macOS Vulnerability Research Training

Overview

This 3-day training focuses on macOS Vulnerability Research (VR) for beginner to intermediate students. While intermediate topics will be discussed, the course focuses on bringing security researchers up to speed with macOS's unique protections and vulnerabilities.

This training focuses mostly on logic vulnerabilities as these are hard to systemically mitigate, unlike memory corruptions. With the recent trend of Apple's move towards shipping increasingly robust user and kernelspace memory-protection mitigations it is our belief that logic vulnerabilities are the future of VR on macOS.

Day 1

On day one we kick things off by introducing macOS's architecture, and briefly discuss Objective-C. We will explain how apps work on macOS, what bundles and cryptexes are and why are they important. We will introduce and explain Apple's unique protections. You will get to know GateKeeper, Launch Constraints, the application Sandbox, TCC, AMFI and last but most importantly the system that underpins these protections: SIP.

With a good idea about the general organization and protection schemes in mind we will turn to introducing some tooling which you will be expected to get hands-on with during the course. These are static and dynamic analyzers, tools for debugging filesystem operations, and if time permits we will talk a bit about advanced topics like dtrace.

Next we will dive into process injection on macOS, a now largely mitigated, but still crucial aspect that every researcher must be familiar with. We will learn the basic process injection techniques.

Next we will deep dive into the platform Sandbox profile, learn how we can extract and decompile it, and then use that knowledge to analyze and exploit CVE-2025-43328, which allowed someone to gain access to all private data on the system.

To close the day off we will learn about CVE-2024-27883, a PackageKit exploit. We will review the vulnerability, the exploit and discuss any further questions that you might have.

Day 2

The second day we'll focus on filesystems, a fertile ground for serious vulnerabilities. We will learn about filesystem history and the POSIX standard. You will be introduced to core filesystem concepts, like inodes, file object types and the path resolution process. We will take some time to dive into the arguably most important aspect of file handling: the permission model. All of this is crucial to understanding the myriad of traps a developer can fall into while writing software for macOS.

We will study CVE-2023-32428 (badmalloc), and explain the techniques that allowed this seemingly innocuous bug to be exploited after being in the codebase for ~20 years. You will see how a cascade of failing preventions eventually lead to a user to root LPE.

After lunch we will review another bug, CVE-2025-43268 (cryptexctl), that is a simpler vulnerability but nevertheless powerful in not only turning user access into root, but also to explain some library injections in current, present-day code.

Finally we will talk about attacking Electron applications, and see how 3rd party software on macOS is particularly ill-equipped to leverage the mitigations that Apple's own software has. Multiple tricks will be showcased that were used to bypass TCC via Discord.

Day 3

On the final day we take a look at some theory about exploitation challenges, tips and tricks. We will do so by analyzing all three of batsignal's versions (CVE-2022-32801, CVE-2022-26704 and CVE-2023-40443). This vulnerability perfectly showcases what can go wrong if a wrong approach is taken to fix a vulnerability, as the patch was bypassed two times.

After this, we'll take a look at CVE-2023-38571 (librarian), which is a relatively simple bug and exploit, but it touches on a particularly interesting topic: filesystem race conditions. Through this bug and the accompanying exercises we will understand and learn to exploit such race conditions.

After lunch we come back to learning about XPC, a crucial aspect of macOS and macOS VR. We will learn what XPC is, how it is implemented, why it's necessary and the most common ways we can attack it. We will analyse a simple XPC exploit affecting Zoom do demonstrate.

As we near the end of the training we will discuss another Zoom XPC exploit, and demonstrate how we can overcome XPC client validation and racy package verifications.

Learning Objectives

Student will learn about:

- macOS architecture
- Basic static and dynamic analysis tools
- Filesystem concepts: filesystem objects, path resolution, inodes and the VFS
- Apple's FS quirks and the resulting attack surfaces
- Common process injection techniques and their limitations
- XPC services and how to attack them
- Common Electron application weaknesses
- What is SIP and how we can bypass it
- Attacking file system issues for fun, LPE and TCC bypass

Agenda

Day 1 - Warm-Up

Module 1: Intro to macOS

- Intro to macOS architecture
- Basic File System organization, APFS, Cryptexes, Bundles
- Basic Objective-C
- What is TCC, GateKeeper, Sandbox

Module 2: Using Tools

- Binary Ninja
- lldb
- fs_usage and fs_usage_ng

Module 3: Intro to Process Injection on macOS

- DYLD_INSERT_LIBRARIES
- dylib hijacking / proxying

Module 4: CVE-2024-27883 case study

- Learn about PackageKit and rootless entitlements
- CVE-2024-27883 vulnerability overview
- CVE-2024-27883 exploitation

Module 5: CVE-2025-43328 case study

- Intro to macOS Sandbox Profile
- Reversing the Platform Profile (=SIP)
- CVE-2025-43328 vulnerability overview and exploitation

Day 2 - FS intro and Easy Exploits

Module 6: FS Theory

- Quick history of Filesystems

- Basic view of filesystem objects
- POSIX history, philosophy and the most interesting APIs
- Filesystems in the bigger picture
- The VFS
- Basic POSIX permissions
- The extended POSIX permissions: ACLs
- File flags
- Extended attributes
- SIP
- Path resolution

Module 7: CVE-2023-32428 case study badmalloc

- Exploitation tips and macOS quirks
- showcase string truncation bugs
- showcase racing
- showcase fd leak trick

Module 8: CVE-2025-43268 cryptexctl

- What is chroot
- CVE-2025-43268 vulnerability overview
- CVE-2025-43268 exploitation

Module 9: Attacking Electron Applications - exploiting Discord left and right

- All 3 ELECTRON injection tricks to bypass TCC

Day 3: Getting Advanced

Module 10: batsignal case study

- Overcoming exploitation challenges
- explain: SIP and it's ineffectiveness due to user mounting
- demonstrate:
- v1 - CVE-2022-32801 - symlink
- v2 - CVE-2022-26704 - hardlink with union
- v3 - CVE-2023-40443 - mountpoint moving
- talk about races briefly, as some racing was required

Module 11: CVE-2023-38571 librarian case study

- Anatomy of race conditions
- showcase the trivial rename() race in Music

Module 12: XPC madness

- what is XPC
- 2 APIs for XPC
- how to make secure XPC communication

Module 13: Easy Zoom XPC exploit

- just a simple XPC exploit

Module 14: Advanced Zoom XPC exploit

- version restriction technique
- bypass pkg verification

Prerequisites

Students should have the following skills in order to successfully participate in the class:

- User level familiarity with macOS
- Capable of performing basic administrative tasks on macOS (change settings)
- Familiarity with basic security concepts
- Basic scripting skills in bash and Python
- Basic understanding of the C programming language
- Very basic understanding of ARM64 assembly

Required software/hardware

- Apple Silicon hardware, which:
- is capable of running the latest version of macOS (Tahoe)
- is capable of running at least 1 VM
- has enough disk space to store 2 VMs (~100GB)

- has the latest version of Xcode (26) installed
- you are the admin user of, and can install software or change settings if required

Trainer BIOs

Gergely is a independent security researcher working mainly on the Apple Security Bounty program, with a research focus on logic vulnerabilities. He has presented his findings at OBTSv6, and blogs at https://gergelykalman.com So far he has found multiple user to root LPEs, multiple TCC bypasses, an app sandbox escape, along with other bugs. He enjoys trying to exploit the unexploitable, as evidenced by multiple bugs of his that were hiding in plain sight for years or in one case, for decades.

Csaba is a Principal macOS Security Researcher working at Kandji, focusing on vulnerability research and EDR detection development. He currently has over 100 CVEs issued by Apple for vulnerabilities ranging from simple info leaks to full macOS exploit chains bypassing all security controls. He frequently presents his findings on conferences, like BlackHat, Objective By The Sea, POC, and many others. Prior Kandji Csaba worked for OffSec developing the EXP-312 training about macOS exploitation.