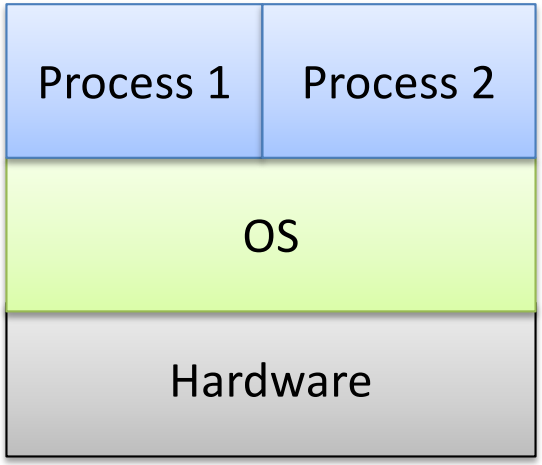
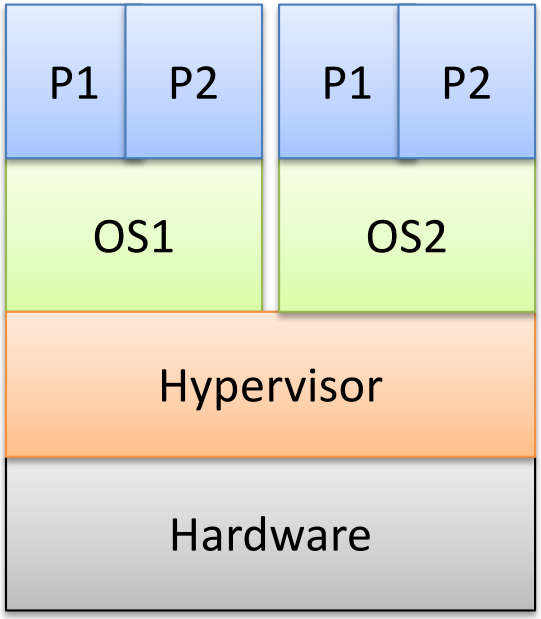
Virtualization

CS642: Computer Security

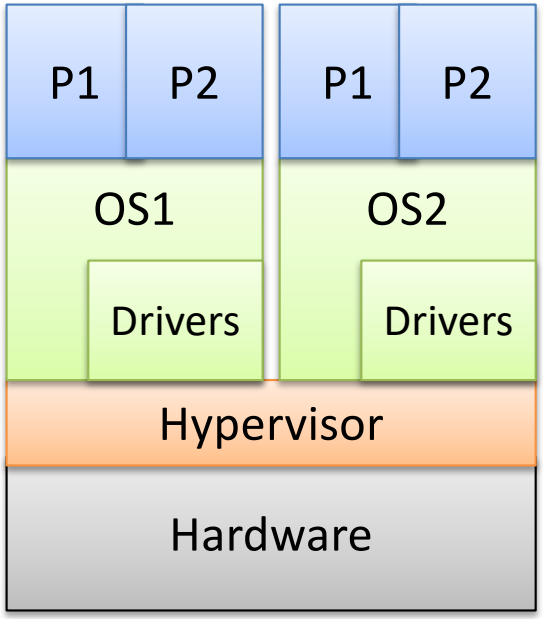
Virtualization



No virtualization



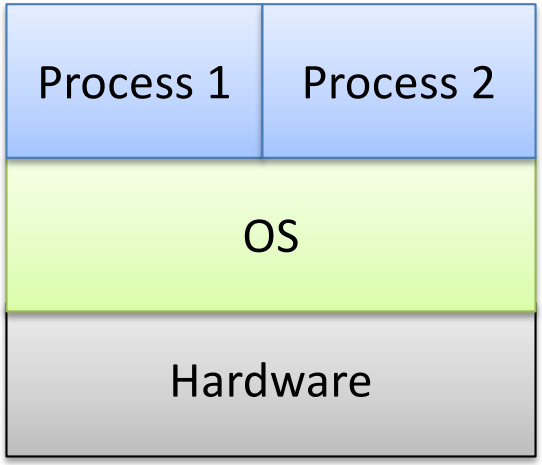
Full virtualization



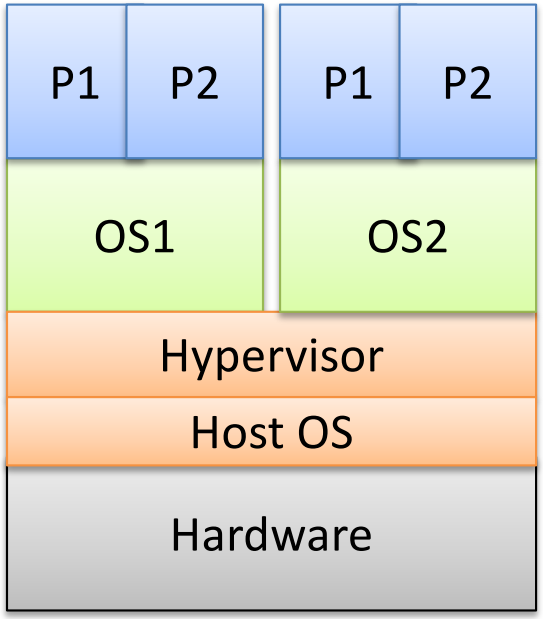
Paravirtualization

Type-1: Hypervisor runs directly on hardware

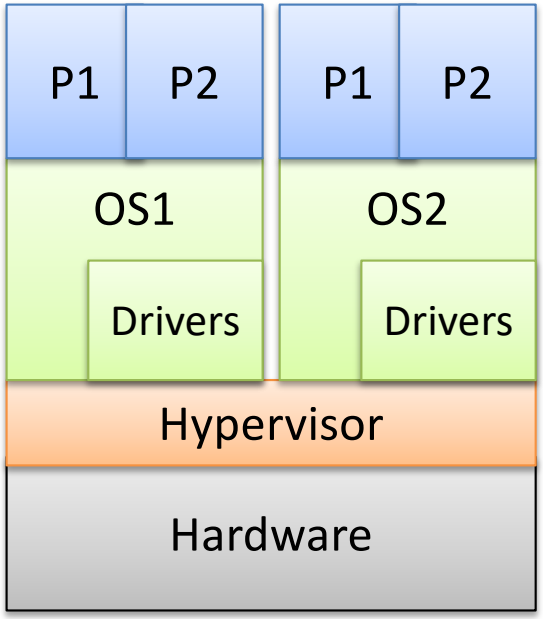
Virtualization



No virtualization



Full virtualization



Paravirtualization

Type-1: Hypervisor runs directly on hardware

Type-2: Hypervisor runs on host OS

IBM VM/370

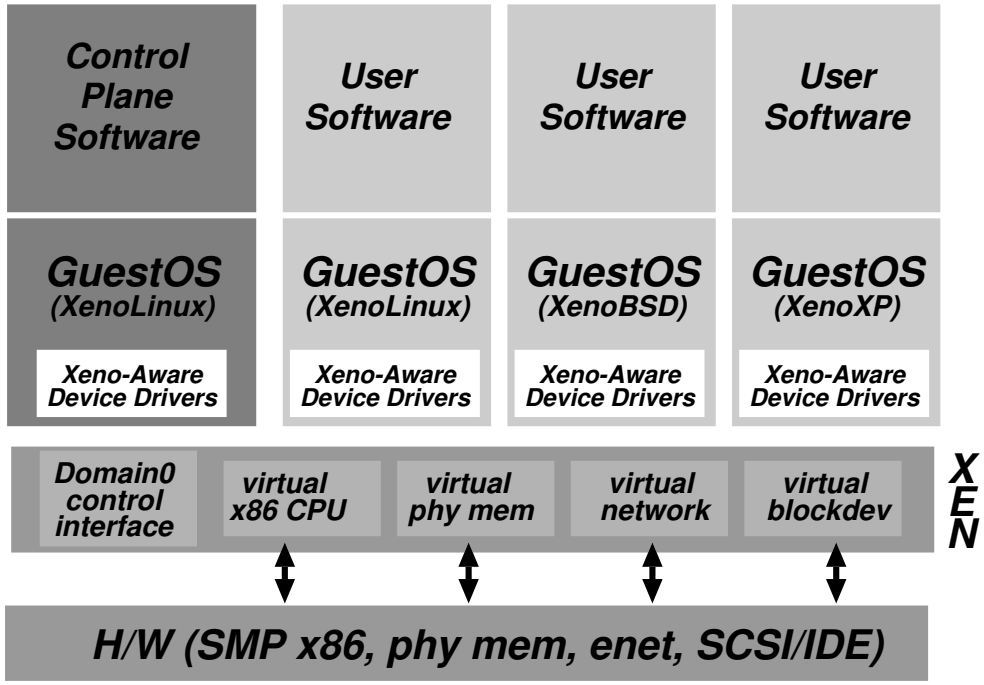
- Released in 1972
  - Used with System/370, System/390, zSeries mainframes
  - Full virtualization
- Supported CP/CMS operating system
  - Initial application was to support legacy OS
- z/VM is newer version, most recent version 2010
  - Better use of 64-bit mainframes

VMware Workstation - 1999

- Developed out of Stanford research project on fast simulation for computer architecture
- Solved how to virtualize x86
- Used for running multiple OS on a single desktop
  - Windows / Linux
  - Older versions of Windows
- Launched renaissance in virtualization

Xen

- 2003: academic paper
  - "Xen and the Art of Virtualization"
- Paravirtualization
  - Hypercalls vs system calls
  - Modified guest OS
  - Each guest given 1 or more VCPUs
- Why?

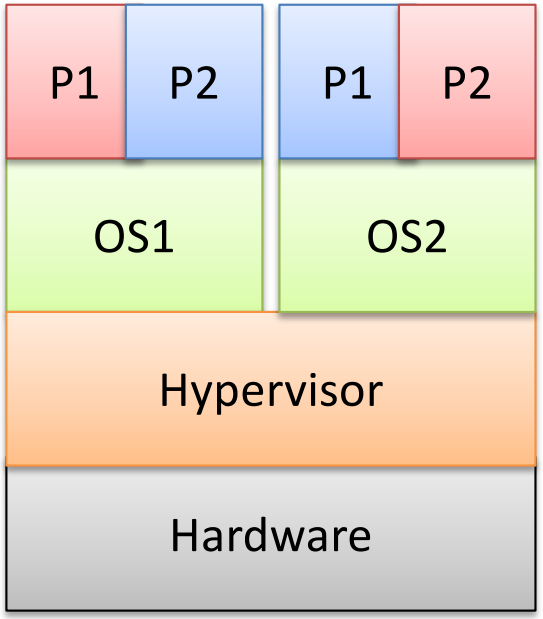


Example VM Use Cases

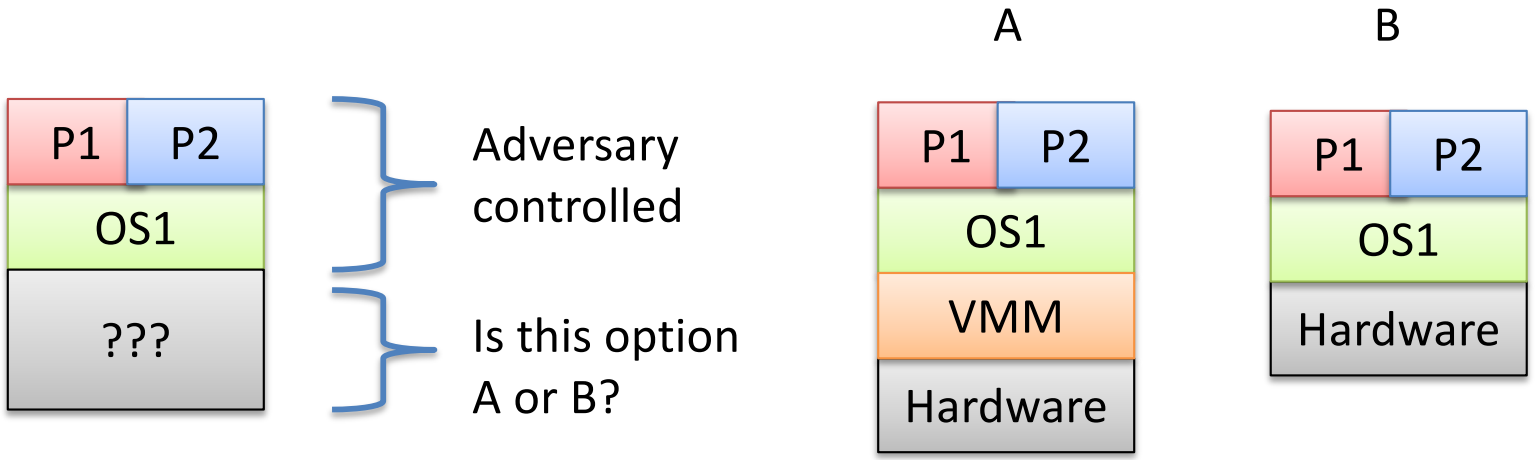
- Legacy support (e.g., VM/370)
- Development
- Server consolidation
- Cloud computing Infrastructure-as-a-Service
- Sandboxing / containment

Study of malware

- Researchers use VMs to study malware
  - Reduce harm
  - Introspection
- How would you evade analysis as a malware writer?
  - split personalities



VMM Transparency



- Adversary can detect if:
  - Paravirtualization
  - Logical discrepancies
    - Expected CPU behavior vs virtualized
    - Red pill (Store Interrupt Descriptor Table instr)
  - Timing discrepancies
    - Slower use of some resources

Garfinkel et al. "Compatibility is not transparency: VMM Detection Myths and Reality"

Detection of VMWare

```
MOV EAX,564D5868 <-- "VMXh"
MOV EBX,0
MOV ECX,0A
MOV EDX,5658 <-- "VX"
IN EAX,DX <-- Check for VMWare
CMP EBX,564D5868
```

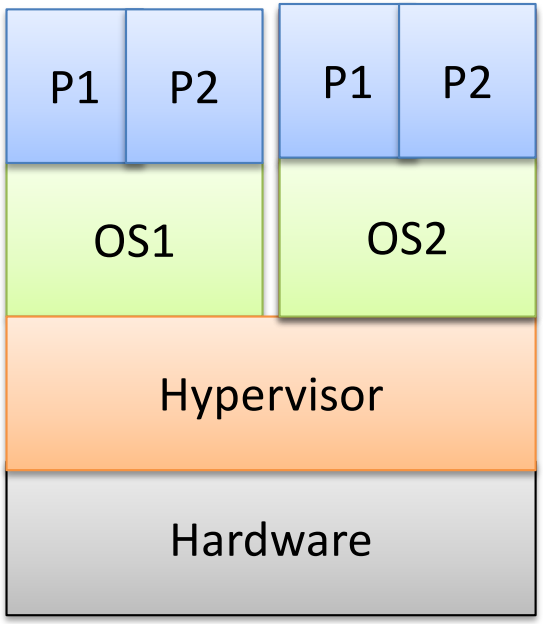
IN instruction used by VMWare to facilitate host-to-guest communication

VMWare:
    places VMXh in EBX
Physical:
    processor exception

From
http://handlers.sans.org/tliston/ThwartingVMDetection_Liston_Skoudis.pdf

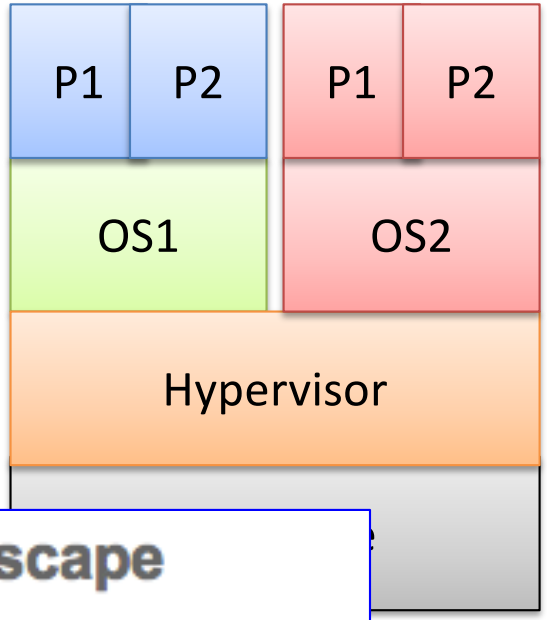
Server consolidation

- Consolidation
  - Use VMs to optimize use of hardware
  - Pack as many VMs onto each server as possible
  - Turn off other servers
- Threat model?
  - Containment
  - Isolation
  - Assume guests are/can be compromised



Violating containment

- Escape-from-VM
  - Vulnerability in VMM or host OS
  - Seemingly rare, but exist



VMware vulnerability allows users to escape virtual environment

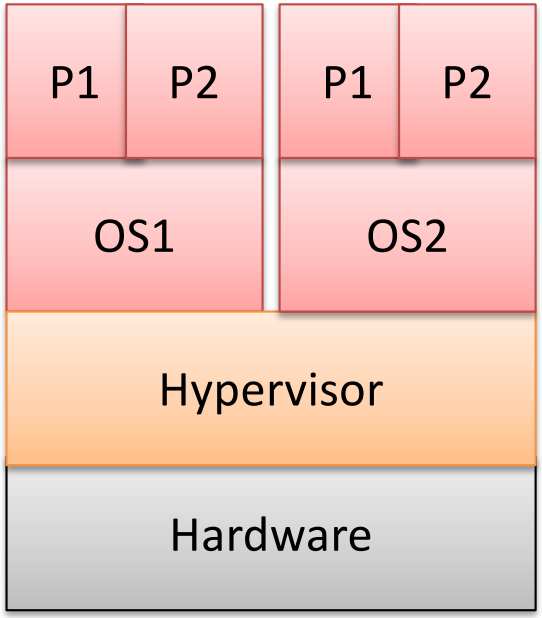
By Joab Jackson Feb 28, 2008

A new vulnerability found in some VMware products allows users to escape their virtual environments and muck about in the host operating system, penetration testing software firm Core Security Technologies announced earlier this week.

This vulnerability (CVE Name: CVE-2008-0923) could poise significant risks to enterprise users who are deploying VMware software as a secured environment.

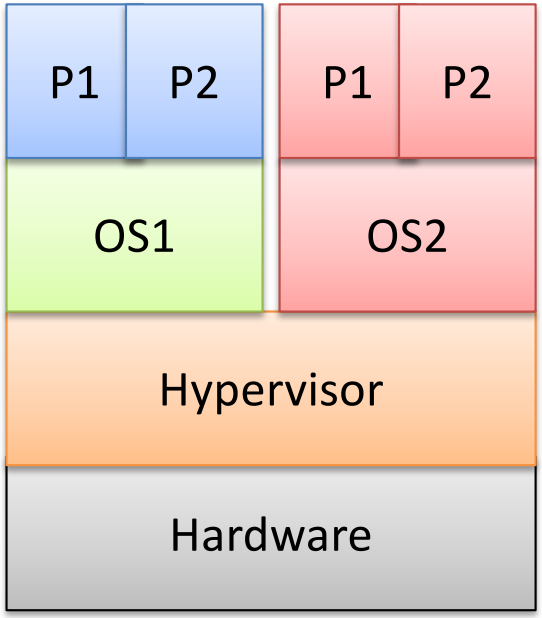
Violating isolation

- Covert channels between VMs circumvent access controls
  - Bugs in VMM
  - Side-effects of resource usage



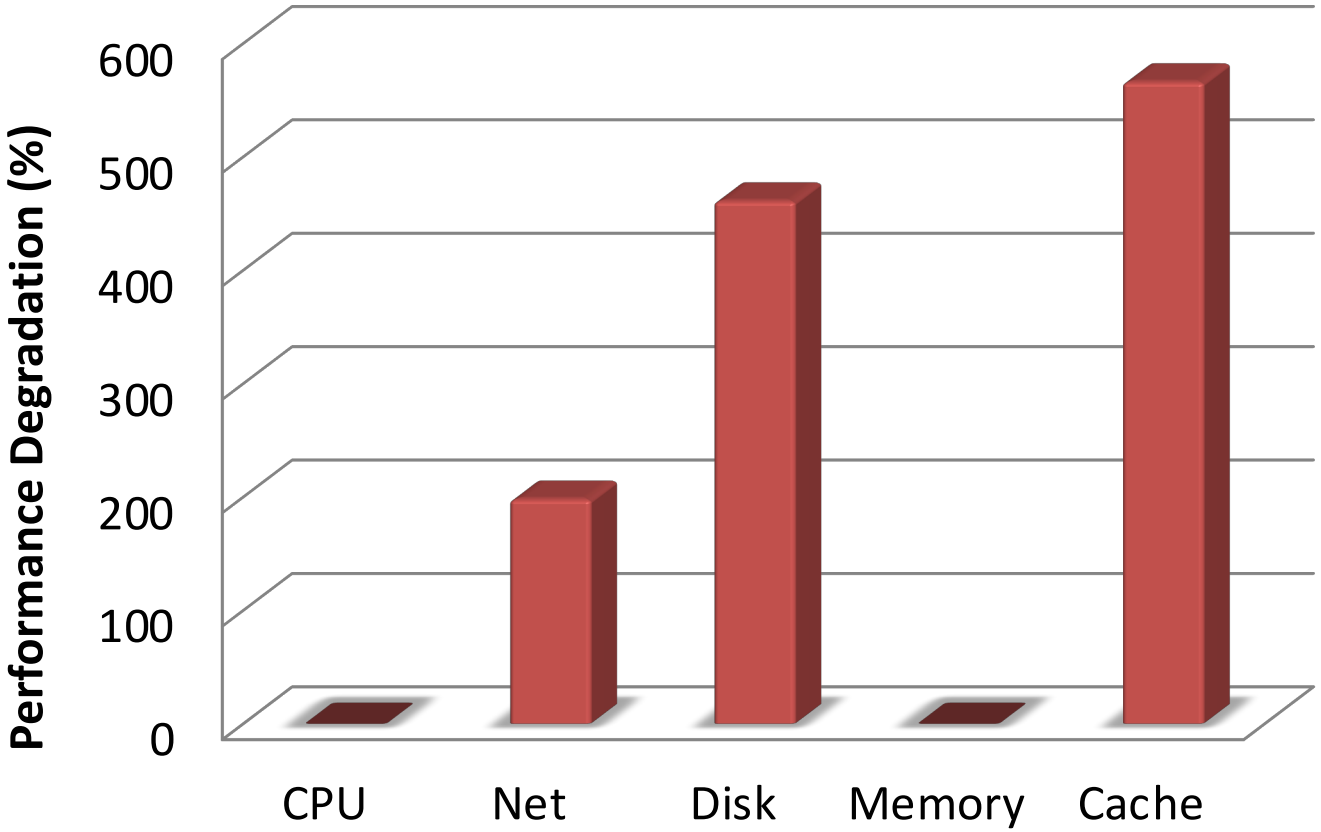
Violating isolation

- Covert channels between VMs circumvent access controls
  - Bugs in VMM
  - Side-effects of resource usage
- Degradation-of-Service attacks
  - Guests might maliciously contend for resources
  - Xen scheduler vulnerability



Measuring Resource Contention

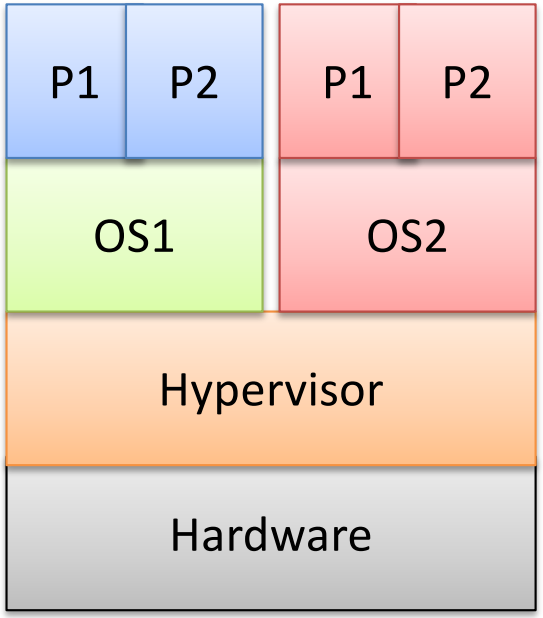
- Contention for the same resource



Local Xen Testbed	
Machine	Intel Xeon E5430, 2.66 Ghz
Packages	2, 2 cores per package
LLC Size	6MB per package

Violating isolation

- Covert channels between VMs circumvent access controls
  - Bugs in VMM
  - Side-effects of resource usage
- Degradation-of-Service attacks
  - Guests might maliciously contend for resources
  - Xen scheduler vulnerability
- Side channels
  - Spy on other guest via shared resources



Square-and-Multiply

/* $y = x^e \bmod N$, from **libgcrypt***/

Modular Exponentiation (x, e, N):

let $e_n \ldots e_1$ be the bits of e

$y \leftarrow 1$

for e_i in $\{e_n \ldots e_1\}$

$y \leftarrow$ **Square**(y) **(S)**

$y \leftarrow$ **Reduce**(y, N) **(R)**

if $e_i = 1$ then

$y \leftarrow$ **Multi**(y, x) **(M)**

$y \leftarrow$ **Reduce**(y, N) **(R)**

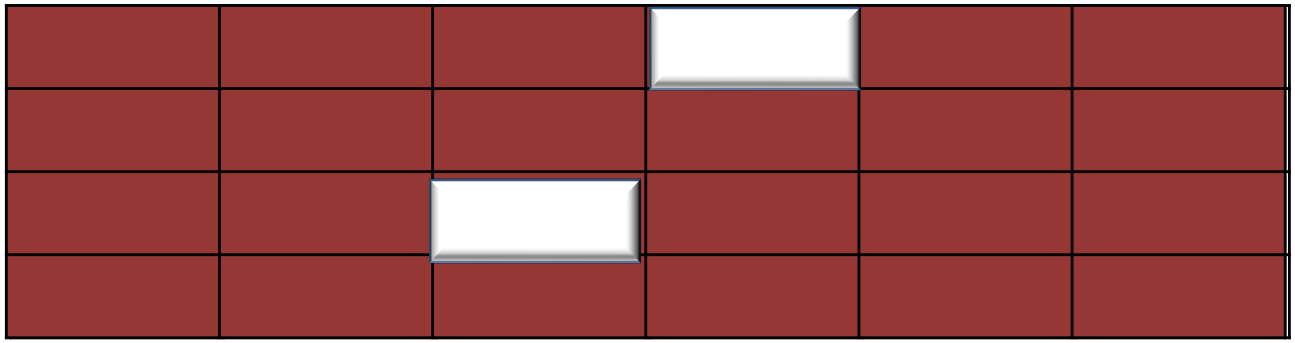
$e_i = 1 \rightarrow$ SRMR

$e_i = 0 \rightarrow$ SR

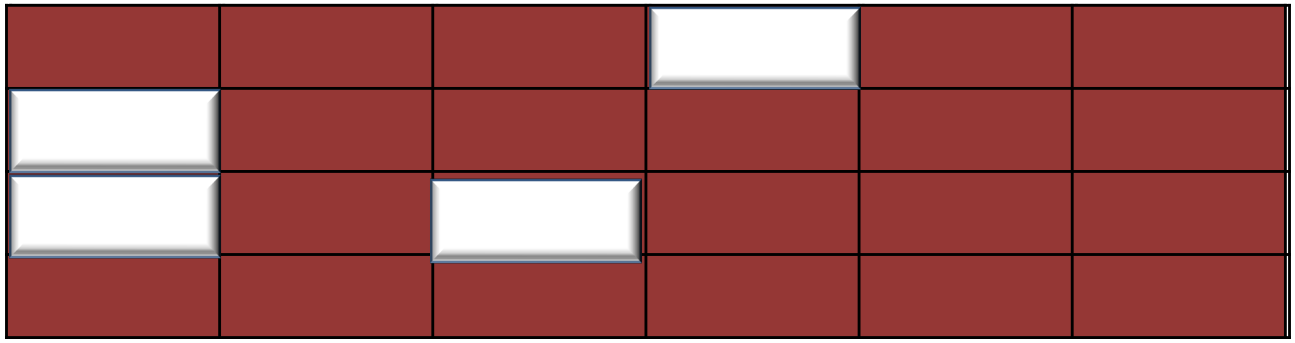
Control flow (sequence of instructions used) leaks secret

Detecting code path

$e_i = 0$



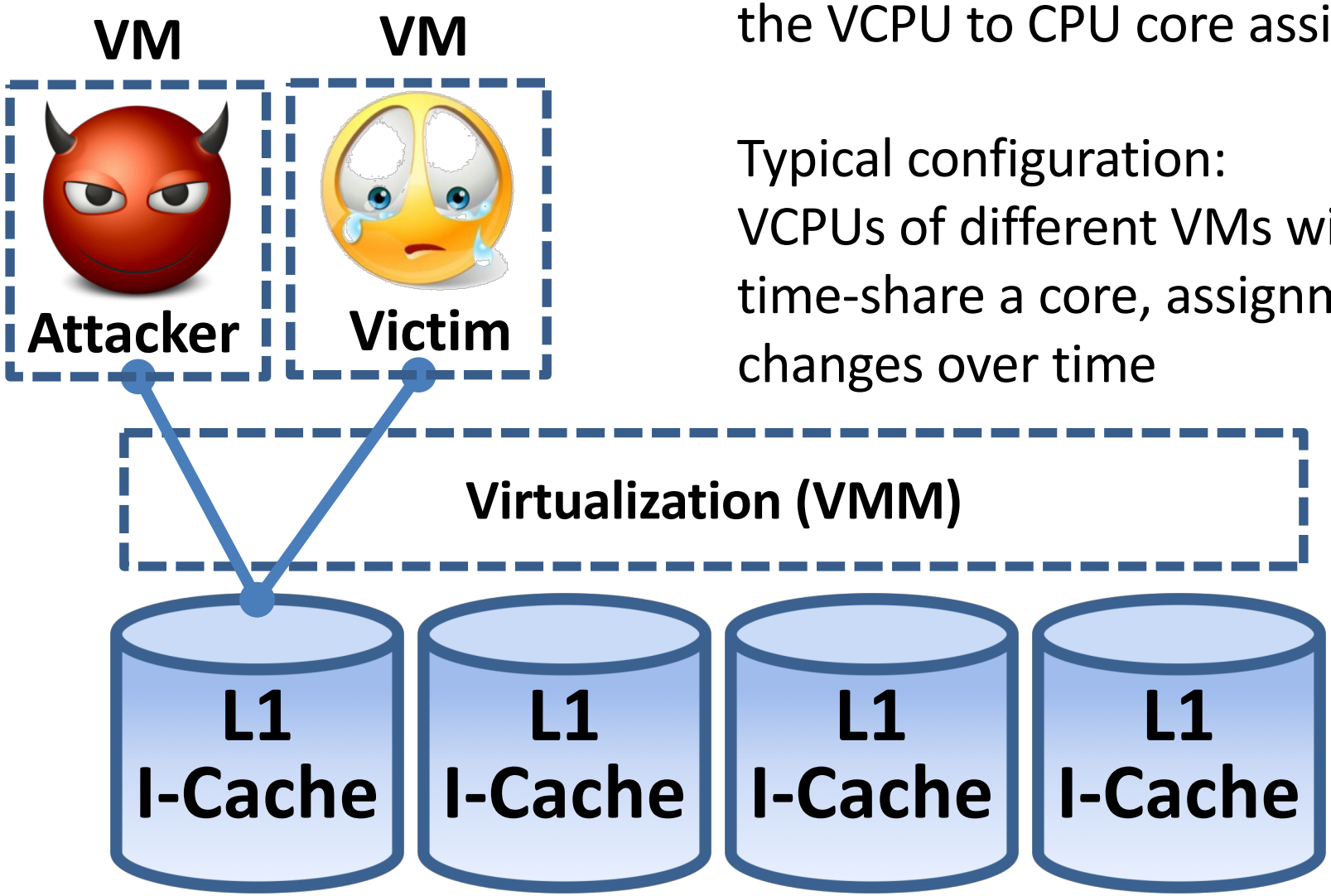
$e_i = 1$: extra instruction cache lines accessed



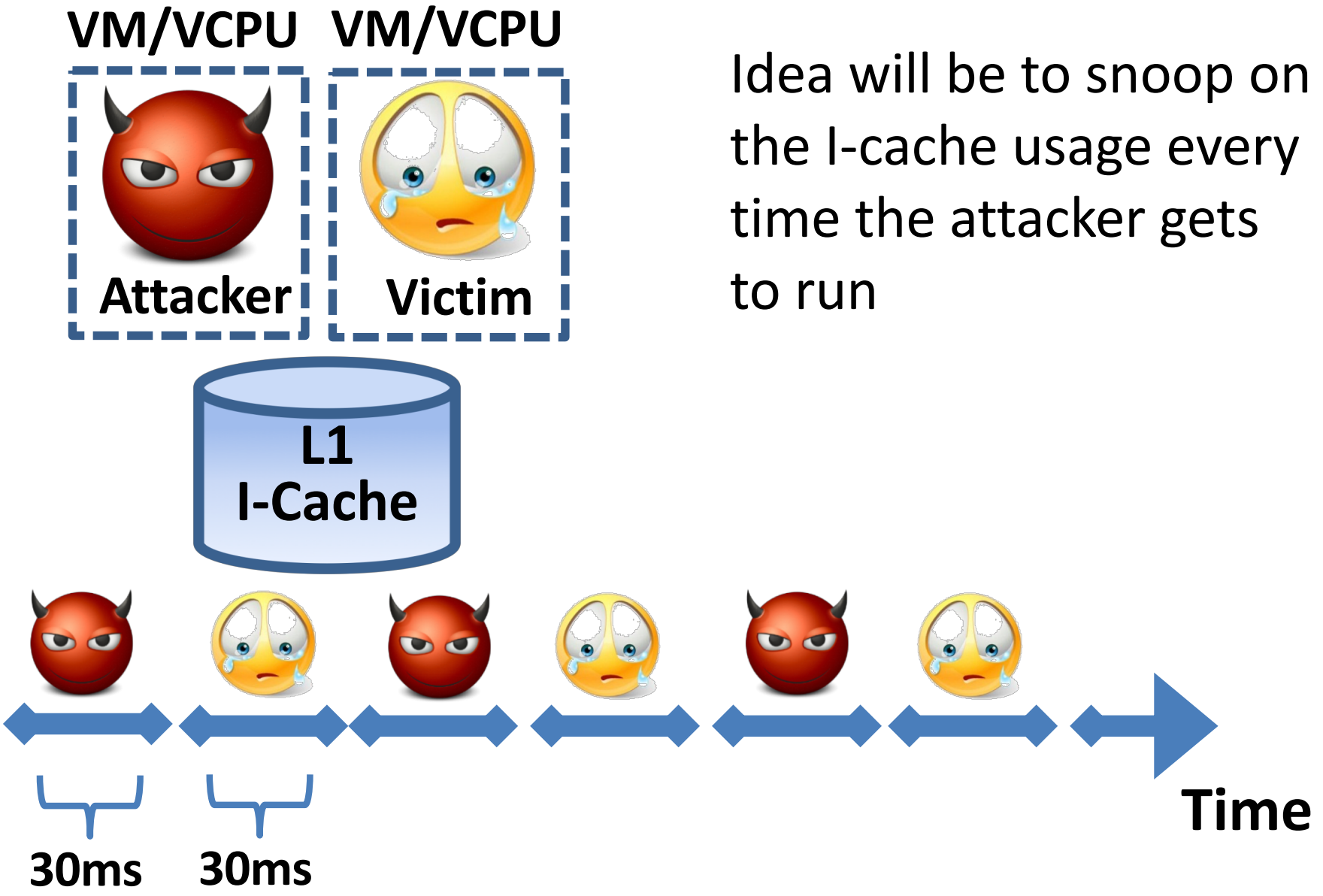
VMM core scheduling

VMM core scheduler determines the VCPU to CPU core assignment

Typical configuration:
VCPUs of different VMs will often time-share a core, assignment changes over time

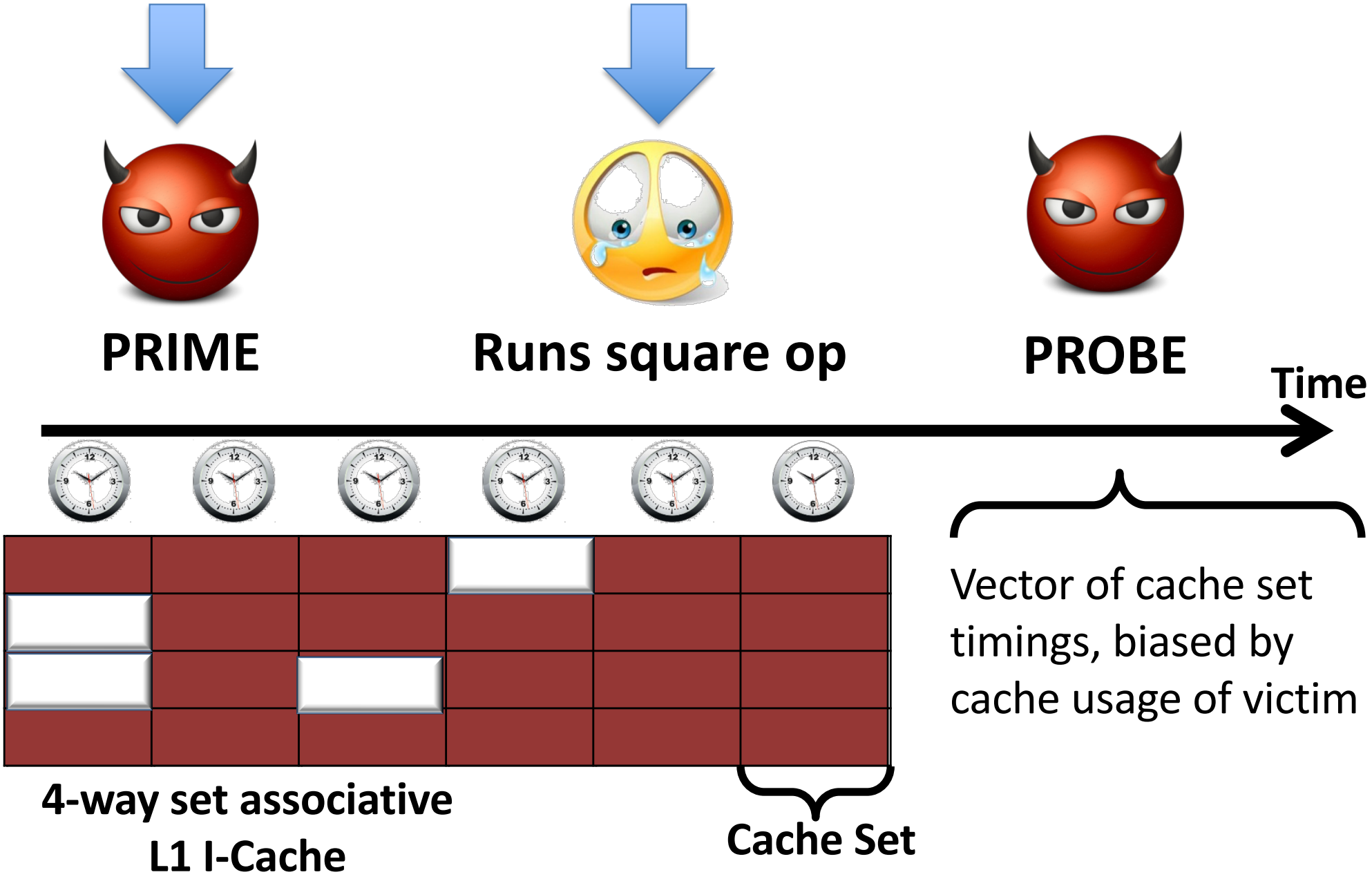


Time-sharing a core

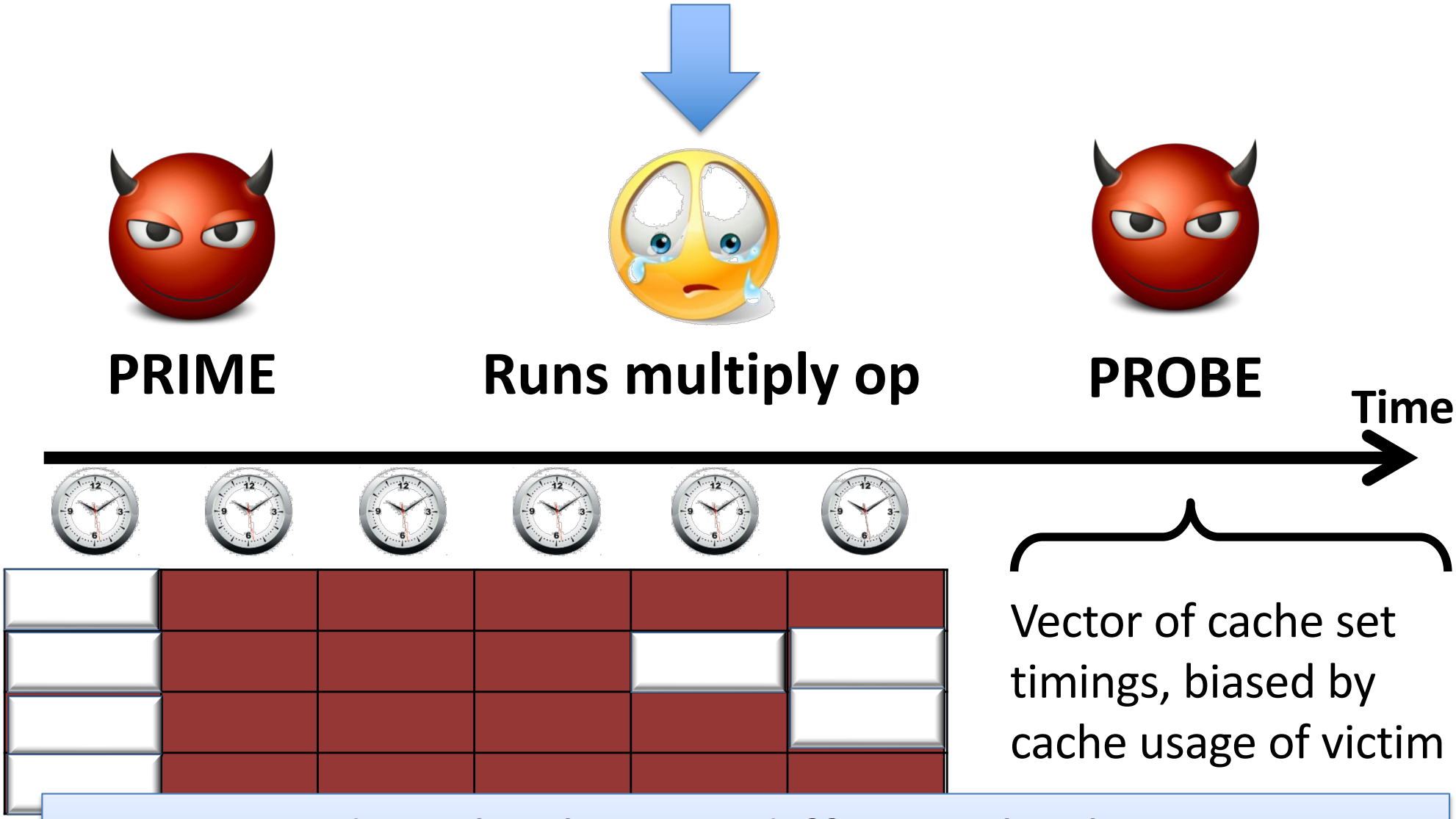


Idea will be to snoop on the I-cache usage every time the attacker gets to run

Prime-Probe Protocol

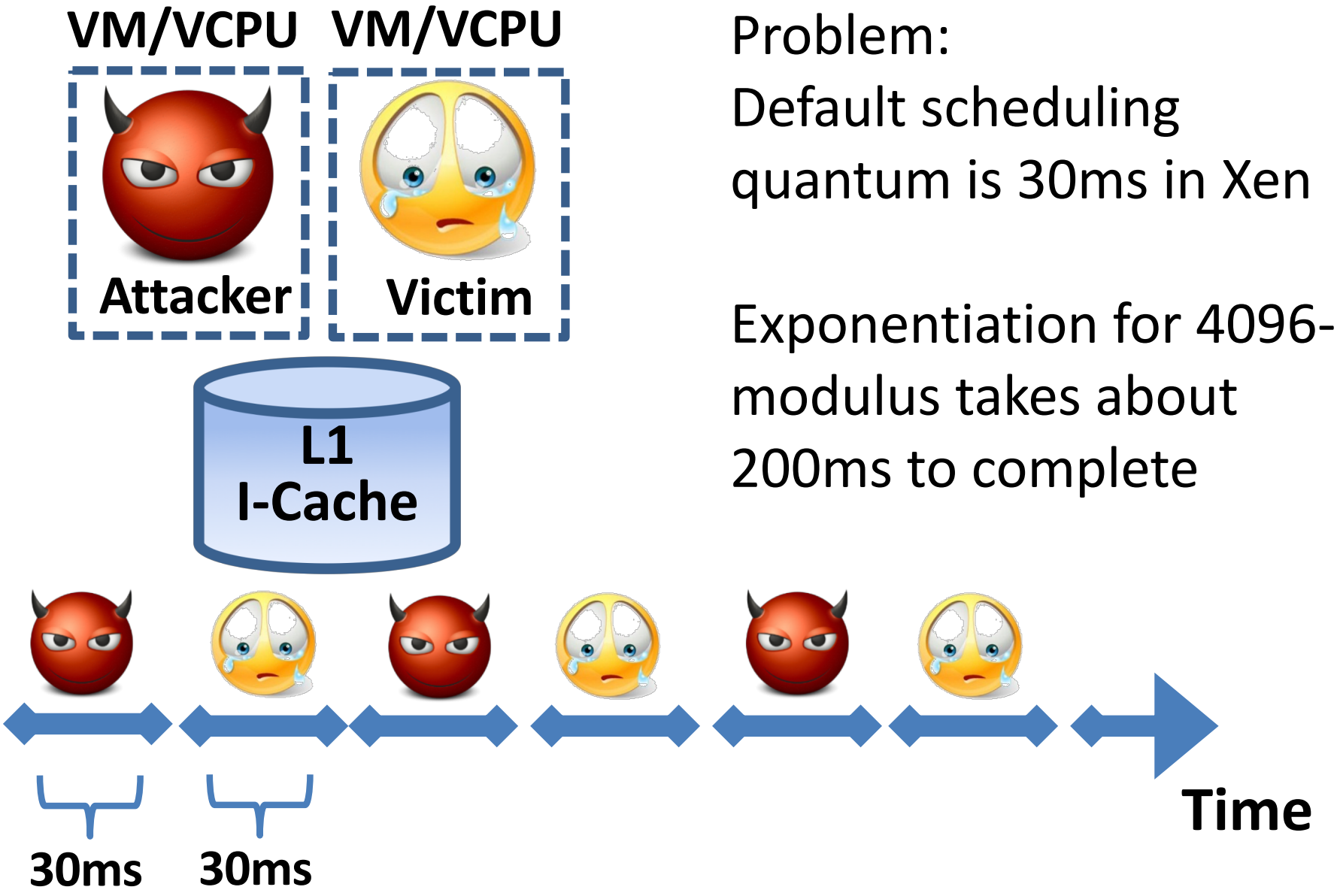


Prime-Probe Protocol



Square and Multiply give different-looking timing vectors (in the absence of noise)

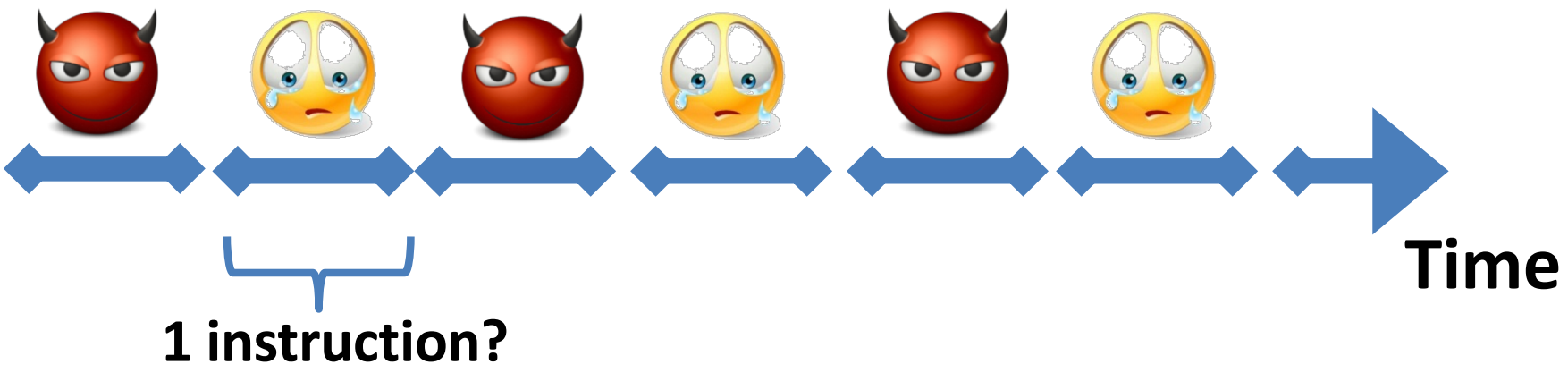
Time-sharing a core



Problem:
Default scheduling quantum is 30ms in Xen

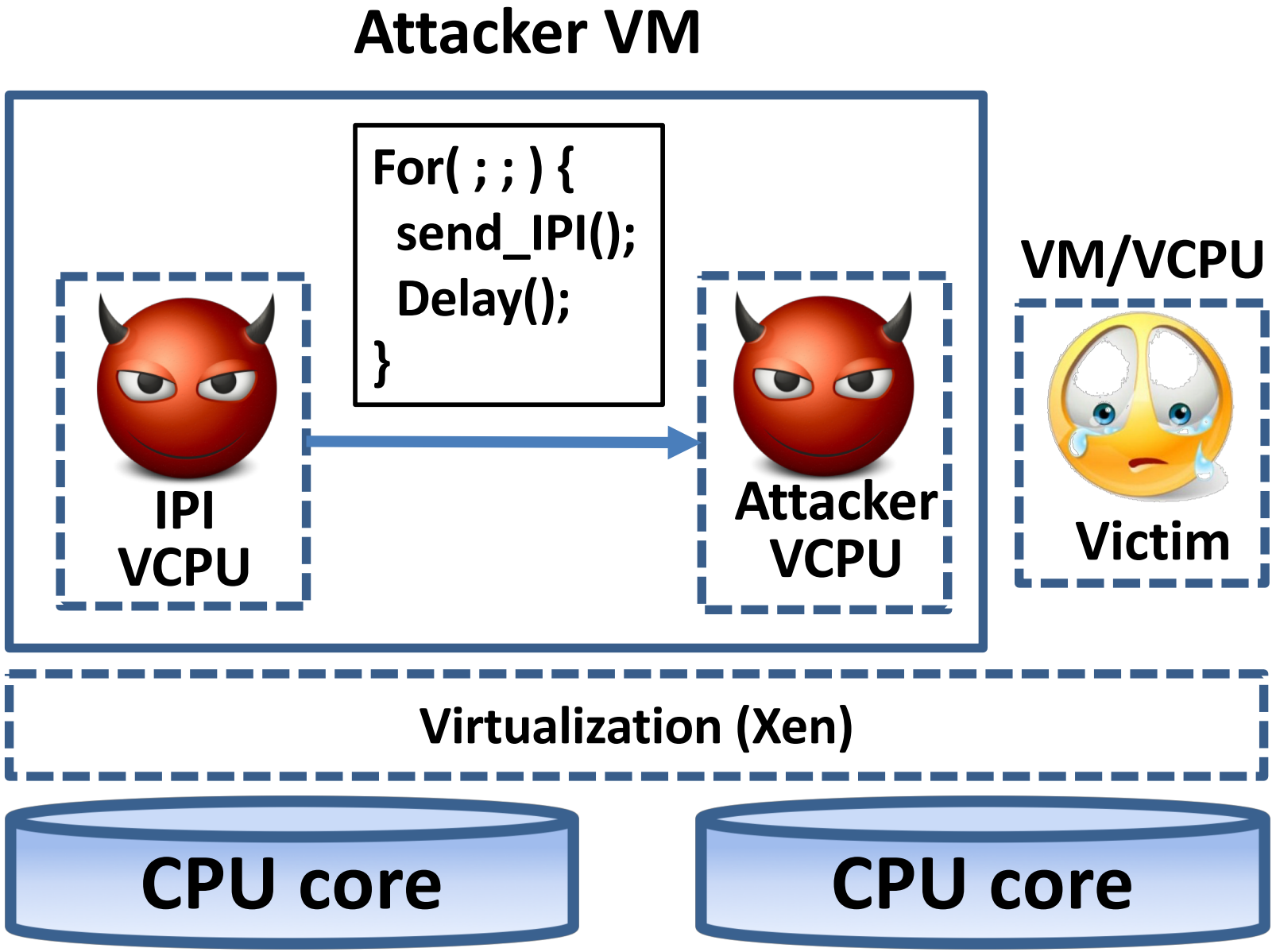
Exponentiation for 4096-bit modulus takes about 200ms to complete

Ideally …

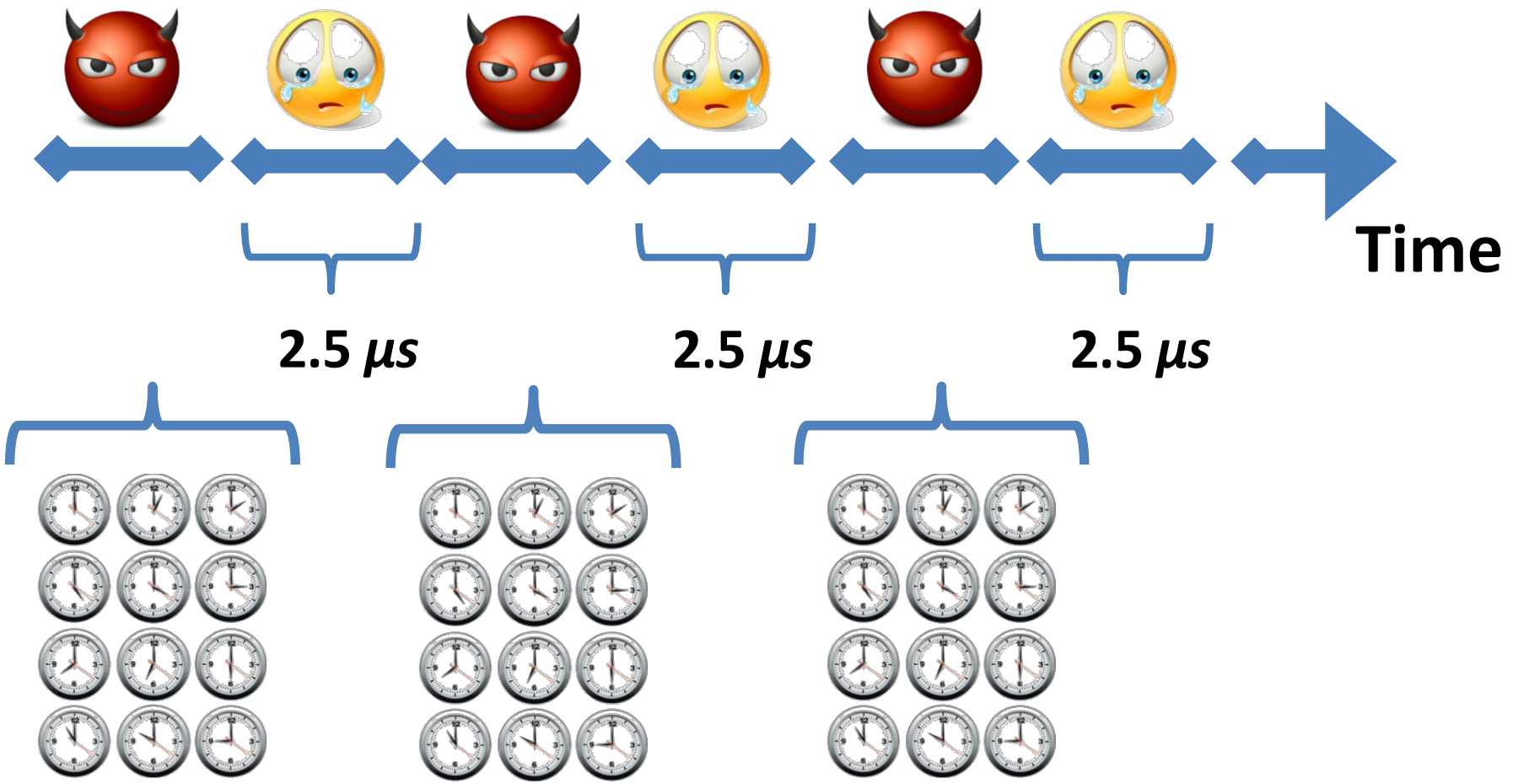


- Use **Interrupts** to preempt the victim:
  - **Inter-Processor interrupts (IPI)!**

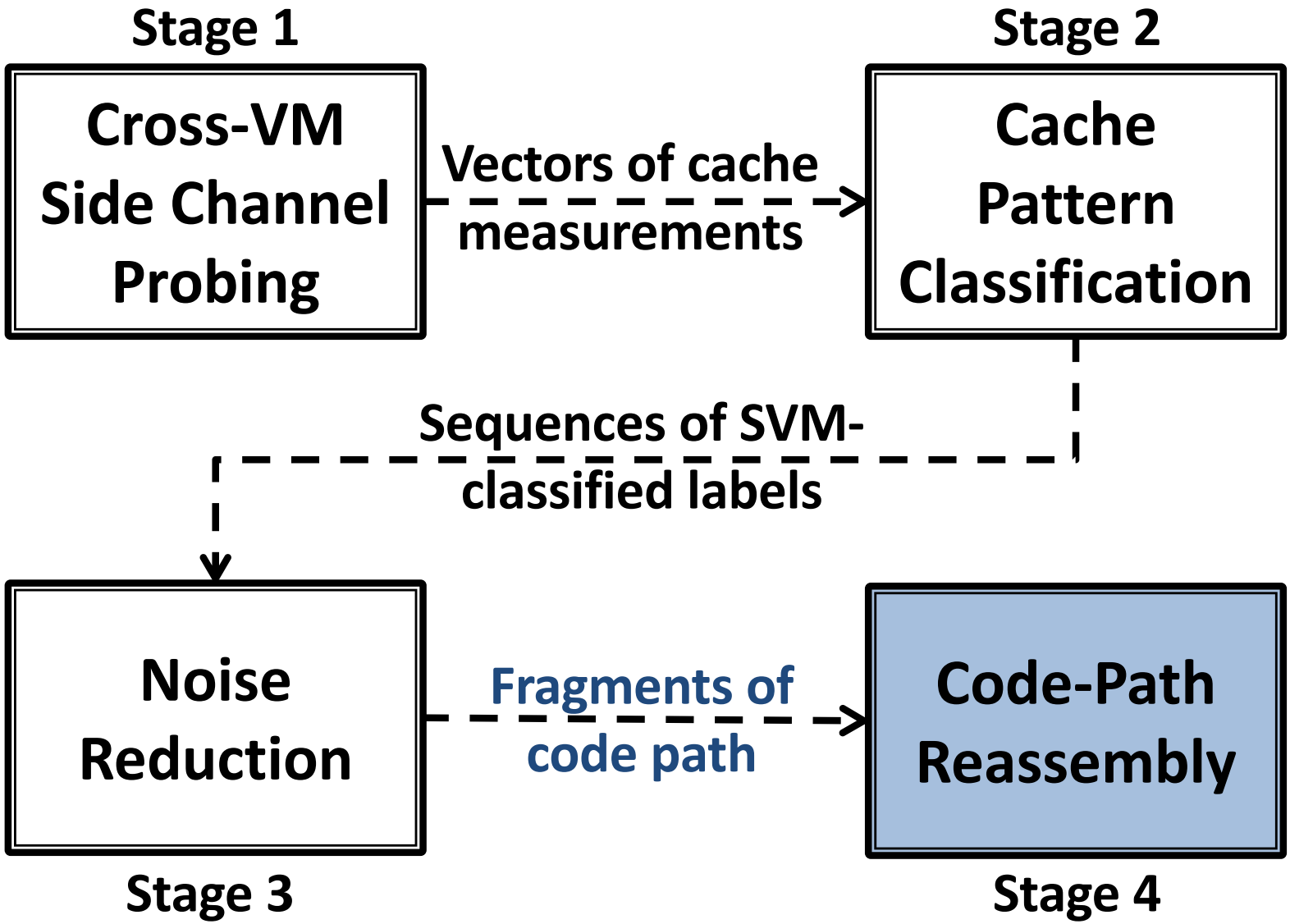
Inter-Processor Interrupts



Cross-VM Side Channel Probing



Outline


Stage 1
Cross-VM Side Channel Probing
Vectors of cache measurements
Stage 2
Cache Pattern Classification
Sequences of SVM-classified labels
Noise Reduction
Fragments of code path
Code-Path Reassembly
Stage 3
Stage 4

Evaluation

- Intel Yorkfield processor
  - 4 cores, 32KB L1 instruction cache
- Xen + linux + GnuPG + libgcrypt
  - Xen 4.0
  - Ubuntu 10.04, kernel version 2.6.32.16
  - Victim runs GnuPG v.2.0.19 (latest)
  - libgcrypt 1.5.0 (latest)
  - ElGamal decryption, 4096 bits

Results

- **Work-Conserving Scheduler**
  - 300,000,000 prime-probe results (6 hours)
  - Over 300 key fragments
  - Brute force the key in ~9800 guesses
- **Non-Work-Conserving Scheduler**
  - 1,900,000,000 prime-probe results (45 hours)
  - Over 300 key fragments
  - Brute force the key in ~6600 guesses

Lessons

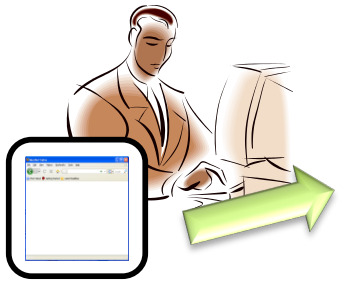
- But don't rely solely on them for:
  - VMM transparency
  - Containment
  - Strong isolation (side channels exist)
- Securing guest OS and host OS still very important for defense-in-depth

Virtual Machine Management

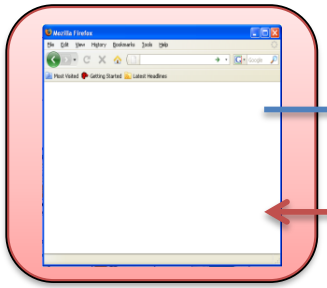
- Snapshots
  - Volume snapshot / checkpoint
    - persistent storage of VM
    - must boot from storage when resuming snapshot
  - Full snapshot
    - persistent storage and ephemeral storage (memory, register states, caches, etc.)
    - start/resume in between (essentially) arbitrary instructions
- VM image is a file that stores a snapshot

Virtual machines and secure browsing

"**Protect Against Adware and Spyware:** Users protect their PCs against adware, spyware and other malware while browsing the Internet with Firefox in a virtual machine."
[http://www.vmware.com/company/news/releases/player.html]

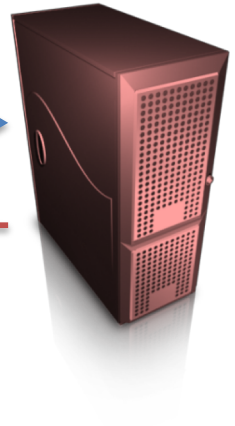


Clean snapshot of VM with browser running



http://www.freesoftware.com/

browser exploit



Virtual machine compromised, but not host OS

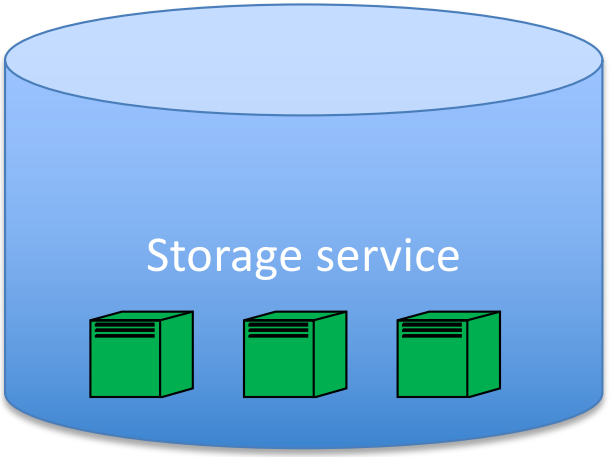
Resetting to snapshot removes malware

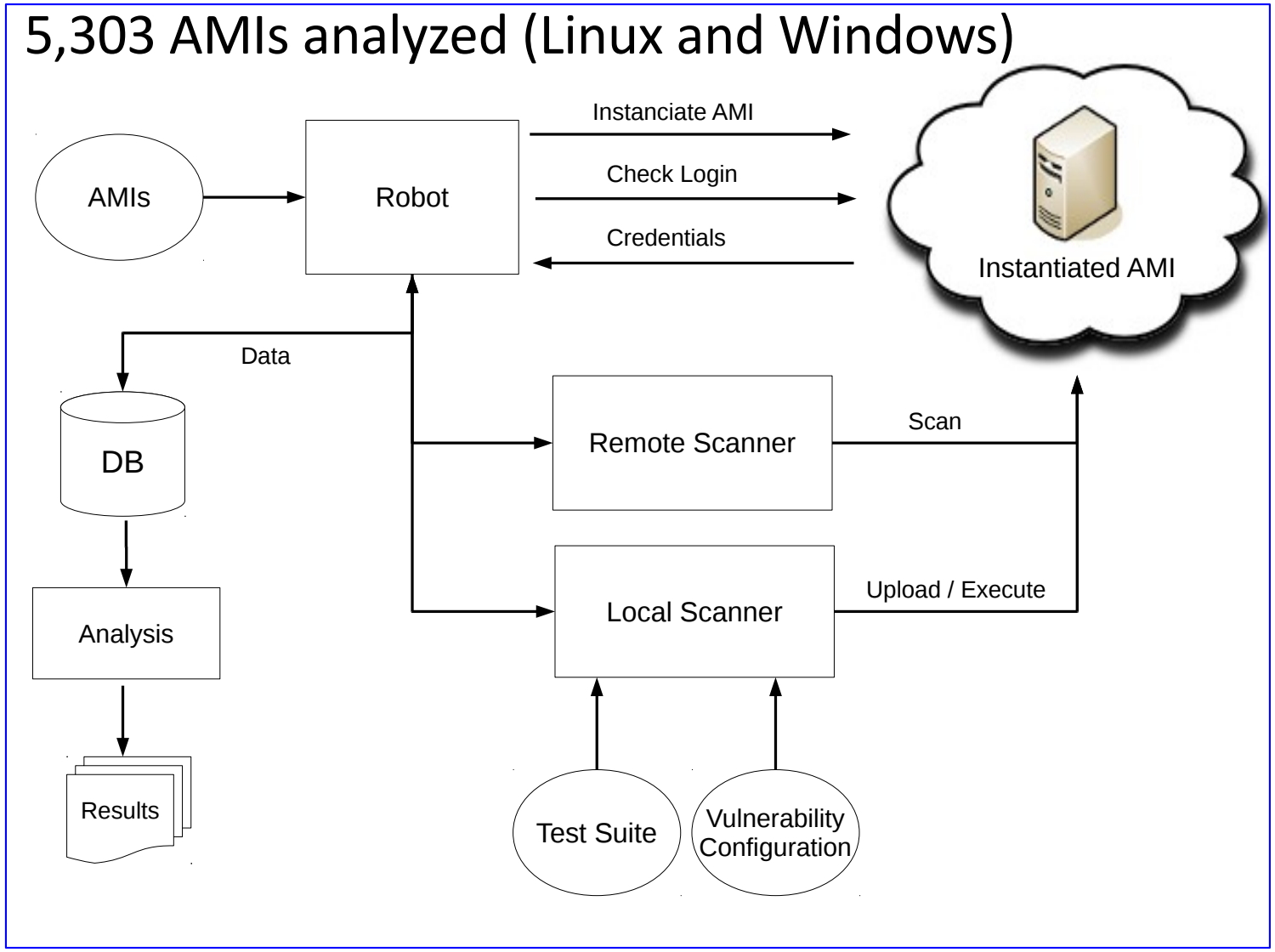
VM Management issues

- Reset vulnerabilities
  - Reuse of randomness
- Lack of diversity
- Identity management / credentials
- Known vulnerabilities

Amazon Machine Images (AMIs)

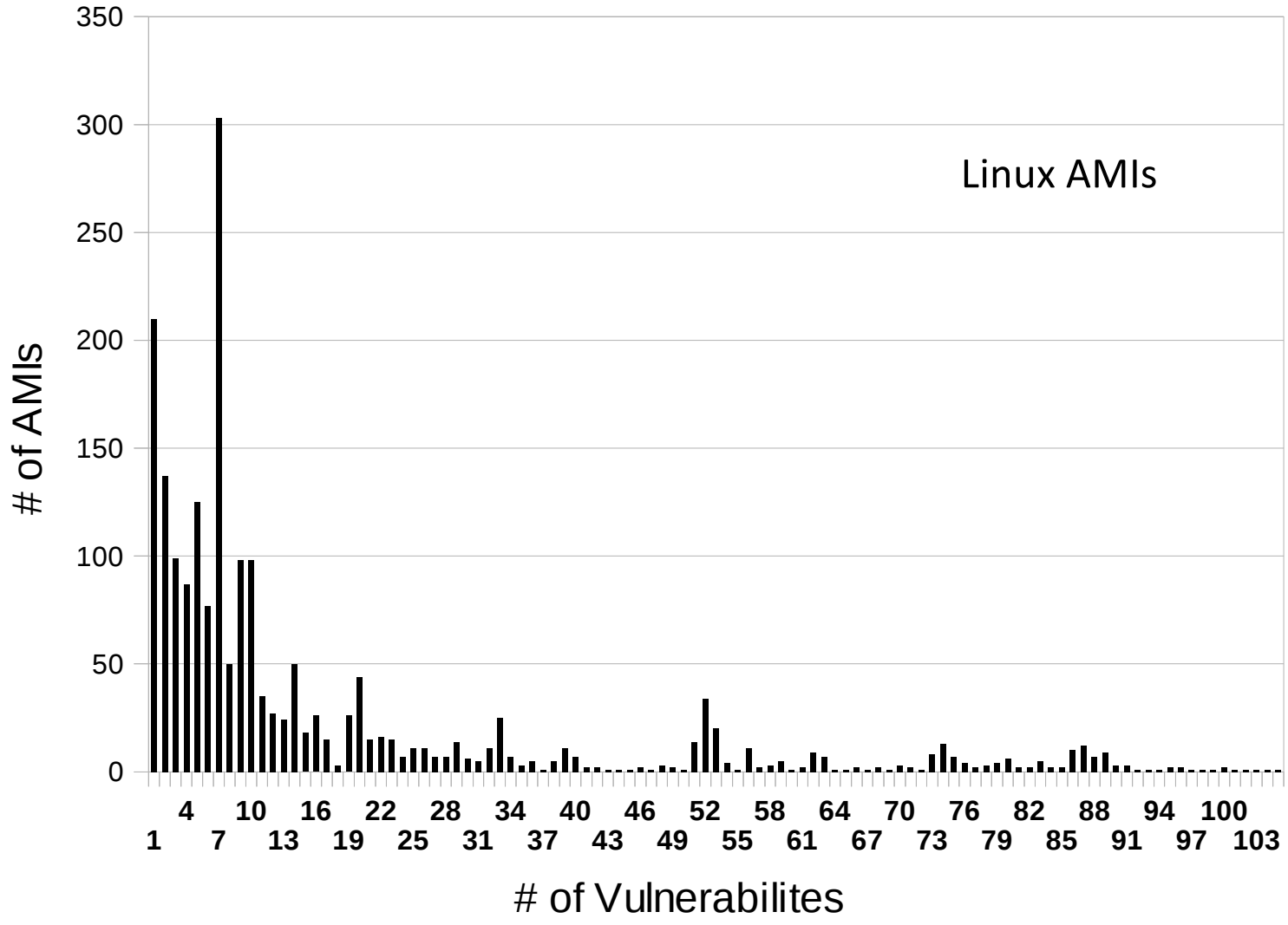
- Users set up volume snapshots / checkpoints that can then be run on the Elastic Compute Cloud (EC2)
- Can be marked as public and anyone can use your AMI





Balduzzi et al. “A Security Analysis of Amazon’s Elastic Compute Cloud Service – Long Version –”, 2011

See also Bugiel et al., “AmazonIA: When Elasticity Snaps Back”, 2011



Also: Malware found on a couple AMIs

Balduzzi et al. analysis

- Backdoors
  - AMIs include SSH public keys within authorized_keys
  - Password-based backdoors

	East	West	EU	Asia	Total
AMIs (%)	34.8	8.4	9.8	6.3	21.8
With Passwd	67	10	22	2	101
With SSH keys	794	53	86	32	965
With Both	71	6	9	4	90
Superuser Priv.	783	57	105	26	971
User Priv.	149	12	12	12	185

Table 2: Left credentials per AMI

Balduzzi et al. analysis

- Credentials for other systems
  - AWS secret keys (to control EC2 services of an account): 67 found
  - Passwords / secret keys for other systems: 56 found

Finding	Total	Image	Remote
Amazon RDS	4	0	4
dDNS	1	0	1
SQL	7	6	1
MySql	58	45	13
WebApp	3	2	1
VNC	1	1	0
Total	74	54	20

Table 3: Credentials in history files

Balduzzi et al. analysis

- Deleted files
  - One AMI creation method does block-level copying

Type	#
Home files (`/home, /root`)	33,011
Images (min. 800x600)	1,085
Microsoft Office documents	336
Amazon AWS certificates and access keys	293
SSH private keys	232
PGP/GPG private keys	151
PDF documents	141
Password file (`/etc/shadow`)	106

Table 5: Recovered data from deleted files

Response

"They told me it's not their concern, they just provide computing power," Balduzzi says. "It's like if you upload naked pictures to Facebook. It's not a good practice, but it's not Facebook's problem."

http://www.forbes.com/sites/andygreenberg/2011/11/08/researchers-find-amazon-cloud-servers-teeming-with-backdoors-and-other-peoples-data/

- Amazon notified customers with vulnerable AMIs
- Made private AMIs of non-responsive customers
- New tutorials for bundling systems
- Working on undelete issues…

Lessons

- New software management practices needed with VM snapshots
- Discussion:
  - New tool support?
  - How much worse is this than non-cloud server deployments?