Table of Contents

Reverse Engineering 3
Core Concepts and Architecture 3
Assembly Language and Machine Code 3
Static Analysis 3
Dynamic Analysis and Debugging 3
Injection, Hooking, and Instrumentation 3
Malware Analysis and Software Protection 4

Reverse Engineering

Core Concepts and Architecture

- Introduction to Reverse Engineering: Ethics, Legality, and Use Cases
- Computer Architecture: CPU, Memory Management (RAM), Stack, and Heap
- OS Internals: Windows API (Win32), Linux Syscalls, and Process Structures
- Safe Lab Setup: Virtual Machines (VM), Snapshots, and Sandbox Environments

Assembly Language and Machine Code

- Registers, Memory Addressing, and Flags
- x86 and x64 Assembly Instructions (MOV, PUSH, POP, CALL, JMP)
- ARM Assembly Fundamentals (for Mobile, IoT, and Apple Silicon)
- Calling Conventions: cdecl, stdcall, fastcall, x64 ABI
- Control Flow: Assembly Representations of If-Else and Loops

Static Analysis

- Executable File Formats: PE (Windows), ELF (Linux), Mach-O (macOS)
- Basic Static Tools: Strings, Binwalk, Detect It Easy (DiE), CFF Explorer
- IDA Pro / IDA Free: Disassembly, Graph View, Cross-References (XREF)
- Ghidra: Project Management, Decompiler Usage, and Code Structuring
- Static Analysis Automation: IDAPython and Ghidra Scripting

Dynamic Analysis and Debugging

- Intro to Dynamic Analysis & Behavioral Monitoring (Procmon, Wireshark, Regshot)
- Debugger Tools: Using x64dbg, GDB, OllyDbg, and WinDbg
- Breakpoints: Software (INT3), Hardware, and Memory Breakpoints
- Stepping: Step In (F7), Step Over (F8), and Call Stack Tracing
- Binary Patching: Modifying Code in Memory (JMP, NOPing) and Saving Patches

Injection, Hooking, and Instrumentation

- DLL Injection Techniques (CreateRemoteThread, SetWindowsHookEx)
- API Hooking Logic: Detours and IAT (Import Address Table) Hooking
- Frida: Dynamic Instrumentation and Mobile Reverse Engineering
- Memory Scanning and Pointer Finding with Cheat Engine

Malware Analysis and Software Protection

- Code Obfuscation Techniques and Deobfuscation
- Packers (UPX, Themida) and Cryptographic Identification
- Unpacking: Finding the OEP (Original Entry Point) and Memory Dumping
- Bypassing Anti-Debugging and Anti-VM (Virtual Machine Detection) Techniques
- Extracting IoCs (Indicators of Compromise) and Reporting

Taken from UCH Wiki.