Attacking the Linux Kernel — Zer0Con 2023

Training proposal
Andrey Konovalov, xairy.io
Jan 5th 2023

Title

Attacking the Linux Kernel

Overview

This training guides researchers through the field of Linux kernel security. In a series of exercise-driven labs, the training explores the process of finding, assessing, and exploiting kernel bugs in a modern Linux distribution on the x86-64 architecture.

Besides providing a foundation for writing Linux kernel exploits, the training covers the no-less important areas of finding kernel bugs and evaluating their security impact. This includes chapters on using and extending dynamic bug-finding tools, writing custom fuzzers, and analyzing crashes.

The training starts with the beginner topics but proceeds into a few advanced areas as well.

Key learning objectives

- Security-relevant Linux kernel internals and attack surface.
- Practical usage of kernel dynamic bug detectors.
- Internals and extension of Kernel Address Sanitizer (KASAN).
- Writing and evaluating kernel-specific fuzzing harnesses.
- Collecting kernel code coverage with KCOV.
- Practical usage and basic internals of syzkaller.
- Kernel privilege escalation techniques.
- In-kernel Return-Oriented Programming (ROP).
- KASLR, SMEP, SMAP, and KPTI bypasses.
- Exploiting stack, global, and slab (heap) vulnerabilities.
- Exploiting use-after-free (UAF) and out-of-bounds (OOB) vulnerabilities.

Student requirements

- Working C knowledge.
- Familiarity with x86-64 architecture and x86-64 assembly.
- Familiarity with GDB (GNU Debugger).

- Familiarity with common types of vulnerabilities and exploitation techniques for userspace applications.

No knowledge about Linux kernel internals is required.

Hardware requirements

- At least 100 GB of free disk space.
- At least 12 GB of RAM.
- Ability to plug in an untrusted USB drive (relevant for corporate laptops).

Software requirements

- Host OS: Linux (recommended) or Windows.
- VMWare Workstation Player.
- 7-Zip.

Provided to students

A USB drive with:

- Presentation slides.
- Detailed lab guides with step-by-step instructions.
- Virtual machine images with tools, exercise binaries, and source code.

Course agenda

Day 1 — Internals and Sanitizers:

- Internals and debugging: x86-64 architecture refresher; introduction to the Linux kernel; attack surface; types of vulnerabilities; setting up a kernel debugging environment; using GDB to debug the kernel and its modules.
- Detecting bugs: using KASAN to detect and analyze memory corruptions; KASAN internals; extending KASAN; KMSAN and other Sanitizers; reading kernel bug reports; assessing impact of kernel bugs.

Day 2 — Fuzzing:

- General fuzzing: writing and evaluating kernel-specific fuzzing harnesses; Human-in-the-Loop fuzzing; collecting coverage with KCOV; using KCOV remote coverage; fuzzing externally-triggerable code paths.
- Fuzzing with syzkaller: API-aware fuzzing; coverage-guided fuzzing; using syzkaller; writing syscall descriptions.

Day 3 — Escalating privileges and bypassing mitigations:

- Escalating privileges: ret2usr; overwriting the cred structure; overwriting modprobe_path; kernel stack buffer overflows; arbitrary address execution and arbitrary read/write primitives.
- Bypassing mitigations: KASLR, SMEP, SMAP, and KPTI bypass techniques; in-kernel Return-Oriented Programming (ROP); out-of-bounds vulnerabilities; information leaks.

Day 4 — Exploiting slab corruptions:

- Exploiting basic slab corruptions: slab out-of-bounds and use-after-free vulnerabilities; slab-specific mitigations; slab spraying; data-only exploitation; the unlinking attack.
- Modern slab exploitation techniques: userfaultfd and FUSE; elastic objects; cross-cache corruptions.
- Beyond: learning advanced exploitation techniques; useful references.

Trainer's bio

Andrey Konovalov is a security researcher focusing on the Linux kernel.

Andrey found multiple zero-day bugs in the Linux kernel and published proof-of-concept exploits for these bugs to demonstrate the impact. Andrey is a contributor to several security-related Linux kernel subsystems and tools: KASAN — a fast dynamic bug detector, syzkaller — a production-grade kernel fuzzer, and Arm Memory Tagging Extension (MTE) — an exploit mitigation.

Andrey spoke at security conferences such as OffensiveCon, Android Security Symposium, Linux Security Summit, LinuxCon, and PHDays. Andrey also maintains a collection of Linux kernel security–related materials and a channel on Linux kernel security.

See xairy.io for all Andrey's articles, talks, and projects.