CROWDSTRIKE

HYPERVISORS IN YOUR TOOLBOX

SATOSHI TANDA & TIMO KREUZER,

SOFTWARE ENGINEERS, STRATEGIC RESEARCH INITIATIVES

TAKEAWAYS

- Developing a simple hypervisor is easier than ever
- Even a simple hypervisor can open many possibilities
- Productization is not a trivial task

- Who We Are
- Background
- What HyperPlatform Is
- VT-x and Extended Page Tables
- Challenges
- Conclusion

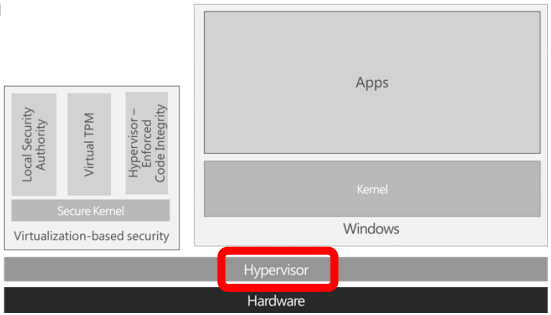
WHO WE ARE

- Satoshi Tanda (@standa_t)
- Creator of HyperPlatform
- Reverse Engineer for years
- Recently joined CrowdStrike

- Timo Kreuzer
- Core contributor to ReactOS
- Interested in hypervisor research
- Joined CrowdStrike 4 years ago

BACKGROUND

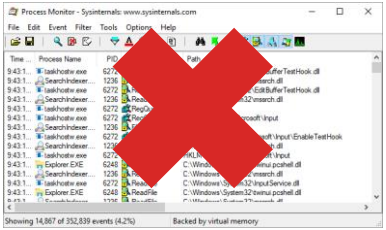
- Technology research, esp. for additional security
  - Memory space isolation and protection
  - Sensitive resource access control
  - Example: virtualization-based security



The VBS architecture: https://technet.microsoft.com/en-us/itpro/windows/keep-secure/windows-10-security-gu

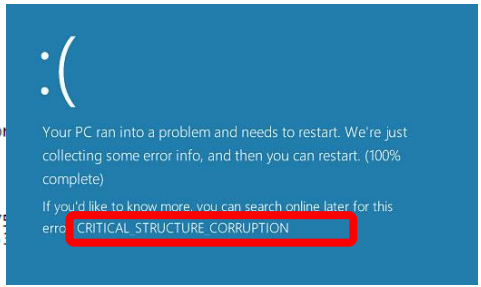
BACKGROUND

- Lack of options for the kernel-mode code reverse engineering
  - Tools?



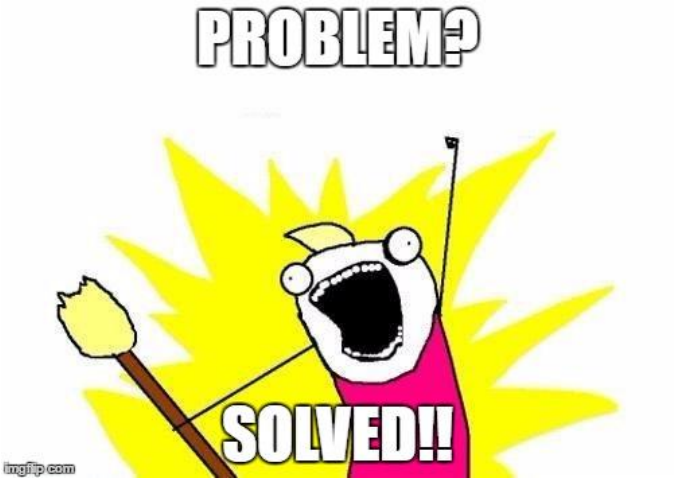
  - Techniques?

```
1: kd> u nt!NtQuerySystemInformation
nt!NtQuerySystemInformation:
fffff800`02d8d6cc jmp     qword ptr [nt!NtQuerySystemInfor
fffff800`02d8d6d2 lock sbb eax,0F80002E7h
fffff800`02d8d6d8 ???
fffff800`02d8d6d9 ???
fffff800`02d8d6da fiadd   dword ptr [rbx+rcx*4-2Fh]
fffff800`02d8d6de jg      nt!NtQuerySystemInformation+0x7
fffff800`02d8d6e0 je      nt!NtQuerySystemInformation+0x5
```



BACKGROUND

- Lack of options? Use Hypervisor!
  - Memory space isolation => invisible API hook
  - Sensitive resource access control => ability to monitor activities




NOPE

CHALLENGES

- Problems in using virtualization technology (VT) by Windows-centric researchers

Flexibility

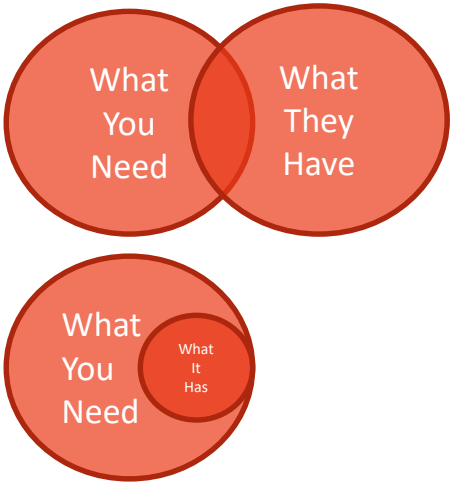
Compatibility

Simplicity

Efficiency

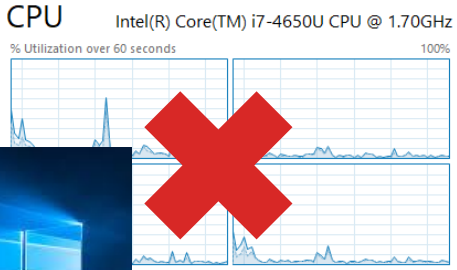
CHALLENGES: FLEXIBILITY

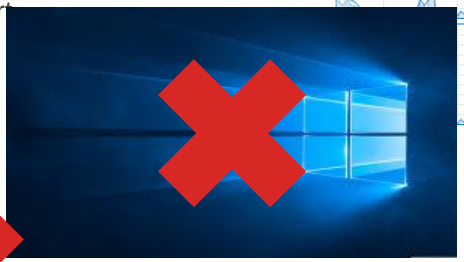
- Hard to repurpose or extend for general usage (security research)
- Designed for specific purposes
  - HyperDbg, VirtDbg, BluePill
- Designed for purely education
  - SimpleVisor



CHALLENGES: COMPATIBILITY

- Not compatible with modern platforms (except SimpleVisor)
  - No multi-processor support
  - No Windows 10 support
  - No x64 support





System type: 64-bit Operating System, x64-based processor

CHALLENGES: SIMPLICITY

- Large code base

6,000 KLOC

3,000 KLOC

- Complicated development and deployment process

300 KLOC

CHALLENGES: EFFICIENCY

- Slow for day-to-day usage

300 KLOC

HYPERPLATFORM

Simplicity

- 8KLOC
- With Visual Studio and Windbg
- No external libs

Compatibility

- Windows 7-10 on x86/x64
- Multi-processors
- Old processors
- On VMware

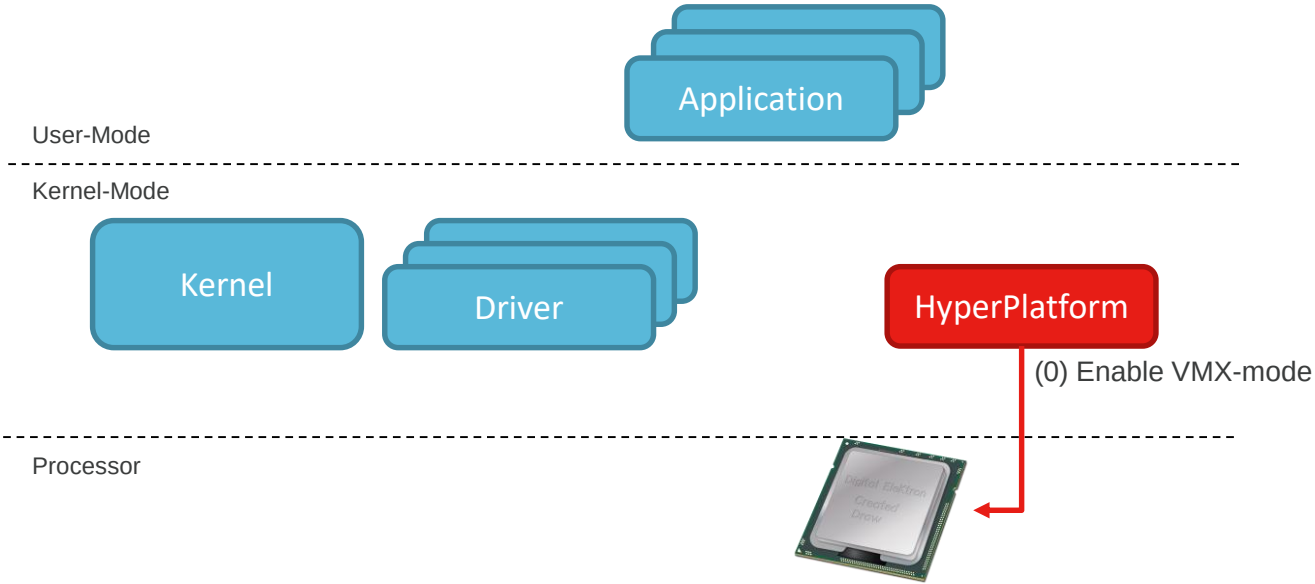
Flexibility

- Designed as a platform
- All major VT features

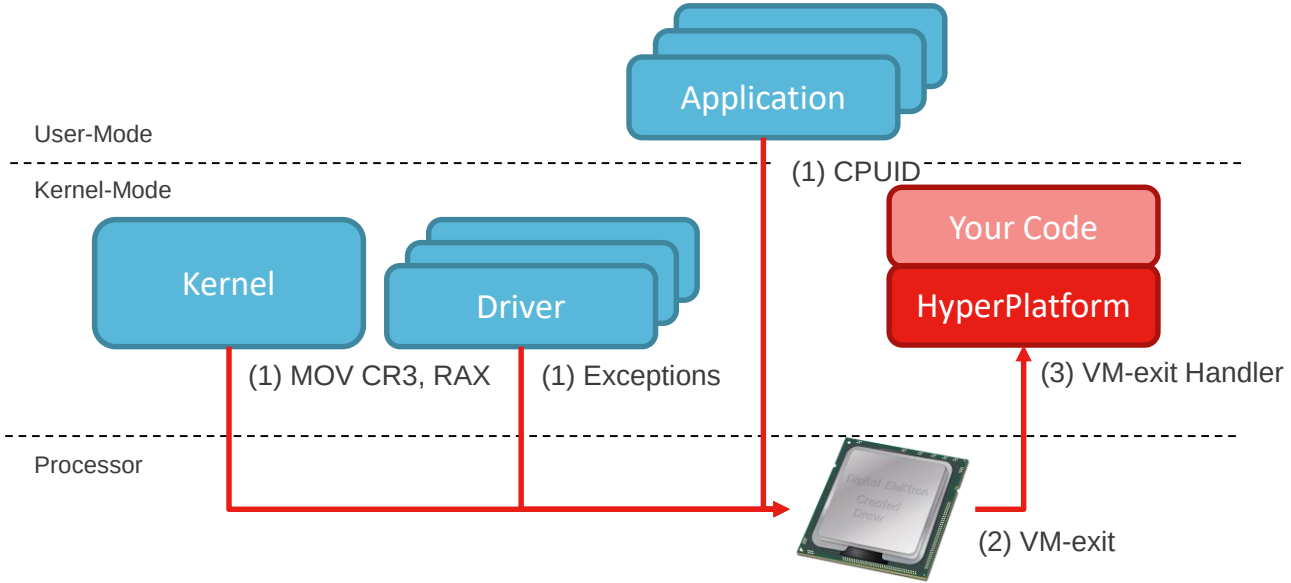
Efficiency

- 10% of performance overhead

HYPERPLATFORM AND VT-X

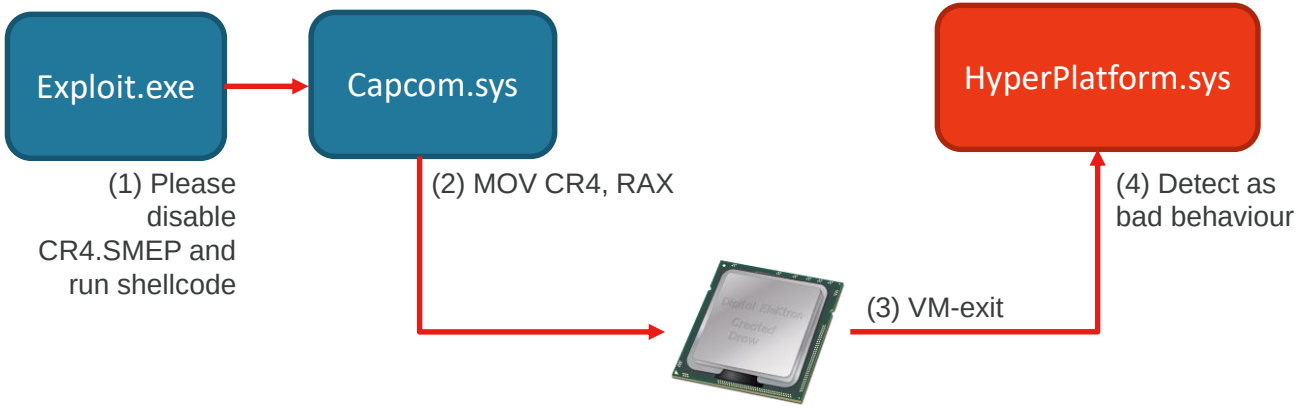


HYPERPLATFORM AND VT-X

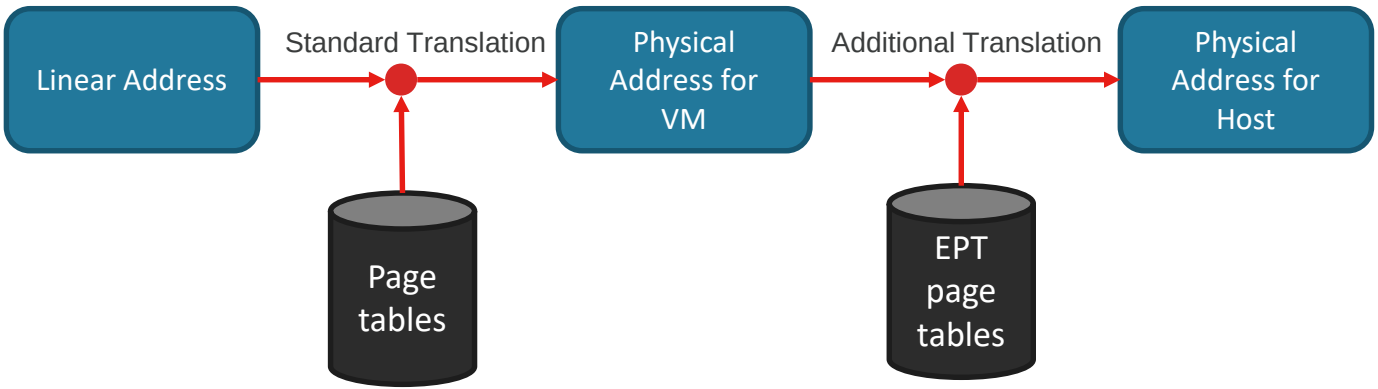


DEMO 1

- Detection of system resource (CR4.SMEP) modification for additional security
- CR4.SMEP bit controls a security feature and should not be modified

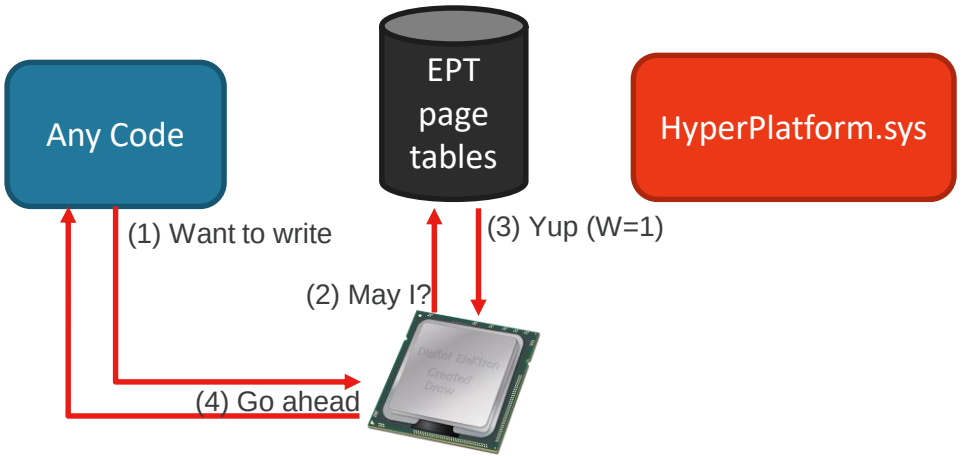


EXTENDED PAGE TABLES (EPT)



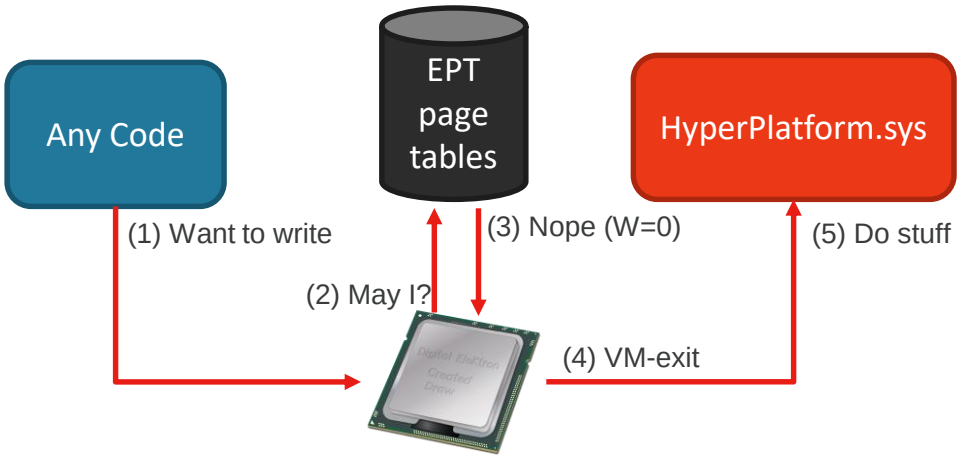
MEMORY ACCESS MONITORING

- EPT page tables can set additional permissions (RWX)



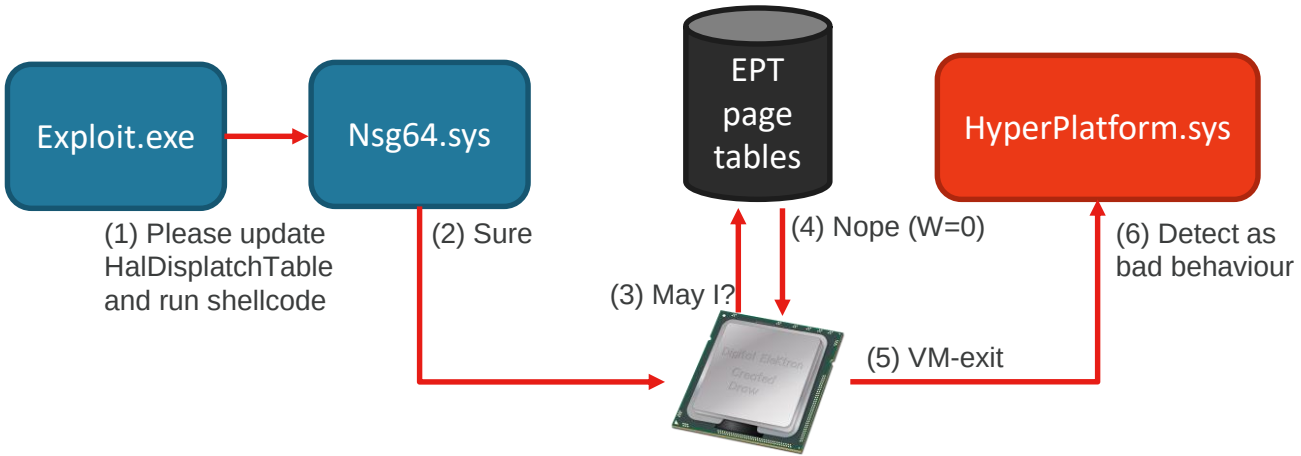
MEMORY ACCESS MONITORING

- EPT page tables can set additional permissions (RWX)



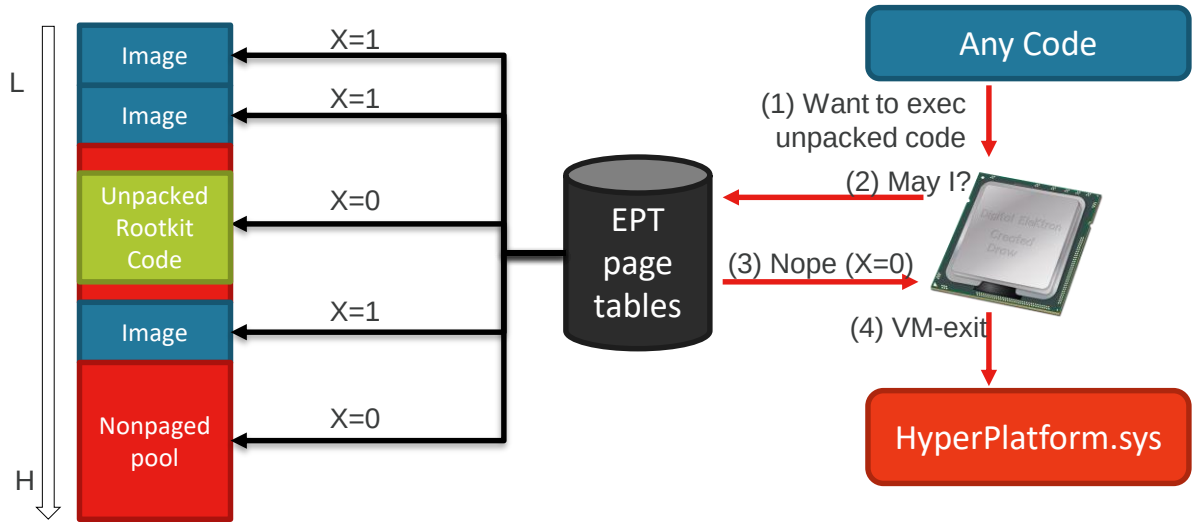
DEMO 2

- Detection of sensitive memory region modification for additional security
- nt!HalDispatchTable is an array of function pointers and should not be modified



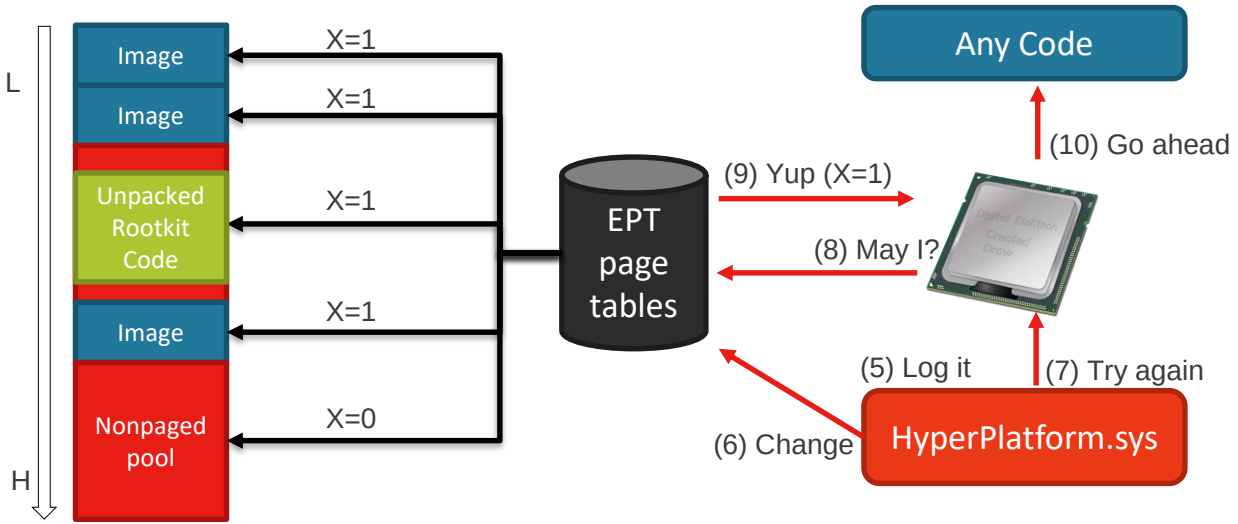
DEMO 3

- vs. Turla (Uroburos) rootkit



DEMO 3

- vs. Turla (Uroburos) rootkit

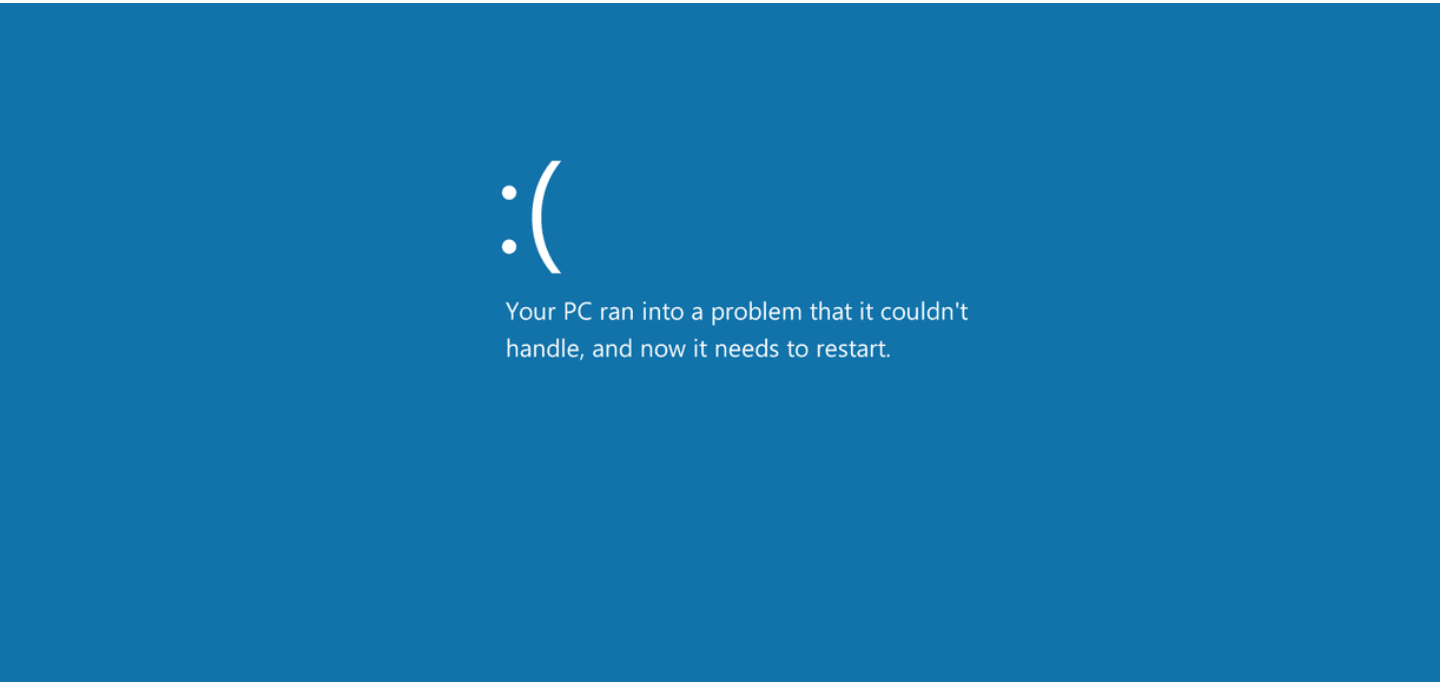


FURTHER LEARNING RESOURCES

- Simple
  - SimpleVisor
- Advanced (nested virtualization, multi-platform support et
  - KSM
  - Bareflank
  - kHypervisor

http://www.flickr.com/photos/backpackphotography/2318055128/sizes/m/in/photostream/
https://www.flickr.com/photos/luchoedu/2452449369

CROWDSTRIKE



ISSUE #1 – HOST/GUEST COMMUNICATION

- Hypervisor needs to communicate with the OS, but…
  …the exit handler does not run in a proper OS context!
- Can not use OS functions directly!
- Requires ISR and injecting vectored events
- => Must be a PnP driver
- But what if interrupts are disabled?
  - A) Use Interrupt Window Exiting (IWE)
  - B) Use posted events
- Watch out for Windows 10 interrupt steering

ISSUE #2 – USING THE FEATURES

- Example: preventing malicious drivers from disabling SMEP
  => easy!
- Prevent patching / execution of memory regions
  - Use EPTs to make pages RO / NX
  - Requires analysis of virtual address space (“Mirror page table”)
  - OS version dependent, KASLR, no driver unload callback
    => not trivial
  - Works for non-paged memory, but what about paged memory?
  - Maybe make pages either RO(/X) or RW/NX?
  - Paged out RO(/X) memory will be written to or zeroed
    => Make pages ~~great~~ RW/NX again! (If linear address is ok)
  - Great! Problem solved! Or ... maybe not?

ISSUE #3 – PATCH GUARD

- Present on all x64 Windows versions
- Only active when the system boots without a kernel-debugger!
- Highly obfuscated and self-decrypting code
- Runs regularly as DPC (DISPATCH_LEVEL) or in a worker thread (PASSIVE_LEVEL)
- Uses non-paged pool or system PTEs
- Self-decryption requires RWX access. This conflicts with the RO(-X) / RW-NX approach :(
  - Need to identify Patch Guard pages and make them RWX
  - These pages need regular cleanup or will "leak“
  - Are we sure, we get all Patch Guard pages and nothing else?

ISSUE #4 – HYPER-V

- Customers might want to run it!
  - Credential Guard
  - Device Guard
  - Hyper Guard
- Present by default in Windows 10 – one checkbox away
- Can we get below Hyper-V maybe?
  - Hyper-V is initialized too early (UEFI Boot Services)
  - => Requires UEFI boot module to get below it
  - === > Requires full nested virtualization support
  - === > Requires supporting full OS boot

CROWDSTRIKE

ISSUE #5 –PERFORMANCE

- How expensive are VM exits?
  - On a Haswell CPU about 500 cycles per exit/entry
  - For comparison: a system call is about 100 cycles
  - On VMWare: > 7000 cycles per exit/entry
- What impact does EPT have?
  - Strongly workload dependent
  - Can add 20% runtime overhead
- What is the overall impact of a hypervisor?
  - Strongly feature dependent
  - Can be < 1% or 50% and higher

CROWDSTRIKE

ISSUE #6 – OS/APP COMPATABILITY

- A hypervisor can introduce subtle timing issues
  - TSC offsetting vs RTC vs OS TSC synchronization
  - Might introduce new attack vectors (timing based memory layout analysis)
  - Detailed analysis by VMWare (“Timekeeping in VMware Virtual Machines”)

ISSUE #7 – FEATURE AVAILABILITY

- Do customers have the required hardware?
  - VT-x: since Pentium 4 (11/2005), AMD-v since Athlon 64 (5/2006)
  - EPT: since Nehalem, NP: since Opteron (9/2007)
  - EPT switching (VMFUNC): since Haswell (6/2013)
  - VMCS shadowing: since Haswell (2013)
  - MBEC: since Kaby Lake (2016)

CONCLUSION

- Developing a simple hypervisor is easier than ever
- Even a simple hypervisor can open many possibilities
- Learn more at GitHub with further examples
- Complexity grows with features & requirements
- Productization is not a trivial task

CROWDSTRIKE

QUESTIONS

CROWDSTRIKE

THANK YOU

Satoshi.Tanda@crowdstrike.com
Timo.Kreuzer@crowdstrike.com

REFERENCE

- Demo 1: Added code “cr4_mask.fields.smep = true;” in VmpSetupVmcs, and

```
Cr4 cr4_current = {UtilVmRead(VmcsField::kGuestCr4)};
Cr4 cr4_requested = {*register_used};
if (cr4_current.fields.smep == 1 && cr4_requested.fields.smep == 0) {
  HYPERPLATFORM_COMMON_DBG_BREAK();
  KeBugCheck(MANUALLY_INITIATED_CRASH);
}
```

  In VmmpHandleCrAccess.

- Demo 1: Exploit: https://github.com/tandasat/ExploitCapcom

REFERENCE

- Demo 2 Hypervisor: https://github.com/tandasat/MemoryMon/tree/rwe_bh
- Demo 2 Exploit: https://github.com/tandasat/CVE-2014-0816

REFERENCE

- Ddimon (Invisible API Hook) : https://github.com/tandasat/DdiMon
- SimpleVisor: https://github.com/ionescu007/SimpleVisor
- Bareflank: https://github.com/Bareflank
- Ksm: https://github.com/asamy/ksm
- kHypervisor: https://github.com/Kelvinhack/kHypervisor