Course Title: Making Exploit Development Fun

Exploit Development is often seen as one of the most mysterious arts in cybersecurity, with limited access to comprehensive training.

This course aims to bridge the gap between software proliferation and the skills needed to identify and exploit vulnerabilities.

From basic concepts to advanced techniques, participants will journey through the essentials of exploit development, focusing on SCADA and IoT environments,

with hands-on practice in real-world scenarios.

Course Objectives:

- Understand the foundational concepts of exploit development.
- Learn to identify and exploit common vulnerabilities like buffer overflows, format string issues, and more.
- Develop practical skills in shellcode writing and advanced exploitation techniques.
- Gain insights into vulnerability research, particularly in critical infrastructure settings.

Course Structure:

Module 1: Overview and Background Knowledge

- Understanding Compilers, Linkers, and Loaders

- Deep dive into x86 and x86-64 architectures

- Linux File Permissions and Set-UID Programs

- Memory Mapping in Linux Processes

- System Calls and Shell Variables

- Introduction to Basic Reverse Engineering

- ELF File Format

Module 2: Stack-based Buffer Overflow

- What is a Stack-based Buffer Overflow?

- Practical Exploitation Techniques

- Defenses and Countermeasures

Module 3: Shellcode Development

- Writing Shellcode from Scratch

- Types of Shellcodes: Bind, Reverse, etc.

- Shellcode Evasion Techniques

Module 4: Format String Vulnerabilities

- Understanding Format String Bugs

- Exploiting Format Strings for Memory Leaks and Code Execution

Module 5: Heap-based Buffer Overflow

- Heap Management Basics

- Exploiting Heap Overflows

Module 6: Integer Overflow

- Integer Vulnerabilities Overview

- Practical Exploitation of Integer Overflows

Module 7: Return-Oriented Programming (ROP)

- ROP Basics

- Advanced ROP Chains

- Bypassing DEP and ASLR

Module 8: Advanced Vulnerability Analysis Techniques

- Fuzzing Methodologies

- Introduction to Symbolic Execution

- Tools and Techniques for Vulnerability Discovery

Module 9: Hands-on: Exploit Development for Open Source Software

- Guided Exploit Development Workshops

- Case Studies on Real Vulnerabilities in Open Source Projects

- Live Debugging and Exploitation Practice

Teaching Methodology:

- Lectures: To cover theoretical aspects with real-world examples from SCADA and IoT.

- Interactive Sessions: Q&A sessions after each module.

- Workshops: Hands-on labs with tools like GDB, IDA Pro, and custom environments.

- Guest Lectures: Insights from industry experts, possibly including your own experiences or those from similar domains.

- Project: Students work on a mini-project where they attempt to find and exploit a vulnerability in an open-source software under supervision.

Assessment:

- Participation in workshops

- Mini-project evaluation

- Quizzes on each module to ensure understanding

Materials Needed:

- Virtual machines pre-configured with necessary tools

- Access to a lab environment (could be cloud-based or local setup)

- Course slides, practical guides, and a reference list of tools and further readings.