CTF Reverse Engineering Bootcamp

Session 1: Introduction to Reverse Engineering for CTFs

- **Topics**:
    - Overview of reverse engineering and its role in CTFs.
    - Introduction to key tools (e.g., IDA Pro, Ghidra, Radare2).
    - Setting up a CTF-ready reverse engineering environment.
- **Goal**: Gain a foundational understanding of reverse engineering in the context of CTFs.
- **Practice**: Analyze a simple CTF binary to extract a flag.

Session 2: Assembly Language Basics

- **Topics**:
    - x86 assembly essentials (instructions, registers, memory).
    - Understanding control flow (e.g., loops, conditionals).
    - Disassembling simple CTF binaries.
- **Goal**: Learn to read and interpret assembly code commonly found in CTF challenges.
- **Practice**: Solve a crackme that requires understanding basic assembly instructions.

Session 3: Debugging Techniques for CTFs

- **Topics**:
    - Using debuggers (e.g., OllyDbg, x64dbg, GDB) for CTF tasks.
    - Setting breakpoints and analyzing runtime behavior.
    - Extracting flags by controlling execution flow.
- **Goal**: Master dynamic analysis techniques for CTF problem-solving.
- **Practice**: Debug a CTF challenge to bypass a key check or extract hidden data.

Session 4: Binary Analysis for CTFs

- **Topics**:
    - Static and dynamic analysis tailored to CTF challenges.
    - Identifying key functions, strings, and logic in binaries.
    - Practical CTF examples of binary analysis.
- **Goal**: Develop skills to statically analyze binaries and uncover CTF solutions.
- **Practice**: Analyze a binary from a past CTF to find the flag without execution.

Session 5: Common Encryption Algorithms in CTFs

- **Topics**:
    - Introduction to cryptography basics (symmetric vs. asymmetric, block vs. stream ciphers).
    - Detailed exploration of common algorithms:
        - **XOR**: Properties, usage, and breaking simple XOR encryption.
        - **RC4**: How it works, common weaknesses, and misuse in CTFs.
        - **AES**: Modes of operation and approaches to AES-encrypted data.
        - **DES**: Basics and known vulnerabilities.
        - **ChaCha**: Modern stream cipher usage in challenges.
        - **RSA**: Asymmetric encryption and its role in CTFs.
    - Identifying encryption algorithms in binaries.
- **Goal**: Understand and break common encryption schemes used in CTF challenges.
- **Practice**: Decrypt data in a CTF challenge using identified encryption algorithms (e.g., XOR, RC4).

Session 6: Anti-Analysis and Anti-Reversing Techniques

- **Topics**:
    - Why anti-analysis techniques are used in malware and CTFs.
    - **Obfuscation techniques**:
        - Code obfuscation (e.g., junk code, variable renaming).
        - Control flow obfuscation (e.g., flattening, opaque predicates).
        - Data obfuscation (e.g., string encryption).
    - Other anti-reversing techniques:
        - API hooking to mislead analysis tools.
        - Virtual machine detection.
        - Self-modifying code.
- **Goal**: Recognize and bypass common anti-analysis techniques in CTF binaries.
- **Practice**: Analyze an obfuscated CTF binary and deobfuscate key parts to find the flag.

Session 7: Anti-Debugging Techniques and Packing

- **Topics**:
    - Introduction to anti-debugging and its purpose.
    - Common anti-debugging techniques:
        - Debugger detection (e.g., IsDebuggerPresent).
        - Timing checks to detect debugging pauses.
        - Hardware breakpoint detection.
        - Exception handling to confuse debuggers.
    - **Packing**:
        - What packers are and why they are used.
        - Identifying common packers (e.g., UPX).

        - Unpacking techniques for packed binaries.
- **Goal**: Learn to bypass anti-debugging measures and unpack binaries in CTF challenges.
- **Practice**: Bypass anti-debugging in a CTF binary and unpack a packed executable (e.g., UPX-packed file).

Session 8: CTF Practice and Crackmes

- **Topics**:
    - Full hands-on session with real CTF challenges and crackmes.
    - Applying techniques from all prior sessions (e.g., debugging, encryption breaking, deobfuscation).
    - Tips for effective CTF participation (e.g., time management, teamwork).
- **Goal**: Reinforce skills through practical application in a CTF-style environment.
- **Practice**: Solve a series of reverse engineering CTF challenges and crackmes, simulating a competition setting.