macOS Exploitation and Penetration Testing (OSMR) Syllabus

Course Summary and Organization of Content	
Course Summary	macOS Control Bypasses (EXP-312) is an in-depth course that explores various macOS security mechanisms and techniques to circumvent them. Learners will delve into essential topics such as binary analysis, shellcode crafting, dylib injection, function hooking, and more. Through hands-on labs and real-world challenges, participants will gain practical experience in identifying and exploiting macOS vulnerabilities, enabling them to tackle complex macOS exploitation scenarios and advance their careers in the cybersecurity field.
Organization of Content	Learning material is divided into **Learning Modules.** Each Learning Module has multiple **Learning Units**, which are digestible, atomic pieces of learning material designed for efficient comprehension and application. This structure allows learners to easily grasp key concepts and integrate them into their skillset, enabling them to progress smoothly through the course. Each Learning Unit is defined by several **Learning Objectives**: discreet and practical goals for the learner to strive for. Learning Objectives are assessed at the end of each Learning Unit via hands-on **Module Exercises**. Many exercises require the learner to show their skills by interacting with OffSec lab machines. Whenever a walkthrough is part of the course material, there are accompanying videos that correspond with the written content.

Who's This Course Designed For?
The macOS Control Bypasses course (EXP-312) is designed for individuals who are seeking to deepen their understanding of macOS security mechanisms and how to bypass them. It is particularly suitable for security professionals, penetration testers, and software developers who want to enhance their skillset in macOS security analysis, evaluation, and exploitation. The course is also valuable for those responsible for developing and maintaining secure software on the macOS platform.

What You'll Learn	
• macOS binary analysis: Static and dynamic analysis, LLDB debugger, and Hopper. • Crafting shellcodes: Writing shellcode in ASM and C, creating custom shell commands and bind shells. • Dylib injection: Using DYLD_INSERT_LIBRARIES	• macOS Sandbox: Studying sandbox internals, profile language (SBPL), and sandbox escapes. • Bypassing Transparency, Consent, and Control (TCC): Understanding TCC internals and exploiting TCC bypass vulnerabilities. • Symlink and Hardlink attacks: Exploring the filesystem permission model and analyzing CVEs related to symlink and hardlink attacks. • Kernel code execution: Understanding KEXT loading

injection and DYLIB hijacking. • Function hooking: Employing function interposing and Objective-C method swizzling. • Mach Microkernel: Understanding Mach IPC concepts, special ports, and injection via Mach task ports. • XPC attacks: Attacking XPC services, exploiting vulnerabilities in various applications.	restrictions, processes, and exploiting unsigned KEXT load vulnerabilities. • Injecting code into Electron applications: Modifying and injecting code into Electron applications for security testing

Learning Module	Learning Units
macOS Control Bypasses: General Course Information	About The EXP-312 Course
	Provided Materials
	Overall Strategies for Approaching the Course
	About the EXP-312 VPN Labs
	About the OSMR Exam
	Wrapping Up
Virtual Machine Setup Guide	Creating VMs on Apple Silicon
	Installing Xcode
	Homebrew

	Old and Other Software
	Third Party Software
	General System Settings
	Specific VM Instructions
Introduction to macOS	macOS System Overview
	High-Level OS Architecture
	The Mach-O File Format
	Objective-C Primer Wrapping Up
macOS Binary Analysis Tools	Command Line Static Analysis Tools
	Static Analysis with Hopper
	Dynamic Analysis
	The LLDB Debugger

	Debugging with Hopper
	Tracing Applications with DTrace
	Wrapping Up
The Art of Crafting Shellcodes	Writing Shellcode in ASM
	Custom Shell Command Execution in Assembly
	Making a Bind Shell in Assembly
	Writing Shellcode in C
	Wrapping Up
Dylib Injection	DYLD_INSERT_LIBRARIES Injection in macOS
	DYLIB Hijacking
	Wrapping Up
The Mach Microkernel	Mach Inter Process Communication (IPC) Concepts

	Mach Special Ports
	Injection via Mach Task Ports
	BlockBlock Case Study - Injecting execv Shellcode
	Injecting a Dylib
	Wrapping Up
Function Hooking on macOS	Function Interposing
	Objective-C Method Swizzling
	Wrapping Up
XPC Attacks	About XPC
	The Low Level C API: XPC Services
	The Foundation Framework API
	Attacking XPC Services

	Apple's EvenBetterAuthorizationSample
	CVE-2019-20057 - Proxyman Change Proxy Privileged Action Vulnerability
	CVE-2020-0984 - Microsoft Auto Update Privilege Escalation Vulnerability
	CVE-2019-8805 - Apple EndpointSecurity Framework Local Privilege Escalation
	CVE-2020-9714 - Adobe Reader Update Local Privilege Escalation
	Wrapping Up
The macOS Sandbox	Sandbox Internals
	The Sandbox Profile Language (SBPL)
	Sandbox Escapes
	Case Study: QuickLook Plugin SB Escape
	Case Study: Microsoft Word Sandbox Escape
	Wrapping Up

Bypassing Transparency, Consent, and Control (Privacy)	TCC Internals
	CVE-2020-29621 - Full TCC Bypass via coreaudiod
	Bypass TCC via Spotlight Importer Plugins
	CVE-2020-24259 - Bypass TCC with Signal to Access Microphone
	Gain Full Disk Access via Terminal
	Wrapping Up
GateKeeper Internals	File Quarantine
	XProtect
	GateKeeper
	Wrapping Up
Bypassing GateKeeper	CVE-2022-42821 GateKeeper Bypass Using AppleDouble Files
	CVE-2021-30990 GateKeeper Bypass using Symbolic Links

	Wrapping Up
Symlink and Hardlink Attacks	The Filesystem Permission Model
	Finding Bugs
	CVE-2020-3855 - macOS DiagnosticMessages File Overwrite Vulnerability
	CVE-2020-3762 - Adobe Reader macOS Installer Local Privilege Escalation
	CVE-2019-8802 - macOS Manpages Local Privilege Escalation
	Wrapping Up
Getting Kernel Code Execution	KEXT Loading Restrictions
	Sample KEXT
	The KEXT Loading Process
	CVE-2020-9939 - Unsigned KEXT Load Vulnerability
	CVE-2021-1779 - Unsigned KEXT Load Vulnerability

	Changes in Big Sur
	Wrapping Up
Injecting Code into Electron Applications	Setting up an Electron Development Environment
	Creating a Simple Electron App
	The Application
	Environment Variable Injection
	Debug Port Injection
	Source Code Modification
	Protecting Electron Applications
	Wrapping Up
Mount(ain) of Bugs (Archived)	The MAC Framework
	The mount System Call

	Disk Arbitration Service
	CVE-2021-1784 - TCC Bypass Via Mounting Over com.apple.TCC
	CVE-2021-30782 - TCC Bypass Via AppTranslocation Service
	16.6. CVE-2021-26089 - Fortinet FortiClient Installer Local Privilege Escalation CVE-2021-26089 - Exploitation
	Wrapping Up
The Art of Crafting Shellcodes (Apple Silicon Edition)	Writing Shellcode in ASM
	Executing Custom Shell Commands in Assembly
	Making a Bind Shell in Assembly
	Writing Shellcode in C
	Wrapping Up
Mach IPC Exploitation	The Mach Interface Generator (MIG)
	CVE-2022-22639 Exploitation Case Study

	Wrapping Up
Chaining Exploits on macOS Ventura	macOS Ventura Mitigations
	Exploit Chain on macOS Ventura
	Wrapping Up
macOS Penetration Testing	Small Step For Man
	The Jail
	I am (g)root
	CVE-2020-26893 - I Like To Move It, Move It
	Private Documents - We Wants It, We Needs It
	The Core
	Wrapping Up