Udajuicer Website Downtime - Vulnerability Assessment

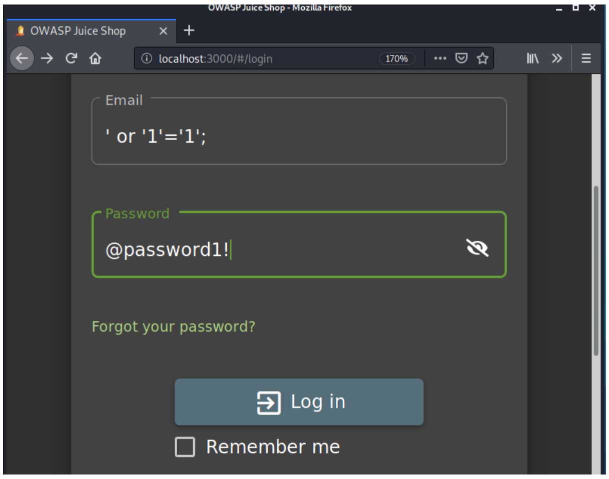
January 4, 2024

Vulnerability Testing

Now that we have made it pass the initial assessment stage, identified the initial attack and shortcomings of the application setup, it is essential to continue with a deeper analysis of the application to see if there are more vulnerabilities. We would exploit two vulnerabilities: SQL Injection and cross site scripting (XSS). The goal is to gain access into the website as an administrator, exploit the site and render a Hacked alert.

1 Testing for SQL Injection Vulnerabilities

Structured Query Language (SQL) injection is a common web attack technique that exploits input validation flaws on applications that accept user input to interact with databases. The image (FIG 1) below shows a screen shot of the commands I used to gain access to Udajuicer's site as an admin



Email
' or '1'='1';

Password
@password1!

Figure 1: A screenshot of the login credentials I used to gain access as an administrator

One can confirm from the account setting image below that I gained access as an admin into the website Using the SQL Injection vulnerability, an attacker can change the account settings of the admin thereby gaining a privilege escalation. It is also possible to gain access to user names of all employees at Udajuicer.

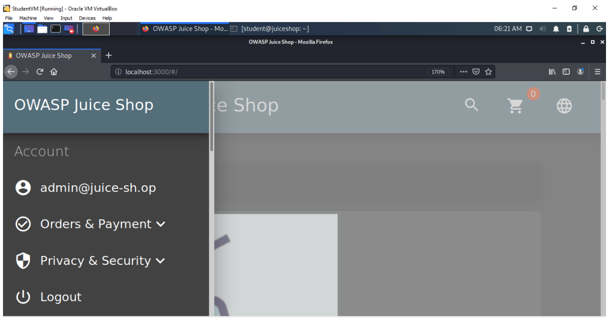
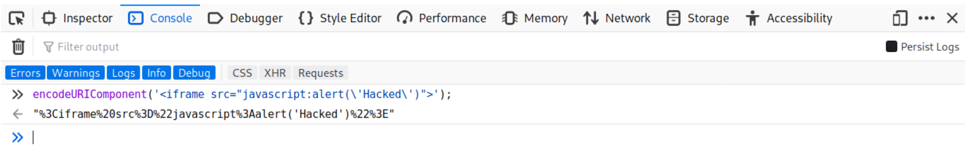


Figure 2: A screenshot showing me as an admin with email address: admin@juice-sh.op

2 Testing for XSS Vulnerabilities

Cross-site Scripting (XSS) is an attack that against an end-user, leveraging a vulnerability on a server that is not performing input or output validation. Here, the user runs a malicious code supplied by the attacker from the vulnerable scanner.



```
>> encodeURIComponent('<iframe src="javascript:alert(\'Hacked\')">');
<- "%3Ciframe%20src%3D%22javascript%3Aalert('Hacked')%22%3E"
>> |
```

Figure 3: A screenshot of the code command

Using the code command, I am attempting to render an alert with the value Hacked. Usually, an attacker would exploit one form of an XSS attack by uploading malicious content to an XSS vulnerable server. The server record and stores the malicious content returning it to any victim that access the vulnerable page. You can see from the image below how the code command has rendered a a message box displaying "Hacked" as shown in the FIG 4 below.

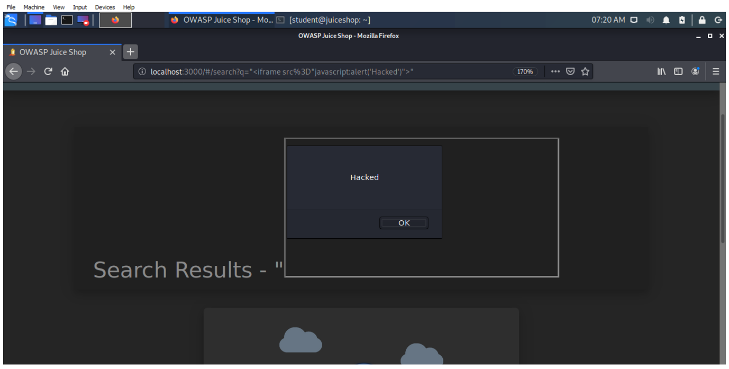

OWASP Juice Shop - Mozilla Firefox
OWASP Juice Shop
localhost:3000/#/search?q="<iframe src%3D"javascript:alert('Hacked')">"
Hacked
OK
Search Results - "

Figure 4: A screenshot of the message box displaying "Hacked"