

**Mastering Exploit Development (MED) — Course Content**

**Module 1 – Introduction to Exploit Development**

* What is exploit development?
* Ethical and legal considerations
* Exploit categories: local, remote, client-side
* Understanding vulnerabilities vs. exploits

**Module 2 – System Architecture & Memory Fundamentals**

* CPU architecture (x86, x64, ARM)
* Memory segments: stack, heap, data, code
* Endianness and alignment
* Registers and calling conventions

**Module 3 – Software Vulnerabilities Overview**

* Common vulnerability types (BOF, UAF, format strings)
* Root causes and examples
* Vulnerability discovery process

**Module 4 – Setting Up the Exploit Development Lab**

* Linux & Windows lab setup
* Debuggers (GDB, WinDbg, x64dbg)
* Disassemblers (IDA Pro, Ghidra, Radare2)
* Safe testing environments

**Module 5 – Stack-Based Buffer Overflows**

* Anatomy of a stack overflow
* Overwriting EIP/RIP
* Crafting simple shellcode
* Exploiting basic vulnerable programs

**Module 6 – Shellcode Development**

* Shellcode basics and constraints
* Writing Linux & Windows shellcode in assembly
* Encoding and avoiding bad characters
* Testing shellcode in exploits

**Module 7 – Structured Exception Handler (SEH) Exploits**

* Understanding SEH in Windows
* SEH-based buffer overflow exploitation
* SafeSEH and SEHOP bypass techniques

**Module 8 – Format String Vulnerabilities**

* Basics of format string bugs
* Memory reading and writing via format strings
* Exploiting to overwrite function pointers or GOT entries

**Module 9 – Heap-Based Exploitation**

* Heap structure and allocation behavior
* Exploiting heap overflows
* Use-after-free (UAF) vulnerabilities
* Heap spraying techniques

**Module 10 – Return-Oriented Programming (ROP)**

* What is ROP and why it’s needed
* Building ROP chains
* Bypassing DEP/NX protections with ROP
* Automated ROP chain generation tools

**Module 11 – Exploiting Modern Memory Protections**

* Address Space Layout Randomization (ASLR) bypass
* Data Execution Prevention (DEP) bypass
* Stack canaries and PIE bypass techniques

**Module 12 – Exploiting Race Conditions**

* Identifying race conditions
* TOCTOU vulnerabilities
* Exploiting multi-threaded environments

**Module 13 – Integer Overflows & Type Confusion**

* Signed vs. unsigned integers
* Exploiting integer overflows for memory corruption
* Type confusion attacks

**Module 14 – Kernel Exploitation Basics**

* Kernel mode vs. user mode
* Kernel memory layout
* Local privilege escalation via kernel bugs

**Module 15 – Fuzzing for Vulnerability Discovery**

* Mutation vs. generation fuzzing
* Tools: AFL, Honggfuzz, Peach Fuzzer
* Automating fuzzing workflows

**Module 16 – Exploit Development for Network Services**

* Writing exploits for TCP/UDP services
* Remote buffer overflows
* Exploiting protocol parsing vulnerabilities

**Module 17 – Client-Side Exploitation**

* Exploiting browsers, PDF readers, and media players
* Drive-by downloads
* Social engineering for client-side attacks

**Module 18 – Metasploit Exploit Modules**

* Writing custom Metasploit modules
* Integrating exploits into Metasploit framework
* Payload customization and evasion

**Module 19 – Reporting & Responsible Disclosure**

* Documenting exploit steps
* Writing PoCs responsibly
* Coordinating with vendors for patching

**Module 20 – Final Exploit Development Project**

* Discover a vulnerability in a test application
* Develop a working exploit bypassing modern protections
* Submit final PoC and presentation