# Exploiting the Android Kernel (3 days)

Training description

Andrey Konovalov, xairy.io

**Title**

Exploiting the Android Kernel

**Overview**

This training guides researchers through the field of Android kernel exploitation. The training is structured as a series of lectures, each followed by one or more hands-on labs executed on a Pixel 8 device. The goal of each lab is to write an Android kernel exploit following the techniques described during the lecture.

The training starts with the chapters on setting up a kernel debugging environment on Pixel 8 and exploiting basic Slab (heap) memory corruptions to escalate privileges. The following core part of the training focuses on modern Android kernel exploitation techniques for memory corruption vulnerabilities.

It's recommended (though not mandatory) for a participant to already have at least some experience with writing Linux kernel exploits (i.e., knowing the basics of exploiting Slab memory corruptions and escalating privileges on x86-64 Linux kernels). Participants without such experience might want to consider taking the Exploiting the Linux Kernel training first.

**Key learning objectives**

- Security-relevant Android kernel internals and attack surface.
- Android kernel privilege escalation techniques.
- Android kernel–specific mitigations and their bypasses.
- Exploiting use-after-free and out-of-bounds vulnerabilities in Slab and page_alloc memory.
- Data-only Android kernel exploitation techniques.
- Cross-cache and cross-allocator attacks.
- Page table–based exploitation techniques.

**Student requirements**

- Working C knowledge.
- Familiarity with ARM64 architecture and assembly.
- Familiarity with GDB.

- Familiarity with common types of vulnerabilities and exploitation techniques for userspace applications.
- Basic experience with Linux kernel exploitation*.

* This training is designed to be more advanced than my Exploiting the Linux Kernel training. Thus, it's expected that a student already knows the basics of exploiting Slab memory corruptions and escalating privileges on x86-64 Linux kernels (though the training will offer refreshers on these topics). Taking this training without any experience of writing Linux kernel exploits might be feasible but will be challenging.

**Hardware requirements**

- **Bring your own** Pixel 8 (not 8a and not 8 Pro) with an unlockable bootloader*.
- Two standard USB Type-C data cables.
- x86-64–based laptop with two available USB ports (or bring a USB hub).
- At least 100 GB of free disk space.
- At least 8 GB of RAM.
- Ability to plug in an untrusted USB drive (relevant for corporate laptops).

* Please have the device on hand at least a week before the training start date to execute the pre-shared testing setup instructions.

**Software requirements**

- Host OS: **Linux only** (preferrably Ubuntu to avoid setup complications).
- Docker.
- 7-Zip.

**Provided to students**

A USB drive with:

- Presentation slides.
- Detailed lab guides with step-by-step instructions.
- Docker image with required tools, kernel images, and source code.

**Course agenda**

Day 1 — Setup and exploitation basics:

- Internals and debugging: ARM64 architecture refresher; setting up kernel debugging environment on Pixel 8; using KGDB to debug Android kernel.

- Escalating privileges: security-relevant Android kernel internals and attack surface; Android kernel–specific mitigations; task credentials, SELinux, seccomp; getting root via arbitrary address read/write primitives.
- Exploiting Slab corruptions: core SLUB internals; Slab-specific mitigations; exploiting Slab out-of-bounds and use-after-free vulnerabilities.

Day 2 — Modern Android kernel exploitation techniques:

- Page pointer corruptions; pipe_buffer-based exploitation techniques; Dirty Pipe.
- Cross-cache and cross-allocator attacks; Dirty Cred.
- Page table–based exploitation techniques; Dirty Pagetable.

Day 3 — Exploiting Android Binder IPC:

- Android Binder IPC: core internals; previous vulnerabilities.
- Writing end-to-end exploit for recent N-day Android Binder vulnerability.
- Learning more exploitation techniques; useful references.

Note: This course is freshly new, so the exact day-by-day agenda is subject to change. But the overall focus will remain: modern Android kernel exploitation techniques.

**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 contributed to several security-related Linux kernel subsystems and tools: KASAN — a fast dynamic bug detector; syzkaller — a widely-used kernel fuzzer; and Arm Memory Tagging Extension (MTE) — an exploit mitigation.

Andrey gave talks at many security conferences such as OffensiveCon, Zer0Con, Android Security Symposium, and Linux Security Summit. Andrey also maintains a collection of Linux kernel security–related materials and a channel on Linux kernel security.

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