i> and <u> are now allowed in blog entries

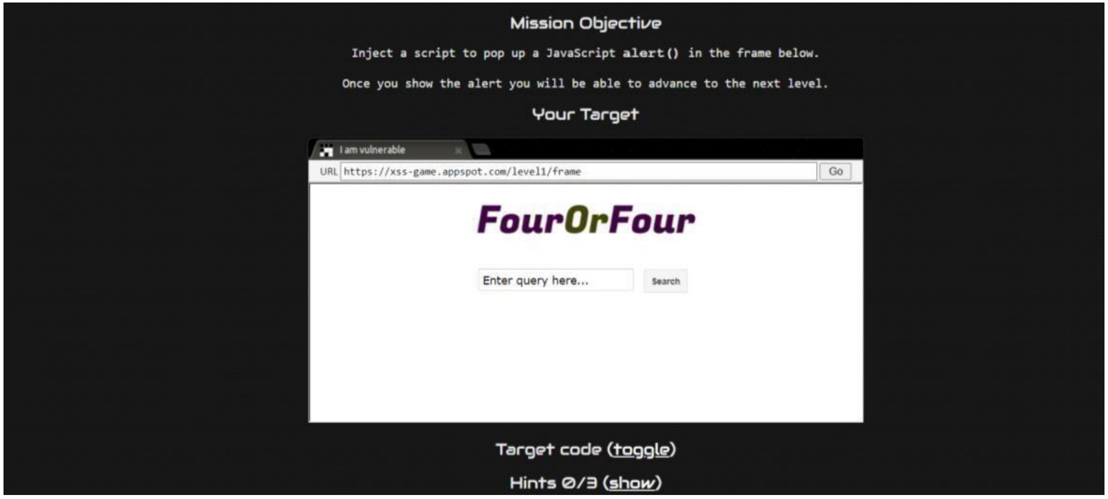
```
<td/> <b>HARSHIT RAJ (20BCS9266) </b> <marquee>
html attack </marquee>
```

HTML injection

12 Current Blog Entries				
	Name	Date	Comment	
1	anonymous	2022-08-28 08:30:03		HARSHIT RAJ (20BCS9266) html attack
2	anonymous	2022-08-28 08:17:32	HELLO	
			VAGESH	

attack

1. Open the link https://xss-game.appspot.com/level1 (or Google XSS game website).



Mission Objective

Inject a script to pop up a JavaScript alert() in the frame below.

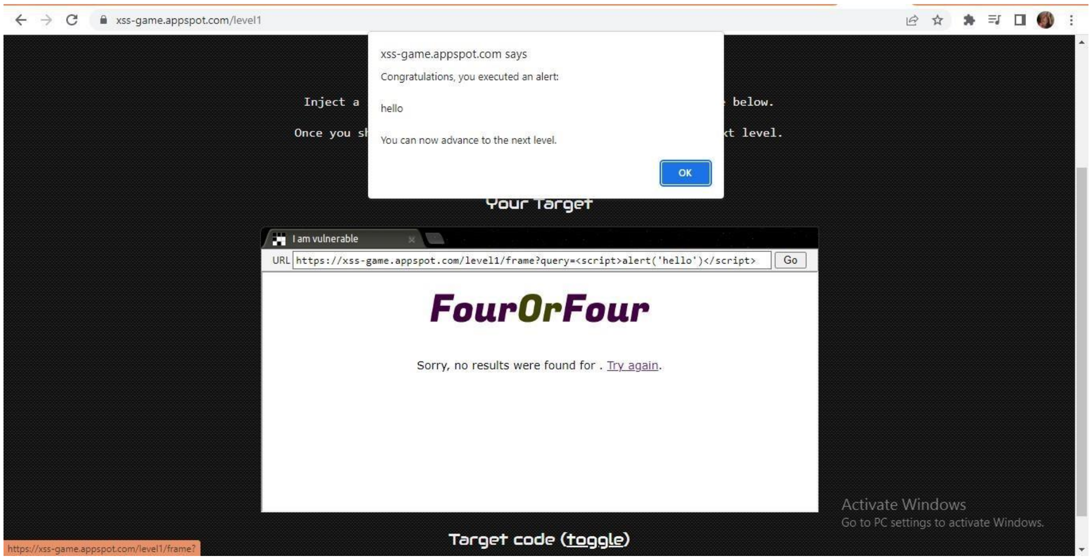
Once you show the alert you will be able to advance to the next level.

Your Target

URL https://xss-game.appspot.com/level1/frame

2. If the search field is vulnerable, when the user enters any script, then it will be executed. Consider, a user enters a very simple script as shown below:

<script>alert(‘ Hello)</script>



xss-game.appspot.com says

Congratulations, you executed an alert:

hello

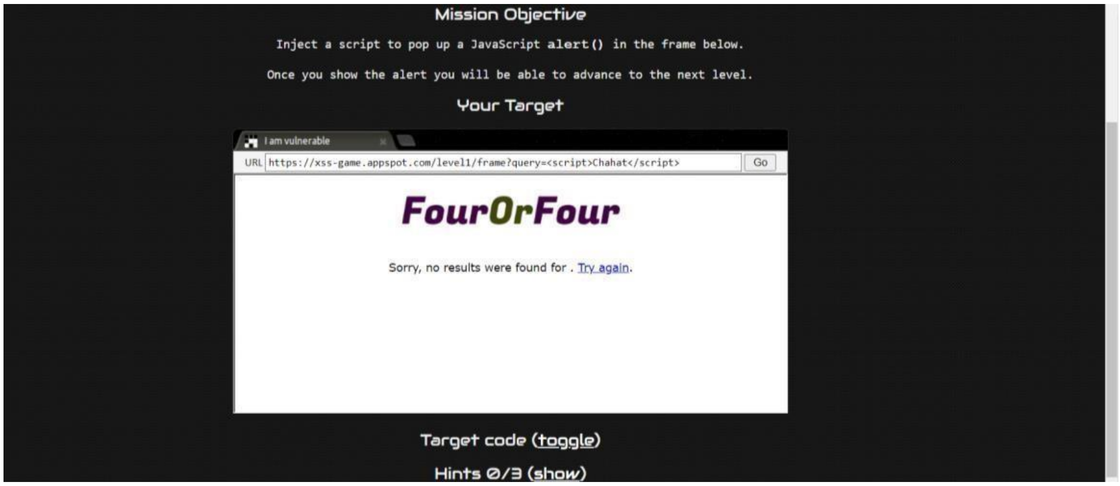
You can now advance to the next level.

URL https://xss-game.appspot.com/level1/frame?query=<script>alert('hello')</script>

Sorry, no results were found for . Try again.

xss-game.appspot.com/level1
OK
Inject a ... below.
Once you sh ... xt level.
Your Target
I am vulnerable
Go
FourOrFour
Activate Windows
Go to PC settings to activate Windows.
Target code (toggle)
https://xss-game.appspot.com/level1/frame?

3. Then after clicking on the **“Search”** button, the entered script will be executed. The script typed into the search field gets executed. This just shows the vulnerability of the XSS attack.



Mission Objective

Inject a script to pop up a JavaScript alert() in the frame below.

Once you show the alert you will be able to advance to the next level.

Your Target

URL https://xss-game.appspot.com/level1/frame?query=<script>Chahat</script>

Sorry, no results were found for . Try again.

Target code (toggle)

Hints 0/3 (show)

Learning outcomes (What I have learnt):

1. **We have learned what HTML injection is and XSS injection .**
2. **An overview of how these attacks are constructed and applied to real system.**
3. **If the app or website lacks proper data sanitization, the malicious link executes the attacker's chosen code on the user's system.**

Evaluation Grid (To be created as per the SOP and Assessment guidelines by the faculty):

Sr. No.	Parameters	Marks Obtained	Maximum Marks
1.			
2.			
3.			