ISOLATION
ATTACKS

GRAD SEC
OCT 03 2017
Exploiting the DRAM rowhammer bug to gain kernel privileges

How to cause and exploit single bit errors

Mark Seaborn and Thomas Dullien
ROWHAMMER
ROWHAMMER
ROWHAMMER
ROWHAMMER

Bit Flip
A hardware glitch
Causes charge to leak in DRAM
DRAM row activations cause **bit flips**
NaCl exploit

Safe instruction sequence:

```assembly
andl $\sim 31, %eax  // Truncate address to 32 bits
               // and mask to be 32-byte-aligned.
addq %r15, %rax  // Add %r15, the sandbox base address.
jmp *%rax        // Indirect jump.
```

NaCl sandbox model:
- Prevent jumping into the middle of an x86 instruction
- Indirect jumps can only target 32-byte-aligned addresses
NaCl exploit

Bit flips make instruction sequence unsafe:

```assembly
andl $~31, %eax  // Truncate address to 32 bits
                // and mask to be 32-byte-aligned.
addq %r15, %rax // Add %r15, the sandbox base address.
jmp *%rax        // Indirect jump.
```

e.g. %eax → %ecx

- Allows jumping to a non-32-byte-aligned address
ROWHAMMER + NaCl

Hiding unsafe code in NaCl

Existing technique for exploiting non-bundle-aligned jump:

```
20ea0: 48 b8 0f 05 eb 0c f4 f4 f4 f4
    movabs $0xf4f4f4f40ceb050f, %rax
```

This conceals:

```
20ea2: 0f 05  syscall
20ea4: eb 0c  jmp ...  // Jump to next hidden instr
20ea6: f4  hlt  // Padding
```

Insert ROP-like gadgets in your own code
ROWHAMMER + MEMORY DEDUPLICATION

Virtualization Host

Victim

Attacker

Backing memory
ROWHAMMER + MEMORY DEDUPLICATION

Virtualization Host

Victim
Attacker

Backing memory
ROWHAMMER + MEMORY DEDUPLICATION

Virtualization Host

Victim
Attacker

Backing memory
Copy-on-write (COW) ensures isolation
ROWHAMMER + MEMORY DEDUPLICATION

Virtualization Host

Victim
Attacker

Backing memory
ROWHAMMER + MEMORY DEDUPLICATION

Virtualization Host

Victim
Attacker

Backing memory
Rowhammer + Memory Deduplication

Rowhammer breaks COW
Flip Feng Shui: Hammering a Needle in the Software Stack

Kaveh Razavi*
Vrije Universiteit Amsterdam

Ben Graa*
Vrije Universiteit Amsterdam

Erik Bosman
Vrije Universiteit Amsterdam

Cristiano Giuffrida
Vrije Universiteit Amsterdam

Herbert Bos
Vrije Universiteit Amsterdam

Bar Herned
Katholieke Universiteit Leuven

* Equal contribution; joint first authors

Abstract

We introduce Flip Feng Shui (FFS), a new exploitation vector which allows an attacker to induce bit flips over arbitrary physical memory in a fully controlled way. FFS relies on hardware bugs to induce bit flips over memory and on the ability to surgically control the physical memory layout to corrupt arbitrary targeted data anywhere in the software stack. We show FFS is possible today with very few constraints on the target data, by implementing, for instance, using the Rowhammer bug and memory deduplication (an OS feature widely deployed in production). Memory deduplication allows an attacker to reverse-map any physical page into a virtual page of known size, as long as the page's contents are known. Rowhammer, in turn, allows an attacker to flip bits in controlled (initially unknown) locations in the target page.

We show FFS is extremely powerful: a malicious VM in a practical cloud setting can gain unauthorized access to a co-located victim VM running OpenSSL. Using FFS, we exemplify end-to-end attacks breaking OpenSSL's public-key authentication, and forging UGI signatures from trusted keys, thereby compromising the HTTPS/TLS protocol mechanism. We conclude by discussing mitigations and future directions for FFS attacks.

1 Introduction

The demand for high-performance and low-cost computing translates to increasing complexity in hardware and software. On the one hand, the semiconductor industry packs more and more transistors into chips that serve as a foundation for our modern computing infrastructure. On the software side, modern operating systems are packed with complex features to support efficient resource management in cloud and other performance-sensitive settings. Both trends come at the price of reliability and, inevitably, security. On the hardware side, components are increasingly prone to failures. For example, a large fraction of the DRAM chips produced in recent years are prone to bit flips [34, 51], and hardware errors in CPUs are expected to become mainstream in the near future [16, 16, 37, 53]. On the software side, widespread features such as memory or storage deduplication may serve as side channels for attackers [8, 12, 31]. Recent work analyzes some of the security implications of both trends, but no for the attacks that abuse these technologies/software features have been shown capable of practical privilege escalation [51], in-browser exploitation [12, 20], and selective information disclosure [12, 24].

In this paper, we show that an attacker exploiting modern hardware/software properties can craft much more sophisticated and powerful attacks than previously believed possible. We describe Flip Feng Shui (FFS), a new exploitation vector that allows an attacker to induce bit flips over arbitrary physical memory in a fully controlled way. FFS relies on two underlying primitives: (i) the ability to induce bit flips in controlled (but not predetermined) physical memory pages; (ii) the ability to control the physical memory layout to reverse map a target physical page into a virtual memory address at attacker control. While we believe the general vector will be increasingly common and relevant in the future, we show that an instance of FFS, which we term FFS (i.e. deduplication-based FFS), can already be implemented on today's hardware/software platforms with very few constraints. In particular, we show that by abusing Linux's memory deduplication system (SSSD) which is very popular in production clouds [8], and the widespread Rowhammer DRAM bug [34], an attacker can maliciously flip a single bit in any physical page in the software stack with known contents.

Despite the complete absence of software vulnerabilities, we show that a practical Flip Feng Shui attack can have devastating consequences in a common cloud setting. An attacker controlling a cloud VM can abuse

See the demo starting at about 17:00

https://www.usenix.org/conference/usenixsecurity16/technical-sessions/presentation/razavi