Vrije Universiteit Amsterdam

Bachelor Project Computer Science - Project Proposal

Breaking Kernel ASLR through TLB sidechannel

Author: Mira Chacku Purakal (2777677)

VU : Herbert Bos
Daily supervisor: Dyon

March 7, 2025

Abstract

We will reproduce parts of Maar et al. [2] and use them to perform a location disclosure attack on security critical kernel data structures. This will allow for further attacks which would usually be prevented by ASLR. To do so we will use kernel defenses that change the page mappings to 4kB and exploit a TLB side channel along with allocator massaging to leak the page-aligned address of critical data structures. We then further deduce precise addresses as well as performing error checking to improve success rate.

1 Introduction

The goal of the project is to expose the location of security critical data structures such as the `cred` struct which would usually be prevented through (Kernel) ASLR. To do so we will use one of the technique presented in prior work by Maar et al. [2] using TLB contention patterns caused by certain kernel defenses in combination with allocator massaging and in the process reproduce a subset of their findings. We will demonstrate the attack on an 8th Gen or newer Intel CPU and a 6.8 kernel. Depending on other factors, more systems may be evalutated.

2 Background

Under usual circumstances kernel objects are memory mapped to 2MB pages, however Maar et al. [2] identify 3 kernel defenses which change the memory mapping (partially) to 4kB. With this an attacker can ensure the target object is located in one of those 4kB mappings and loaded into the TLB. Then using access primitives creates a TLB contention pattern, based on which the page-aligned address of the target can be inferred and further the exact address of the target can be deduced.

The 3 kernel defenses are `CONFIG_STRICT_MODULE_RWX`, `CONFIG_SLAB_VIRTUAL` and `CONFIG_VMAP_STACK`. As the name suggests the last one only changes the mapping of the stack to 4kB and therefore only allows leakage of the kernel stack which is not interesting to us. `CONFIG_STRICT_MODULE_RWX` is more interesting, however Maar et al. [2] were unable to reliably leak the `cred` struct specifically using this exploit. This leaves `CONFIG_SLAB_VIRTUAL` which is a kernel defense introduced in the patched kernel for the Google KernelCTF. Other than `CONFIG_STRICT_MODULE_RWX` this changes the entire heap mapping to 4kB instead of just the area around a loaded module. This potentially increases TLB noise but achieving a near 100% success rate should still be possible with use of error correction.

3 Problem

KASLR as a defense obfuscates the location of security critical objects which could be used in many exploits if exposed. If we are able to find a stable exploit to expose the location of these objections and therefore partially break KASLR many attacks previously prevented by it become possible again. In this use case, specifically data-only attacks greatly benefit from the potential data leaked here.

4 Related Work

Maar et al. [2] is clearly related as my work will be largely reproducing a subset of this work and provide everything required to use it for further data-only attacks.
Further Gruss et al. [1] may be relevant as it is used to distinguish mapped pages without violating access permissions.

5 Research Question(s)

We invastigate the location disclosure attacks presented in Maar et al [2] and investigate if we can provide a simple, usable disclosure attack to leak the location of security critical data structures which can be chained with further attacks.

6 Approach

In the first step we will analyze the most feasible way to leak the information of interest. Then we will attempt to perform such a disclosure attack on interesting data structures in a way which can be used in combination with follow-up attacks. If possible a last step may involve testing the attack on later kernels or different hardware.

7 Plan

The first step is mostly theoretical, however it may be necessary to reexamine the decisions made if further challenges arise in later steps.
The next step will be performed on provided hardware, preferrably using VMs but if this proves impossible also on bare metal following the steps learned in the setups of the VM.

8 Conclusion

We will use a kernel defense to force 4kB page mapping for security critical data structures. Then we will use kernel allocator massaging and a TLB side channel to leak their

location despite KASLR being enabled. We will further use error correction to attempt to provide a stable exploit which can be used for further exploits.

References

[1] D. Gruss, C. Maurice, A. Fogh, M. Lipp, and S. Mangard. Prefetch side-channel attacks: Bypassing smap and kernel aslr. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, CCS '16, page 368–379, New York, NY, USA, 2016. Association for Computing Machinery.

[2] L. Maar, L. Giner, D. Gruss, and S. Mangard. When good kernel defenses go bad: Reliable and stable kernel exploits via defense-amplified tlb side-channel leaks. In *Proceedings of the 34rd USENIX Security Symposium*, Proceedings of the 34rd USENIX Security Symposium, United States, Aug. 2025. USENIX Association. 34th USENIX Security Symposium : USENIX Security 2025, USENIX'25 ; Conference date: 13-08-2025 Through 15-08-2025.