[redacted]

Malware vs

Virtualization

The endless cat and mouse play

Aurélien Wailly

[redacted]

aurelien.wail.ly/publications/hip-2013-slides.html

[redacted]

Plan

[redacted]

Plan

Malwares today

Plan

Malwares today

Research environment

Plan

Malwares today

Research environment

Detection

Plan

Malwares today

Research environment

Detection

Reaction

Plan

Malwares today
Research environment
Detection
Reaction
Roadmap

Virtualization

Virtualization

Easy provisionning

Virtualization

Easy provisionning

Rollback

Virtualization

Easy provisionning

Rollback

Consolidation

Virtualization

Easy provisionning

Rollback

Consolidation

Resource control

Virtualization

Easy provisionning

Rollback

Consolidation

Resource control

Rarely used

Virtualization

Easy provisionning

Rollback

Consolidation

Resource control

Rarely used

Often the sign of an analysis

Virtualization usages

Bright side

Virtualization usages

Bright side

Easy sandbox VMWare player

Virtualization usages

Bright side

Easy sandbox VMWare player No traces

Virtualization usages

Bright side

Easy sandbox VMWare player No traces

Debug kernels

Virtualization usages

Bright side

Easy sandbox VMWare player No traces

Debug kernels Try other OSes

Virtualization usages

Bright side

Easy sandbox VMWare player No traces

Debug kernels Try other OSes

Virtualization usages

Bright side

Easy sandbox VMWare player No traces

Debug kernels Try other OSes

Virtualization usages

Bright side

Easy sandbox VMWare player No traces

Debug kernels Try other OSes

Dark Side

Virtualization usages

Bright side

Easy sandbox VMWare player No traces

Debug kernels Try other OSes

Dark Side

Intercept BluePilling see after

Virtualization usages

Bright side

Easy sandbox VMWare player No traces

Debug kernels Try other OSes

Dark Side

Intercept BluePilling see after Ultimate obfuscation

mainly for

testing

purposes!

Malware

Malware

Largely dissected Anubis, malwr, GFI

Malware

Largely dissected Anubis, malwr, GFI

Adaptable behavior

Malware

Largely dissected Anubis, malwr, GFI

Adaptable behavior

Alex recommends VM Detection

Conclusion

- The list of recommendations is not exhaustive. (e.g. memory usage versus disk usage, custom crypto, logging, debug, VM detection,...)

On the bright side

Malwares **may** detect virtualized environments

On the bright side

Malwares **may** detect virtualized environments

Adopt **clean** behavior

On the bright side

Malwares **may** detect virtualized environments

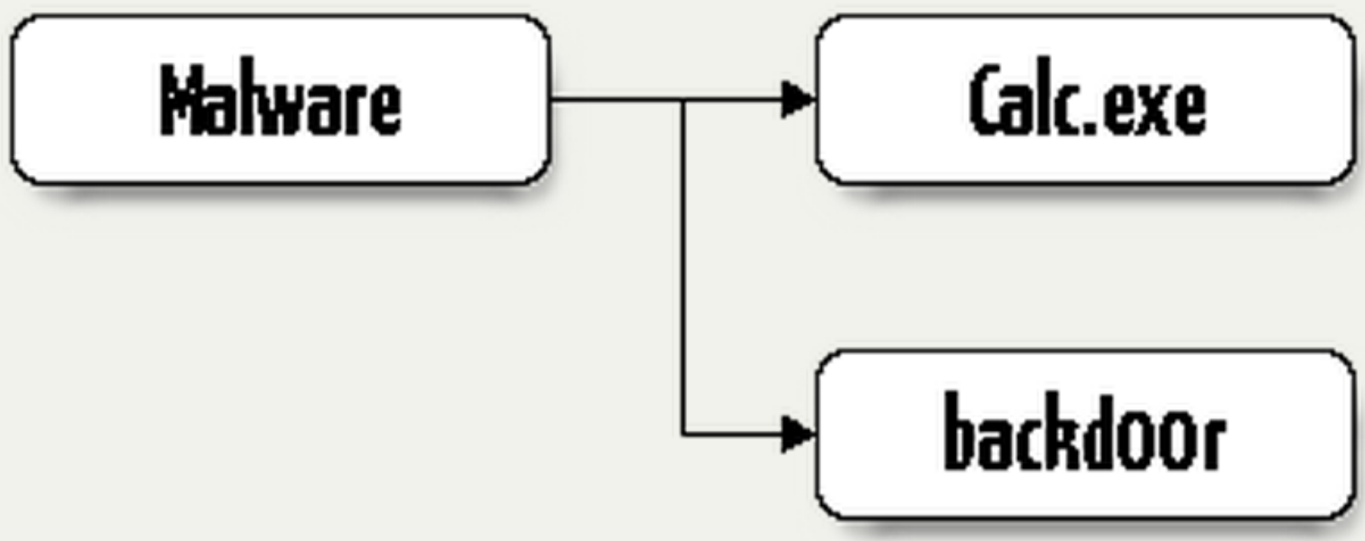
Adopt clean behavior

Targeted attacks

On the bright side

Malwares **may** detect virtualized environments

Adopt clean behavior
Targeted attacks



On the dark side

Games **may** detect virtualized environments

On the dark side

Games **may** detect virtualized environments

Cheat detection

On the dark side

Games **may** detect virtualized environments

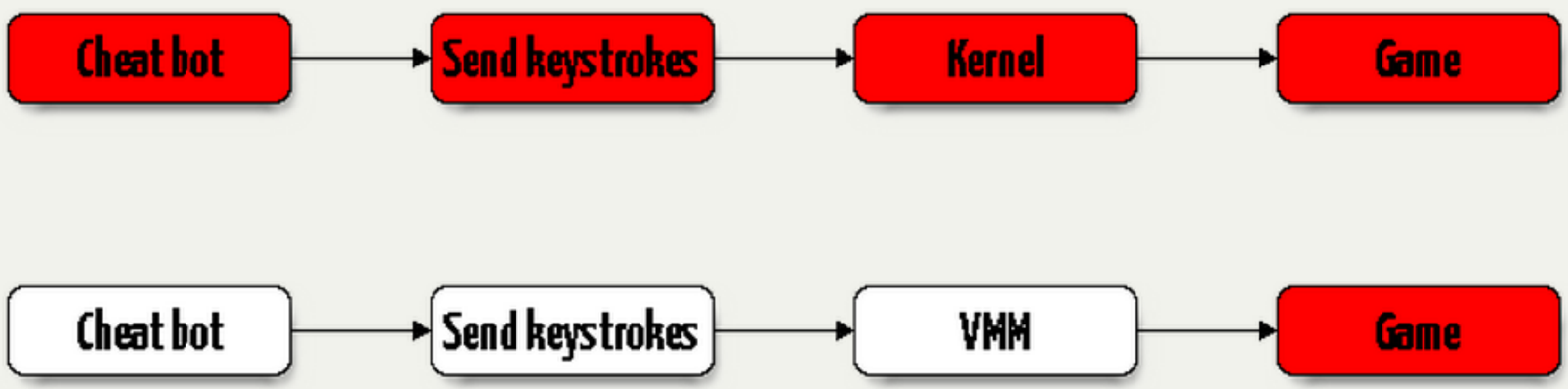
Cheat detection

Protect against fake hardware

On the dark side

Games **may** detect virtualized environments

Cheat detection
Protect against fake hardware



who is

leading?

how to detect virtualized environments ?

Is it easier to hide or to ?

how to detect virtualized environments ?

Is it easier to hide or to detect ?

how to detect virtualized environments ?

Targeted escape

Sandbox environments have to

Targeted escape

Sandbox environments have to

Extract executable actions

Targeted escape

Sandbox environments have to

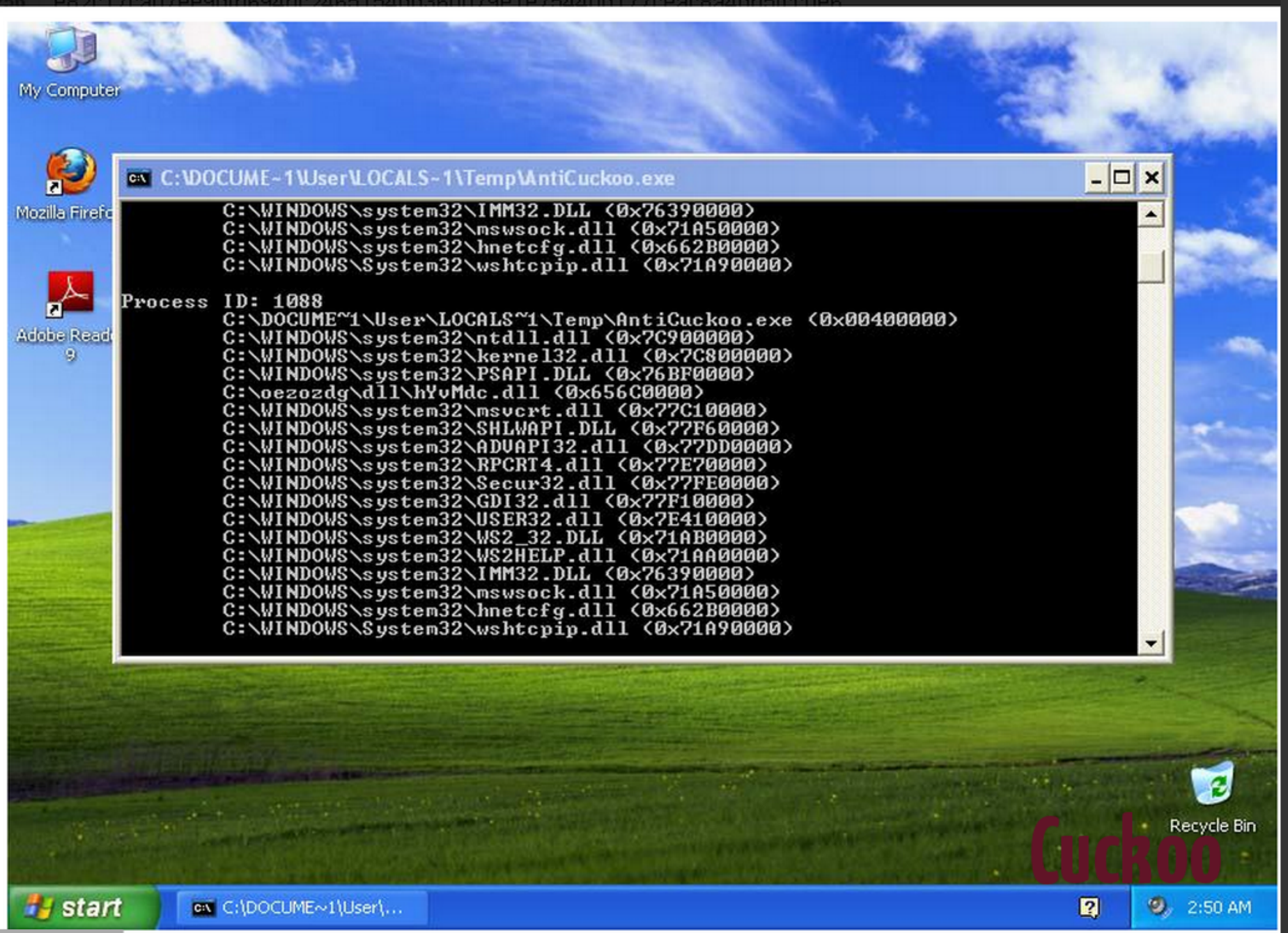
Extract executable actions

Communicate results

https://malwr.com/analysis/ZGM3MDI4MWQ0ZTk3NDA3MTk3NjkyMDQzZDMwZTE3YTE/

analyses Submit About

SHA1	fc7d455f3cc01c3e41b472ac508da92681ad9940
SHA256	e82c17ca07ee9dfb694bc2465154dd360079e1e75440d177ceac8a4db5fcc0e6
SHA	
CRC	
SSD	
YAR	



Sig

No si

Scr

Dor

No d

Sur

https://malwr.com/analysis/ZGM3MDI4MWQ0ZTk3NDA3MTk3NjkyMDQzZDMwZTE3YTE/

alyses Submit About

SHA1 fc7d455f3cc01c3e41b472ac508da92681ad9940

SHA256 e82c17ca07ee9dfb694bc2465154dd360079e1e75440d177ceac8a4db5fcc0e6

SHA

CRC

SSD

YAR

Sig

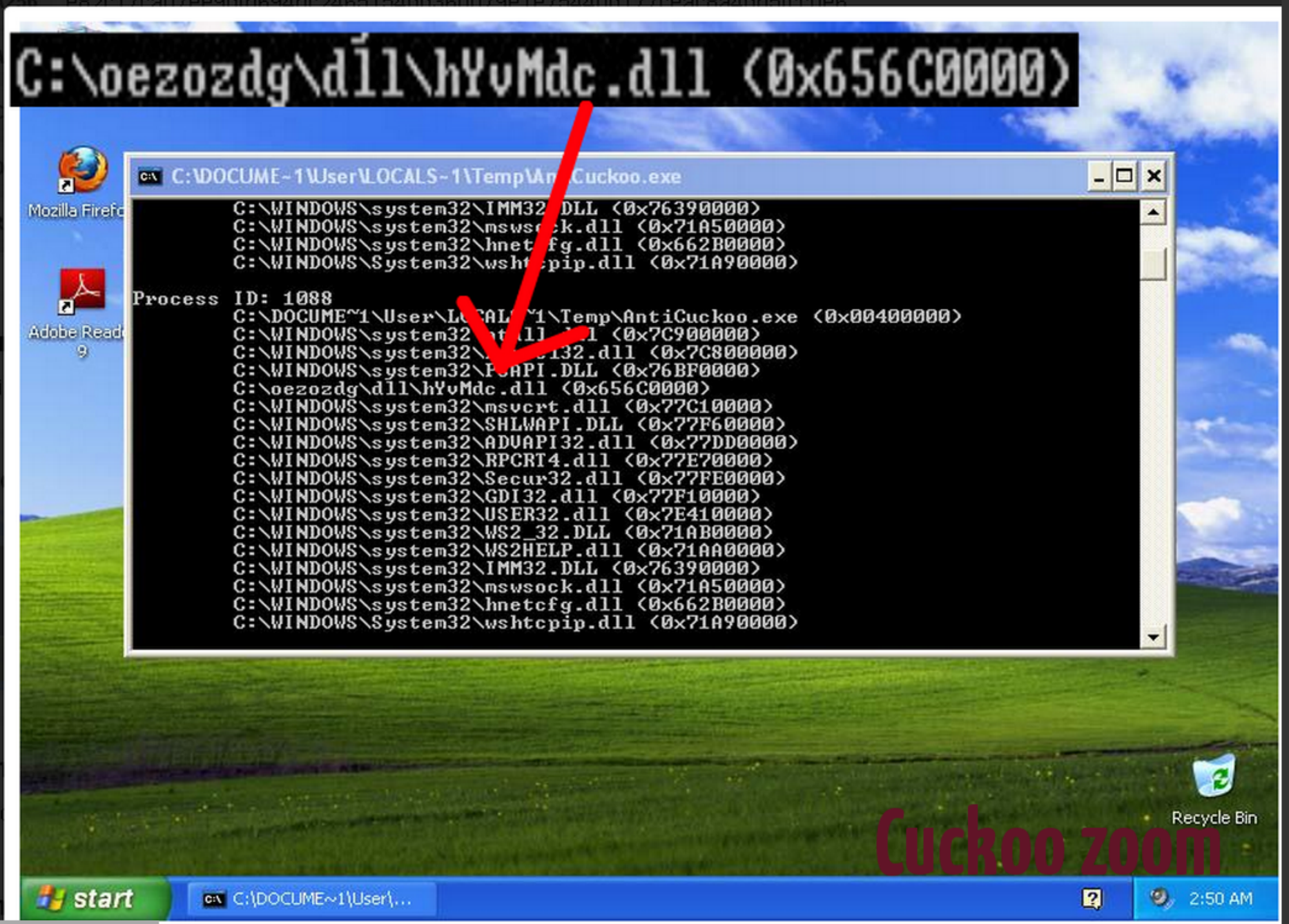
No si

Scr

Dor

No d

Sur




ARE YOU A
WIZARD
Meh :)
memegenerator.net

Going deeper

Going deeper

Virtualization

- CPU ⟶ vCPU
- Memory ⟶ Another MMU layer

Information tables

Each processor have its **own** IDT

CPU overview

Information tables

Interrupt Descriptor Table IDT

Each processor have its **own** IDT

CPU overview

Information tables

- Interrupt Descriptor Table IDT
- Local/Global Descriptor Table LDT/GDT

Each processor have its **own** IDT

Redpill

Table 1. Location differences

Physical	Virtual
0x80000000	0xc0000000

[2004, J.Rutkowska]

Redpill

Where to put vCPU's IDT ?

Table 1. Location differences

Physical	Virtual
0x80000000	0xc0000000

[2004, J.Rutkowska]

Processor features

CPU Informations

Processor features

CPU Informations

Processor brand string 0x80000002

Processor features

CPU Informations

Processor brand string 0x80000002

IsHypervisorPresent!

Virtualization overhead

VM Entry example

Virtualization overhead

VM Entry example

VM Entry / VM Exit cost

Virtualization overhead

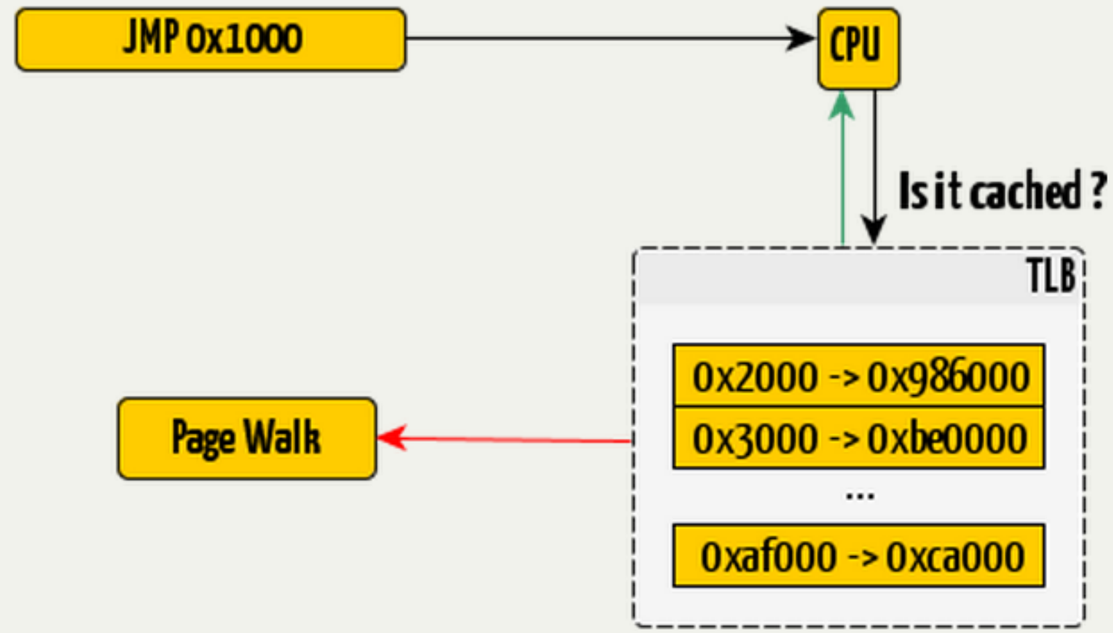
VM Entry example

VM Entry / VM Exit cost Measurements

Translation Lookaside Buffer

TLB Illustrated

Page Walking is expensive



Translation Lookaside Buffer

Virtual memory has pecularities

How to test **VMM presence** ?

Translation Lookaside Buffer

Virtual memory has pecularities

- Flush TLB while VMEXIT

How to test **VMM presence** ?

Translation Lookaside Buffer

Virtual memory has pecularities

- Flush TLB while VMEXIT

How to test VMM presence ?

- Fill TLB ⇒ VM Exit

Translation Lookaside Buffer

Virtual memory has pecularities

- Flush TLB while VMEXIT

How to test VMM presence ?

- Fill TLB ⟹ VM Exit
- Modification of at least one TLB entry

Translation Lookaside Buffer

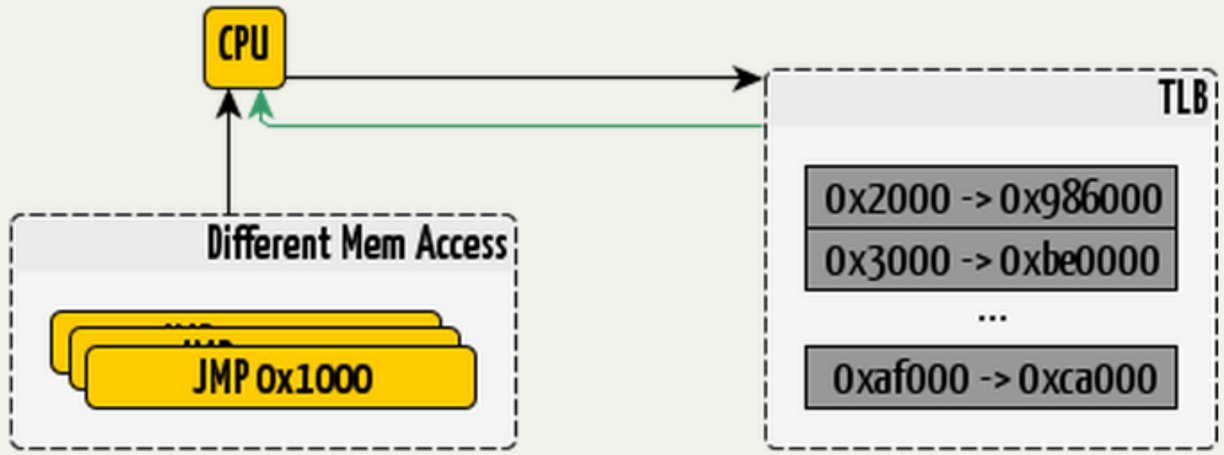
Virtual memory has pecularities

- Flush TLB while VMEXIT

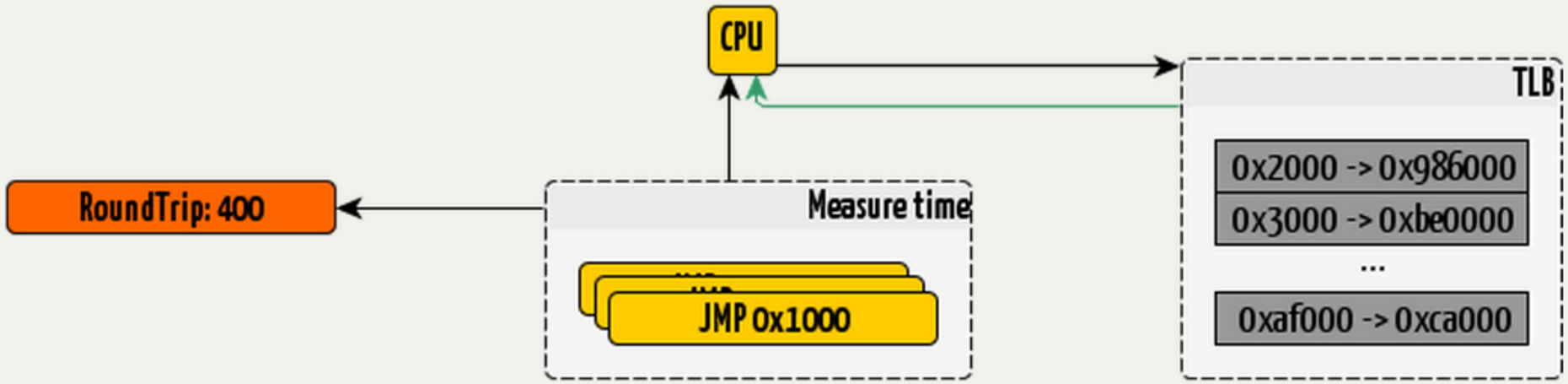
How to test VMM presence ?

- Fill TLB ⟹ VM Exit
- Modification of at least one TLB entry
- Compare access times

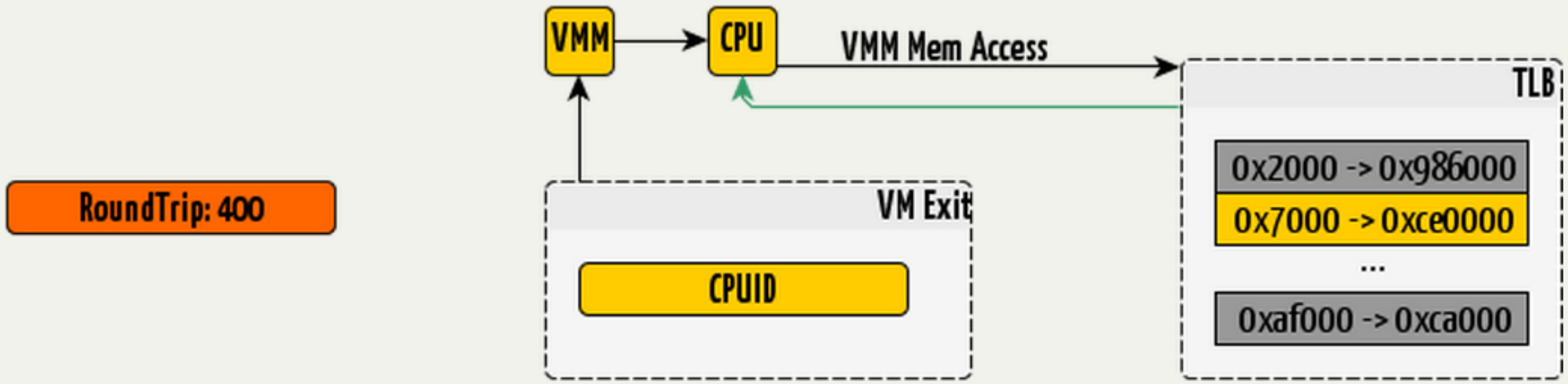
TLB detection



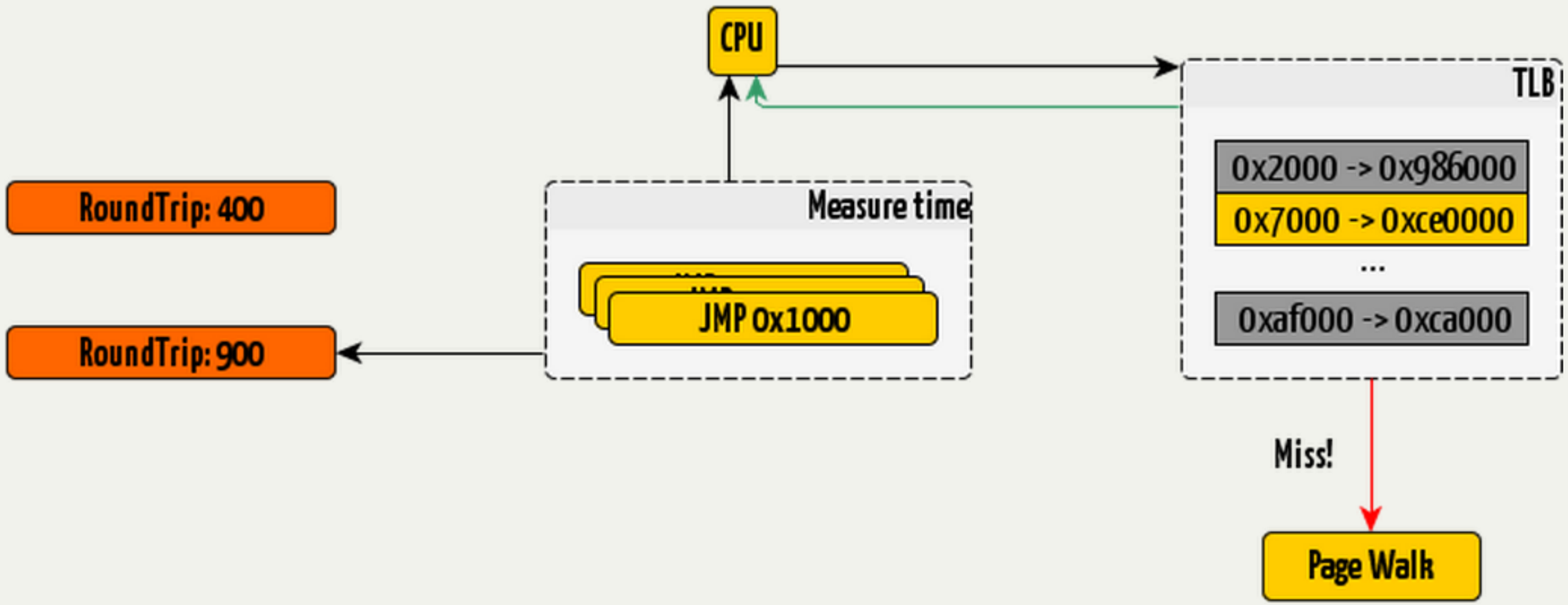
TLB detection



TLB detection



TLB detection



how to benchmark

Processor facilities

Integrated instructions

[http://download.intel.com/embedded/software/IA/324264.pdf]

Processor facilities

Integrated instructions

Time Stamp Counter: RDTSC, RDTSCP

[http://download.intel.com/embedded/software/IA/324264.pdf]

Processor facilities

Integrated instructions

Time Stamp Counter: RDTSC, RDTSCP

Real-Time Clock: ioctl(/dev/rtc0)

[http://download.intel.com/embedded/software/IA/324264.pdf]

Processor facilities

Integrated instructions

Time Stamp Counter: RDTSC, RDTSCP

Real-Time Clock: ioctl(/dev/rtc0)

Periodic Interrupt Timer (PIT)

[http://download.intel.com/embedded/software/IA/324264.pdf]

High precision Timers

Higher frequecy

64b resolution

External Timers

Rely on external protocol

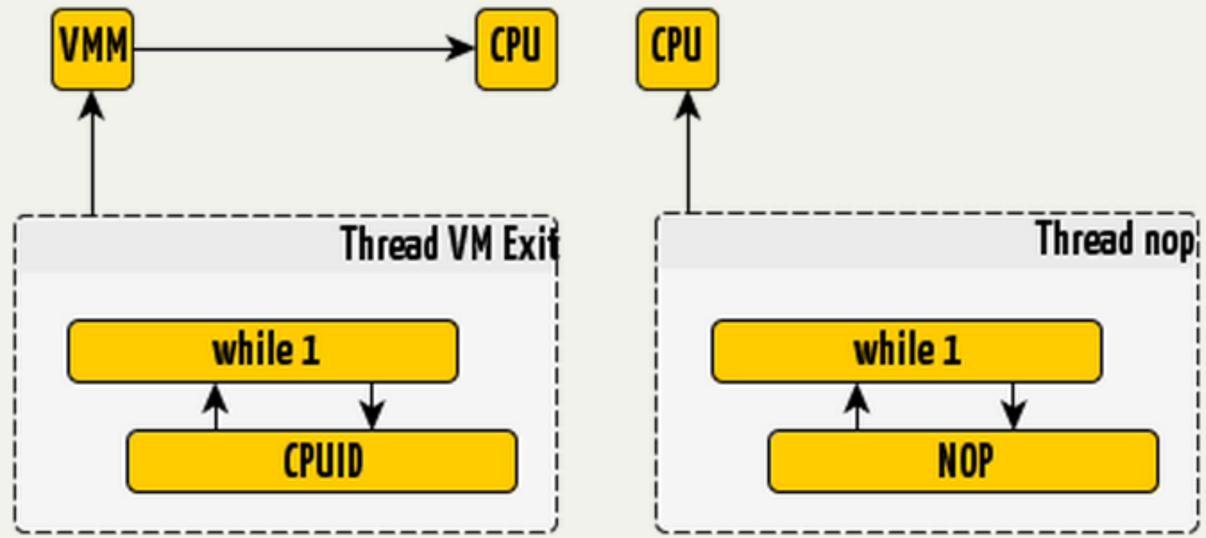
External Timers

Rely on external protocol

NTP/SNTP: Not precise

No reference

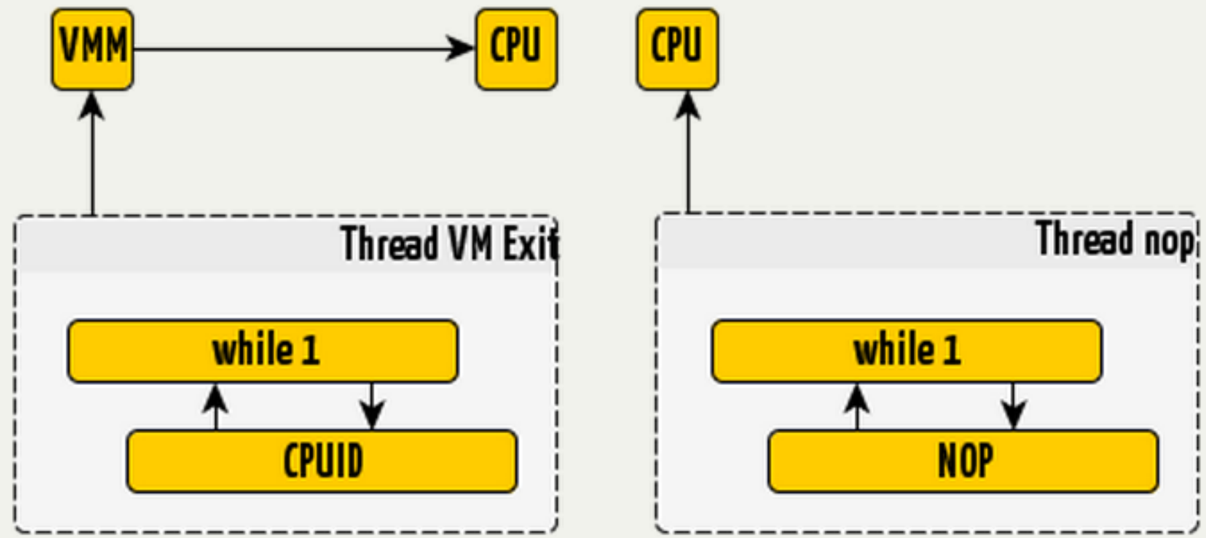
Ratio



No reference

Ratio

Compare counters



Discrepancies

Processors does not produce expected behavior

Discrepancies

Processors does not produce expected behavior

Wrong emulation foof bug, smsw

Discrepancies

Processors does not produce expected behavior

Wrong emulation foof bug, smsw

Specific hypercalls, accelerated graphic/drag and drop Peter Ferrie

Discrepancies

Processors does not produce expected behavior

Wrong emulation foof bug, smsw

Specific hypercalls, accelerated graphic/drag and drop Peter Ferrie

Typical attack: Oversized instruction more than 15B

Integrated facilities

Démo

Integrated facilities

Démo

WIN

BAREMETAL

```
dad@gambas ~/Projets/DetectHypervisor % ./detect2
000000: 50 65 6e 74                                      Pent
000000: 69 75 6d 28                                      ium(
000000: 52 29 20 44                                      R) D
000000: 75 61 6c 2d                                      ual-
[+] IDT base: 819da000
[+] SIDT[5] : 0x81
[+] SIDT[5] : 0x81
cpuid 1 ecx: 0c00e3bd bit:0
MSW: 8005003b
Ratio: 207.865799
```

VWARE PLAYER

```
dad@debian:~$ ./detect2
000000: 50 65 6e 74                                      Pent
000000: 69 75 6d 28                                      ium(
000000: 52 29 20 44                                      R) D
000000: 75 61 6c 2d                                      ual-
[+] IDT base: 8172d000
[+] SIDT[5] : 0x81
[+] SIDT[5] : 0x81
cpuid 1 ecx: 8c202201 bit:1
MSW: 8005003b
Ratio: 1588.099243
```

ESXI

```
ubuntu@ubuntu:~$ ./detect2
000000: 49 6e 74 65                                      Inte
000000: 6c 28 52 29                                      l(R)
000000: 20 58 65 6f                                       Xeo
000000: 6e 28 52 29                                      n(R)
[+] IDT base: 81dd9000
[+] SIDT[5] : 0x81
[+] SIDT[5] : 0x81
cpuid 1 ecx: 82982203 bit:1
MSW: 8005003b
Ratio: 615.849609
```

Thanks Pascal!

QEMU

```
root@debian:~# ./detect2
000000: 51 45 4d 55                                      QEMU
000000: 20 56 69 72                                       Vir
000000: 74 75 61 6c                                      tual
000000: 20 43 50 55                                       CPU
[+] IDT base: 8172d000
[+] SIDT[5] : 0x81
[+] SIDT[5] : 0x81
cpuid 1 ecx: 80802001 bit:1
MSW: cccc003b
Ratio: 3.352355
```

KVM

```
root@debby:~# ./detect2
000000: 51 45 4d 55                                      QEMU
000000: 20 56 69 72                                       Vir
000000: 74 75 61 6c                                      tual
000000: 20 43 50 55                                       CPU
[+] IDT base: 81738000
[+] SIDT[5] : 0x81
[+] SIDT[5] : 0x81
cpuid 1 ecx: 80802001 bit:1
MSW: 8005003b
Ratio: 901.177551
```

XEN

```
root@Xenny:~# ./detect2
000000: 49 6e 74 65                                      Inte
000000: 6c 28 52 29                                      l(R)
000000: 20 58 65 6f                                       Xeo
000000: 6e 28 52 29                                      n(R)
[+] IDT base: 8172d000
[+] SIDT[5] : 0x81
[+] SIDT[5] : 0x81
cpuid 1 ecx: 81b82221 bit:1
MSW: 8005003b
Ratio: 681.817383
```

Thanks Alex!

p. ferrie: status in progress

Formalization

VM detection categories

Formalization

VM detection categories

Logical discrepancies

Unexpected CPU behavior

Formalization

VM detection categories

Logical discrepancies

- Unexpected CPU behavior

Resources discrepancies

- Qemu hard drive
- "VMware" in windows registry

Formalization

VM detection categories

- **Logical** discrepancies
  - Unexpected CPU behavior
- **Resources** discrepancies
  - Qemu hard drive
  - "VMware" in windows registry
- **Timing** discrepancies
  - VMM overhead

How to find discrepancies

[T.Raffetserder]

NO.	D0	E0	Plans	ERRATA
Q1	X	X	NoFix	Transaction is not retried after BINIT#
Q2	X	X	NoFix	Invalid opcode 0FFFH requires a ModRM byte
Q3	X	X	NoFix	Processor may hang due to Speculative Page Walks to Non-Existent System Memory
Q4	X	X	NoFix	Memory type of the load lock different from its corresponding store unlock
Q5	X	X	NoFix	Machine check architecture error reporting and recovery may not work as expected
Q6	X	X	NoFix	Debug mechanisms may not function as expected
Q7	X	X	NoFix	Cascading of performance counters does not work correctly when forced overflow is enabled
Q8	X	X	NoFix	EMON event counting of X87 loads may not work as expected
Q9	X	X	NoFix	System bus interrupt messages without data and which receive a HardFailure response may hang the processor
Q10	X	X	NoFix	The Processor Signals Page-Fault Exception (#PF) Instead of Alignment Check Exception (#AC) on an Unlocked CMPXCHG8B Instruction
Q11	X	X	NoFix	FSW may not be completely restored after page fault on FRSTOR or FLDENV instructions
Q12	X	X	NoFix	Processor Issues Inconsistent Transaction Size Attributes for Locked Operation
Q13	X	X	NoFix	When the processor is in the System Management Mode (SMM), Debug Registers may be fully writeable

How to find discrepancies

Random tests

[T.Raffetserder]

NO.	D0	E0	Plans	ERRATA
Q1	X	X	NoFix	Transaction is not retried after BINIT#
Q2	X	X	NoFix	Invalid opcode 0FFFH requires a ModRM byte
Q3	X	X	NoFix	Processor may hang due to Speculative Page Walks to Non-Existent System Memory
Q4	X	X	NoFix	Memory type of the load lock different from its corresponding store unlock
Q5	X	X	NoFix	Machine check architecture error reporting and recovery may not work as expected
Q6	X	X	NoFix	Debug mechanisms may not function as expected
Q7	X	X	NoFix	Cascading of performance counters does not work correctly when forced overflow is enabled
Q8	X	X	NoFix	EMON event counting of X87 loads may not work as expected
Q9	X	X	NoFix	System bus interrupt messages without data and which receive a HardFailure response may hang the processor
Q10	X	X	NoFix	The Processor Signals Page-Fault Exception (#PF) Instead of Alignment Check Exception (#AC) on an Unlocked CMPXCHG8B Instruction
Q11	X	X	NoFix	FSW may not be completely restored after page fault on FRSTOR or FLDENV instructions
Q12	X	X	NoFix	Processor Issues Inconsistent Transaction Size Attributes for Locked Operation
Q13	X	X	NoFix	When the processor is in the System Management Mode (SMM), Debug Registers may be fully writeable

How to find discrepancies

Random tests

Learn from won't fix (think tuesday patch) !

[T.Raffetserder]

NO.	D0	E0	Plans	ERRATA
Q1	X	X	NoFix	Transaction is not retried after BINIT#
Q2	X	X	NoFix	Invalid opcode 0FFFH requires a ModRM byte
Q3	X	X	NoFix	Processor may hang due to Speculative Page Walks to Non-Existent System Memory
Q4	X	X	NoFix	Memory type of the load lock different from its corresponding store unlock
Q5	X	X	NoFix	Machine check architecture error reporting and recovery may not work as expected
Q6	X	X	NoFix	Debug mechanisms may not function as expected
Q7	X	X	NoFix	Cascading of performance counters does not work correctly when forced overflow is enabled
Q8	X	X	NoFix	EMON event counting of X87 loads may not work as expected
Q9	X	X	NoFix	System bus interrupt messages without data and which receive a HardFailure response may hang the processor
Q10	X	X	NoFix	The Processor Signals Page-Fault Exception (#PF) Instead of Alignment Check Exception (#AC) on an Unlocked CMPXCHG8B Instruction
Q11	X	X	NoFix	FSW may not be completely restored after page fault on FRSTOR or FLDENV instructions
Q12	X	X	NoFix	Processor Issues Inconsistent Transaction Size Attributes for Locked Operation
Q13	X	X	NoFix	When the processor is in the System Management Mode (SMM), Debug Registers may be fully writeable

What about using network

[X.Chen, and M.Lindorfer]

What about using network

TCP RTT differs for virtualized environments

[X.Chen, and M.Lindorfer]

What about using network

TCP RTT differs for virtualized environments

Mimic virtualization to evade malware infection

[X.Chen, and M.Lindorfer]

What about using network

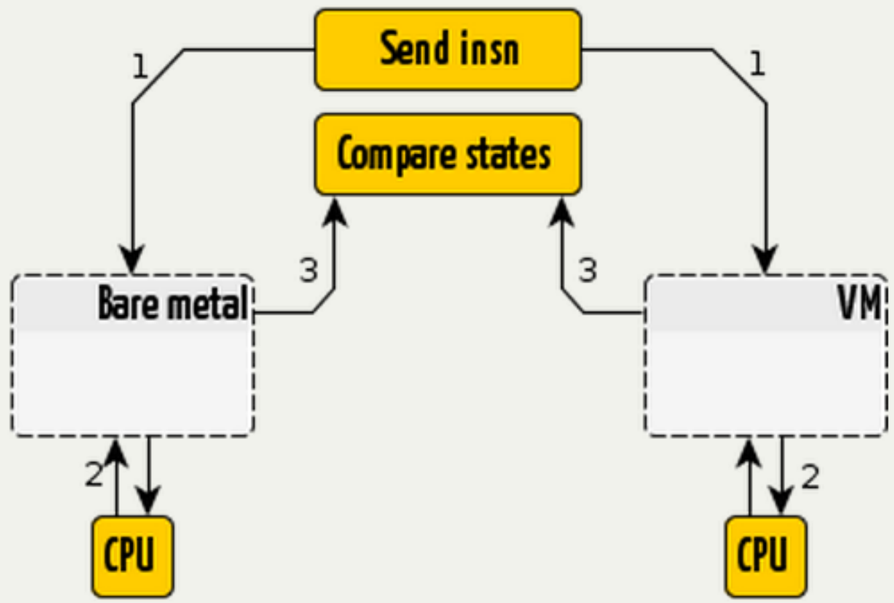
TCP RTT differs for virtualized environments

Mimic virtualization to evade malware infection

Taxonomy, extended by M. Lindorfer

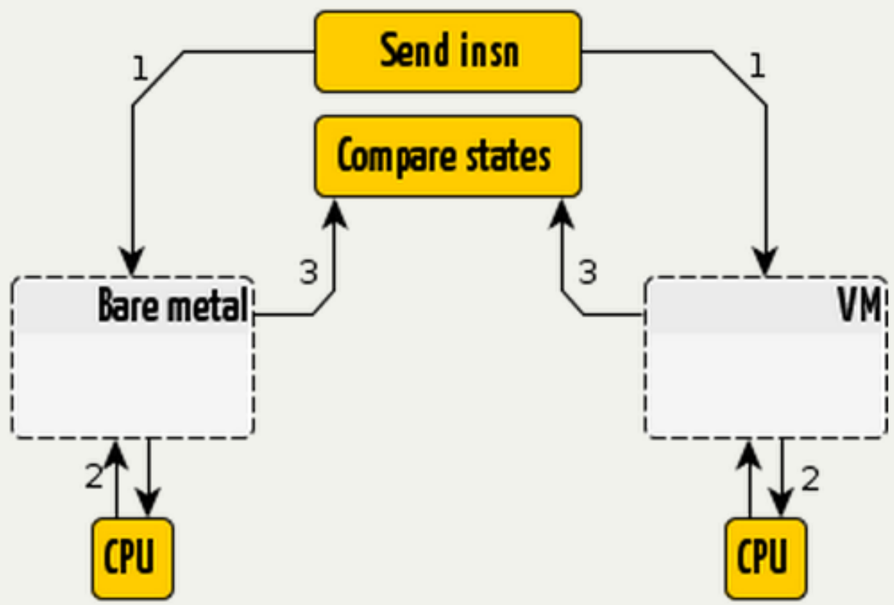
[X.Chen, and M.Lindorfer]

Automatization



Automatization

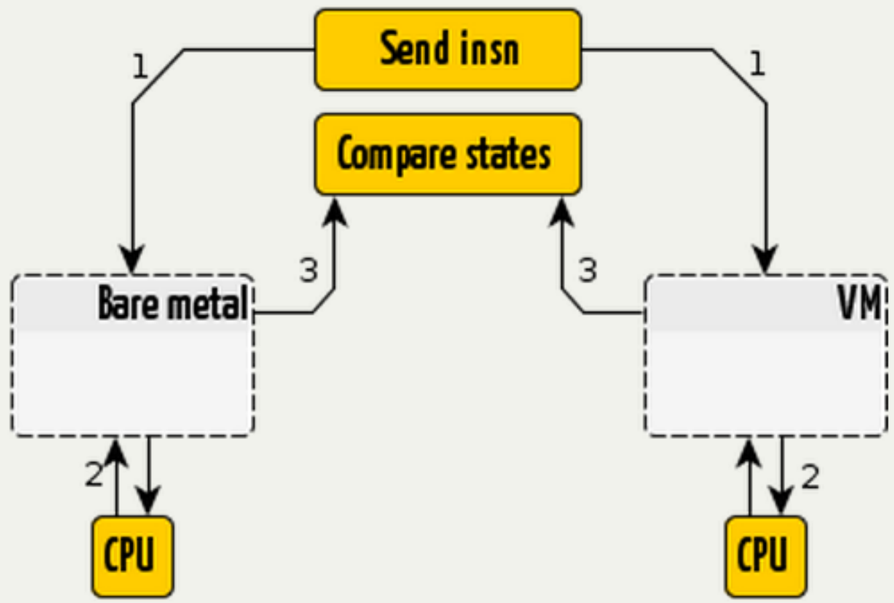
Fistful of redpills



Automatization

Fistful of redpills

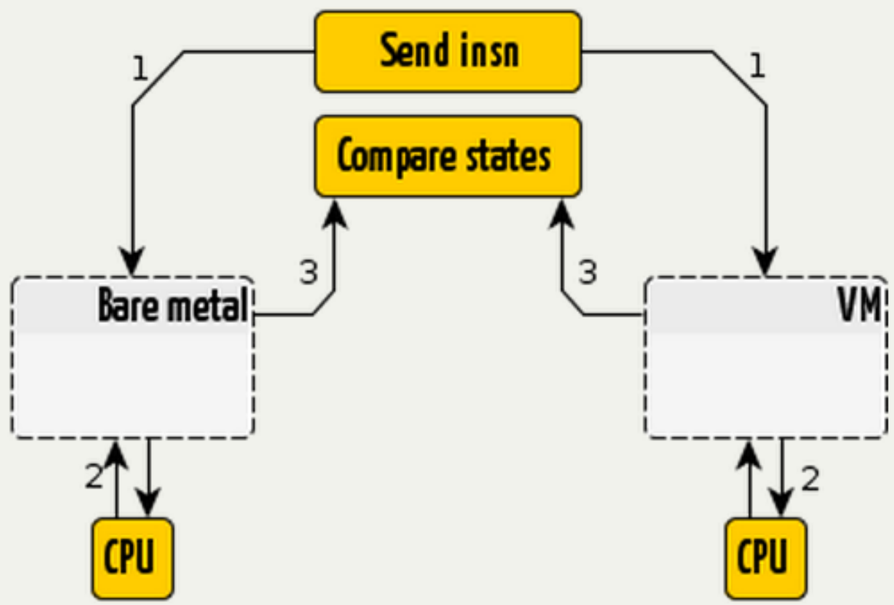
Create similar context and compare results



Automatization

Fistful of redpills
Create similar context and compare results

20 000 in few

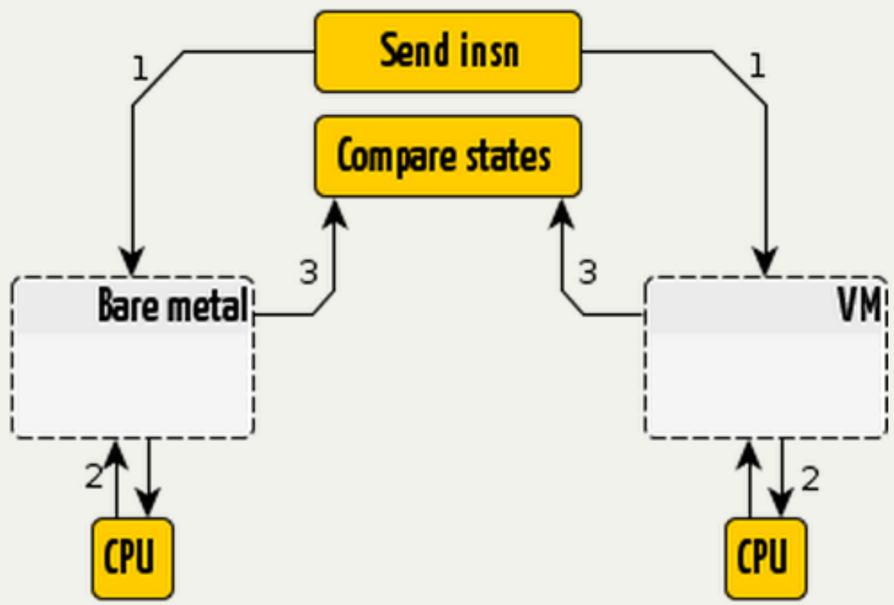


Automatization

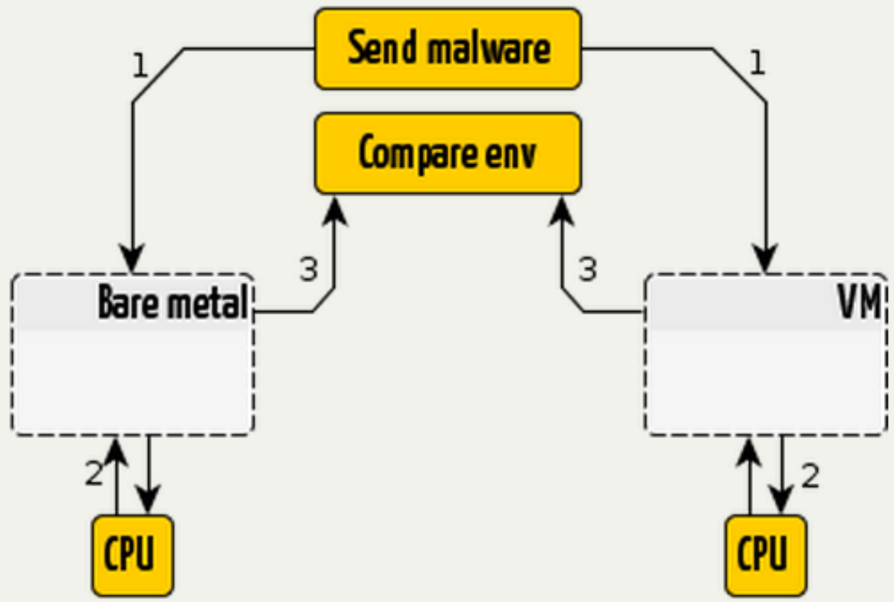
Fistful of redpills

Create similar context and compare results

20 000 in few hours

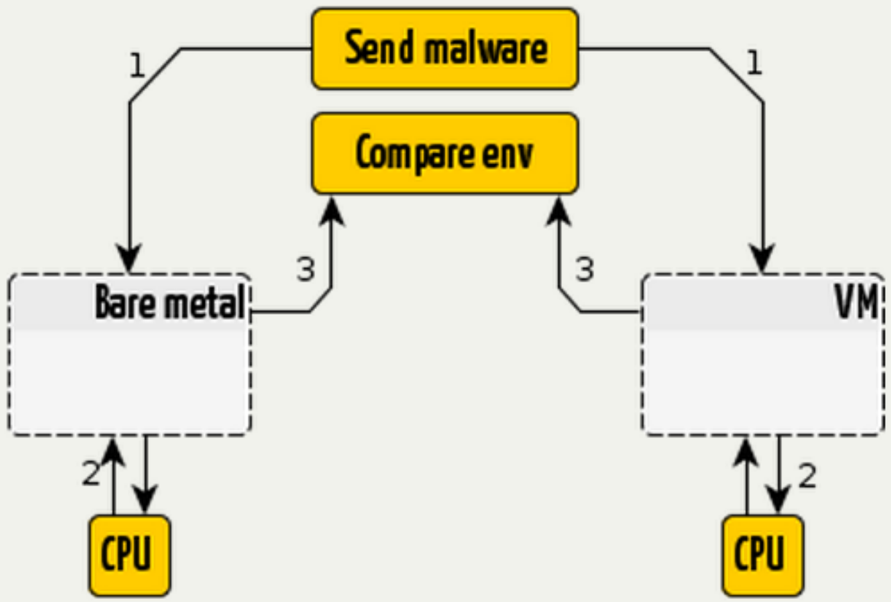


Are malwares using detection ?



Are malwares using detection ?

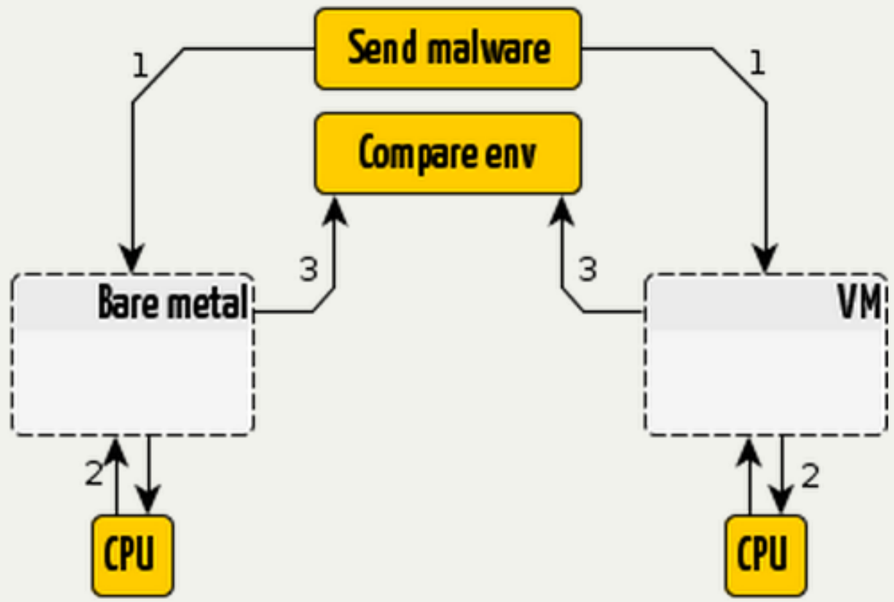
Tool to analyze malware behaviour through differential analysis



Are malwares using detection ?

Tool to analyze malware behaviour through differential analysis

Less than 2% tries to detect virtualization

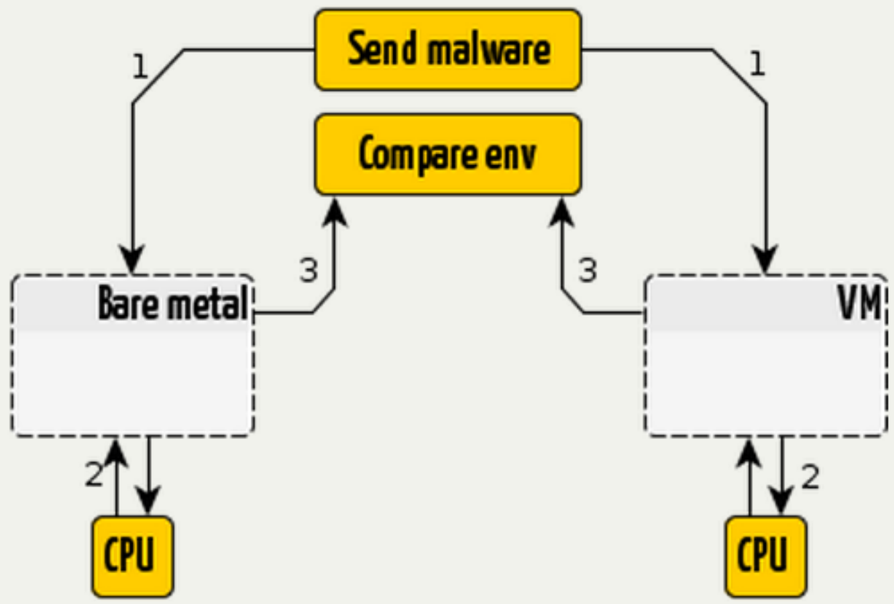


Are malwares using detection ?

Tool to analyze malware behaviour through differential analysis

Less than 2% tries to detect virtualization

Only used by worsts




Good catch, send it to me :)

hiding

Anti-detection

Anti-detection

Mechanisms formalization

Anti-detection

Mechanisms formalization

First architecture and implementation (Ether)

Anti-detection

Mechanisms formalization

First architecture and implementation (Ether)

Consider unexpected behaviors (foof)

Anti-detection

Mechanisms formalization

First architecture and implementation (Ether)

Consider unexpected behaviors (foof)

Setup several counter measures

- Modify CPU registers
- Use shadow page tables
- Prevent timing with TSC_OFFSET

Least virtualization

[A.Nguyen and E.Keller]

Least virtualization

To hinder virtualization effects, only vital components are virtualized

[A.Nguyen and E.Keller]

Least virtualization

To hinder virtualization effects, only vital components are virtualized

Mostly passthrough, few VMEXITs but only support 1 VM

[A.Nguyen and E.Keller]

Least virtualization

To hinder virtualization effects, only vital components are virtualized

Mostly passthrough, few VMEXITs but only support 1 VM

Reduce hypervisor footprint

[A.Nguyen and E.Keller]

Going deeper

Physical virtualization

Example: Barebox

Example: Barebox

Support snapshots

Example: Barebox

Support snapshots

Volatile memory with RAID mirroring

Example: Barebox

Support snapshots

Volatile memory with RAID mirroring

Save machine state (registers and interrupts) when OS boots

Example: Barebox

Support snapshots
Volatile memory with RAID mirroring
Save machine state (registers and interrupts) when OS boots
Allows quick rollback (reboot on clean machine < 4s)

Patching defects

MAVMM avoid TLB flushing with new VT-d

Awesome part: Free code

available! (!Ether)

New architectures

Virtualize without Hypervisors

New architectures

Virtualize without Hypervisors

Physicalization

New architectures

Virtualize without Hypervisors

Physicalization

Ether

New architectures

Virtualize without Hypervisors

Physicalization

Ether

NoHype

New architectures

Virtualize without Hypervisors

Physicalization

Ether

NoHype

DeHype

Protection

Slow CPUs

Protection

Slow CPUs

Need 25000 times slower to hide VMEXITs overhead

Behaviors

Calling **thousands** of VMEXITs is doubious

Behaviors

Calling **thousands** of VMEXITs is doubious

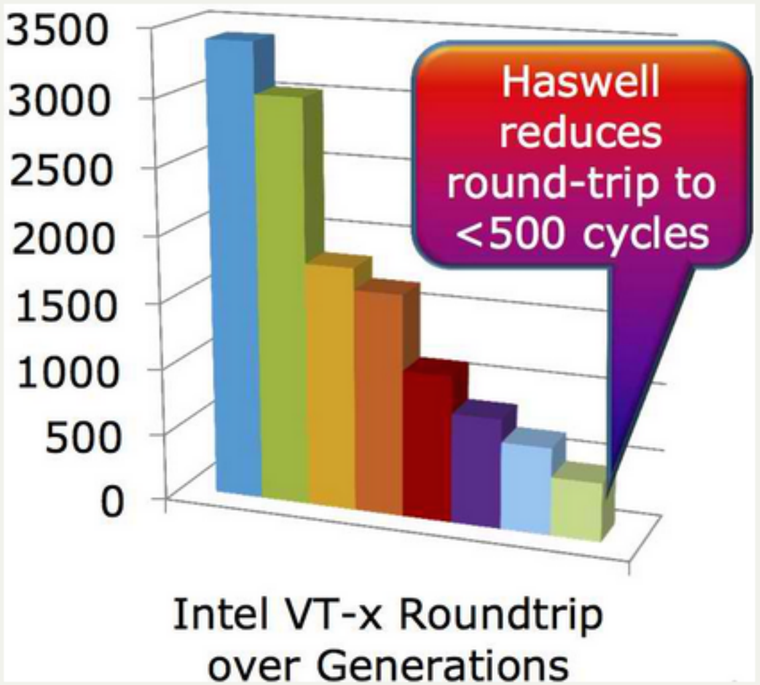
...Set up a threshold, and **hide VMM when hit**

Intel Haswell

Less VMEXITs

Intel Haswell

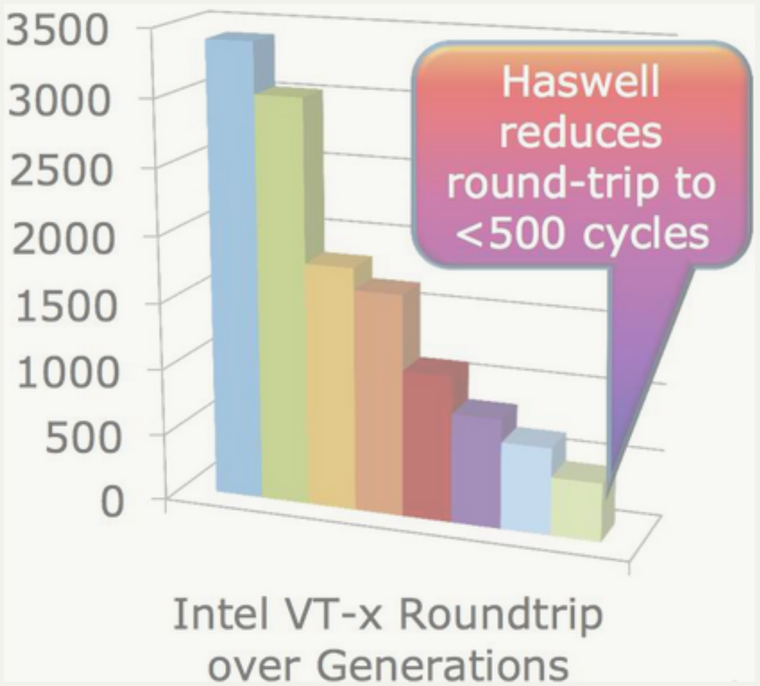
Less VMEXITs



www.pcworld.fr

Intel Haswell

Less VMEXITs



www.pcworld.fr

Reducing ratio and time-based attack reliance

Roadmap

toward **hypervisor** into the **CPU**

counter counter measure

Nether

PDF

Detect ether!

Nether

Formally correct

PDF

Detect ether!

Nether

Formally correct

Practically: weaknesses

PDF

Detect ether!

Conclusion

Conclusion

Should we need detection ?

Conclusion

Should we need detection ?

New challenges of physical hypervisor

merci!

http://aurelien.wail.ly