

Software-Induced Side Channels in the Linux Kernel




Advisor: **Lukas Maar**

Motivation

The number of vulnerabilities found in the Linux kernel, as well as the number of defenses, has increased significantly. This results in a situation where many potentially exploitable vulnerabilities exist in the kernel, while their exploitation is difficult. Many of the defenses that have been introduced rely on isolating potentially exploitable objects from security-critical objects on a locality basis. Related work [1, 2] has presented a software-induced timing side channel on the memory allocator of the Linux kernel to improve vulnerability exploitation. In particular, Maar et al. [1] showed how their timing side channel bypasses several isolation-based defenses.

This project will further investigate the possibility of using a software-induced timing side-channel attack to bypass additional isolation-based defenses in the Linux kernel. Identifying new timing side channels is important for both attackers and defenders.

Goals and Tasks

-  Get familiar with the Linux kernel and timing side channels
-  Develop a timing side channel leaking information of the kernel
-  Demonstrate and evaluate the side channels



Literature

- > [L. Maar et al.](#)
SLUBStick: Arbitrary Memory Writes through Practical Software Cross-Cache Attacks within the Linux Kernel
[USENIX Security](#)
- > [Y. Lee et al.](#)
PSPRAY: Timing Side-Channel based Linux Kernel Heap Exploitation Technique
[USENIX Security](#)

Courses & Deliverables

- Introduction to Scientific Working**
Short report on background
Short presentation
- Bachelor Project**
Project code and documentation
- Bachelor's Thesis**
Project code
Thesis
Final presentation

Recommended if you're studying

- CS
- ICE
- SEM

Prerequisites

- > Programming: C/C++
- > Interest in the Linux kernel and side channels
- > Interest in system security

Advisor Contact

lukas.maar@tugraz.at