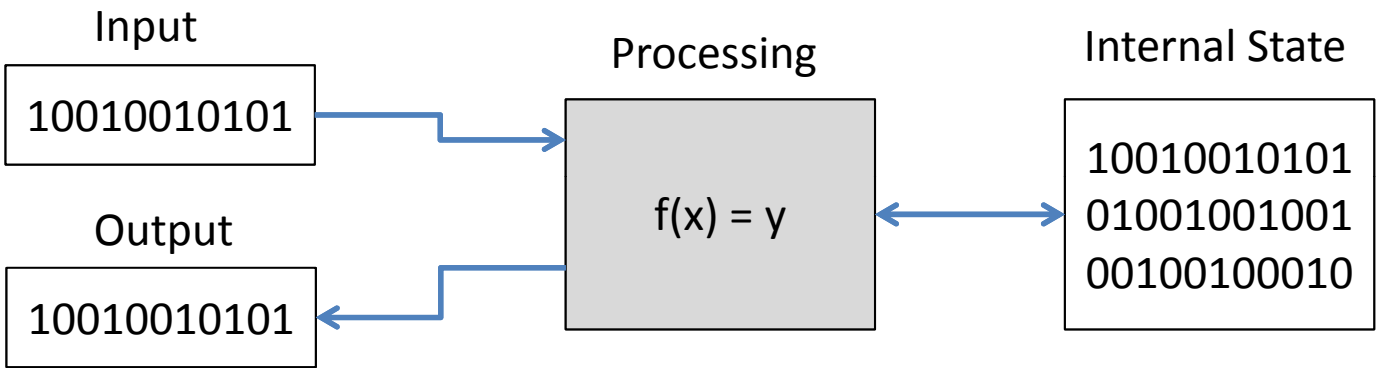
CPU as the New Perimeter

Attestation and Memory Encryption Protect Sensitive Data in the Cloud

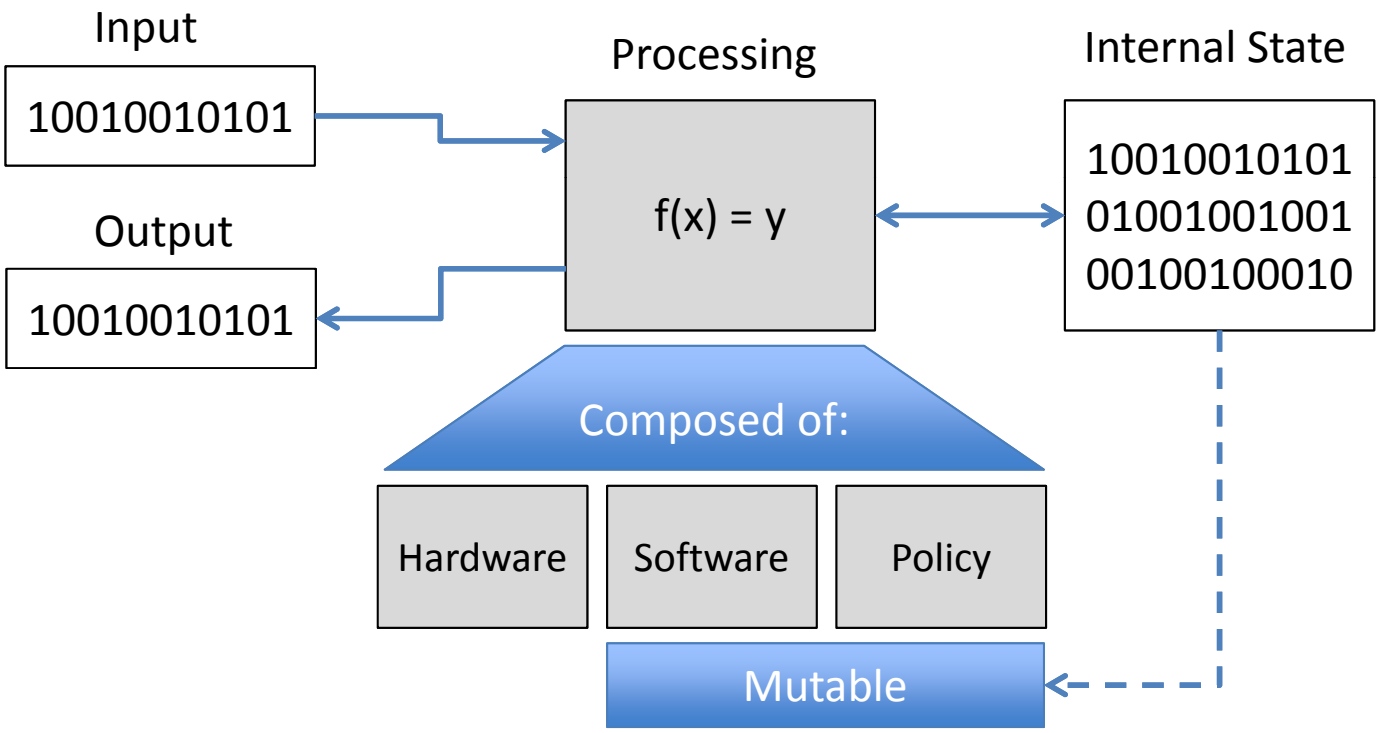
Oded Horovitz
Co-Founder & CEO
PrivateCore Inc


privatecore
the private computing company

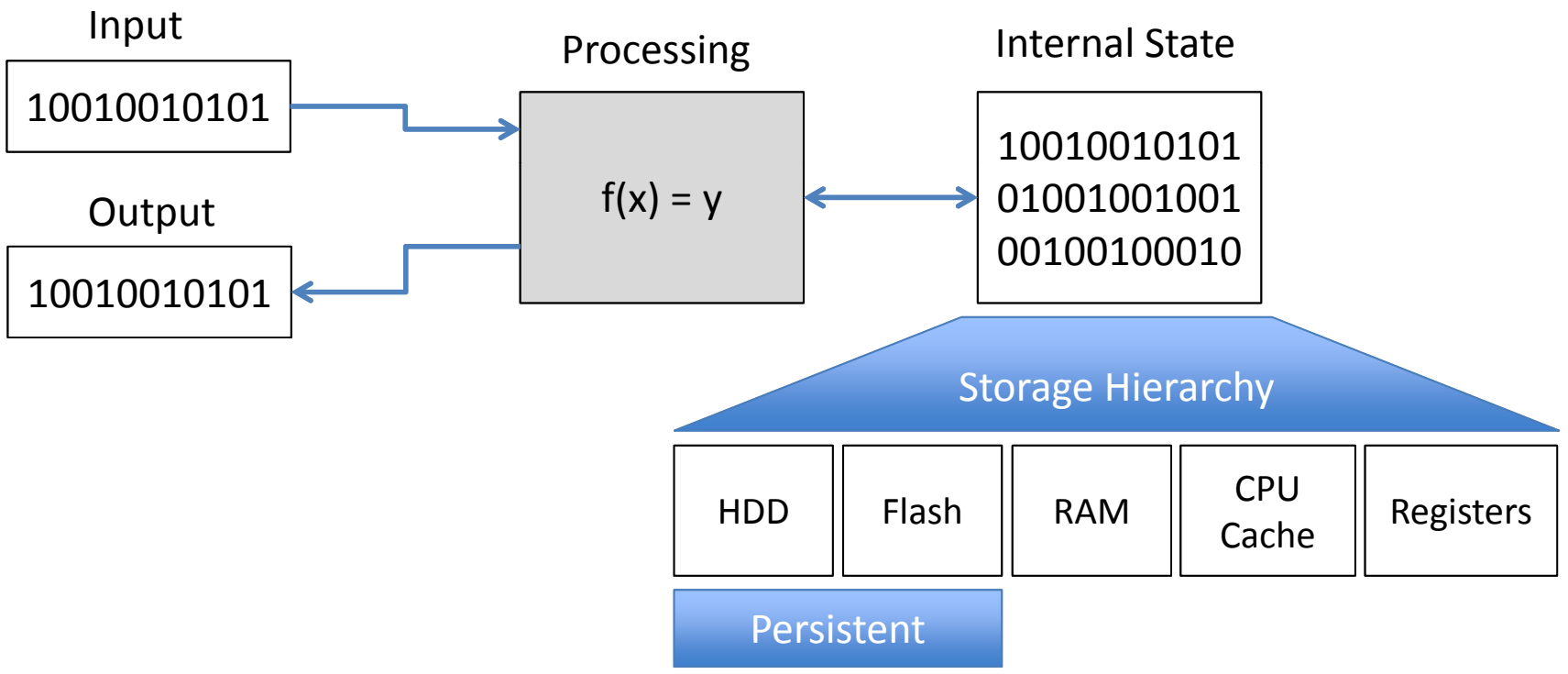
Computation



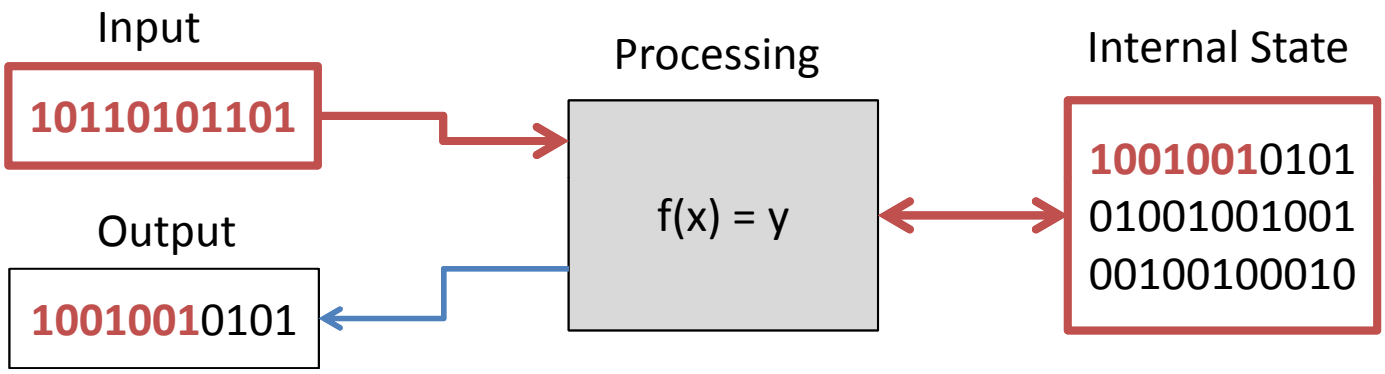
Processing composition



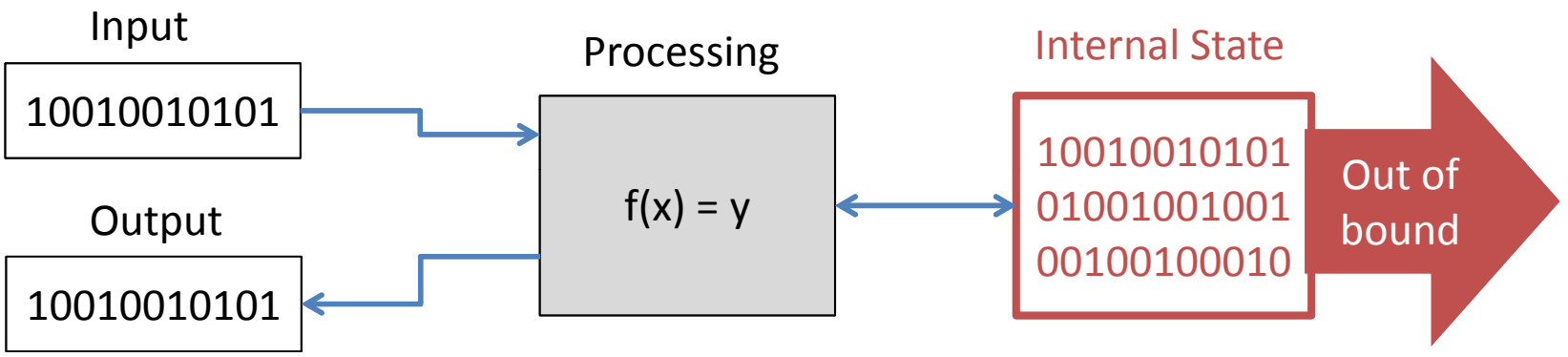
Internal Storage hierarchy



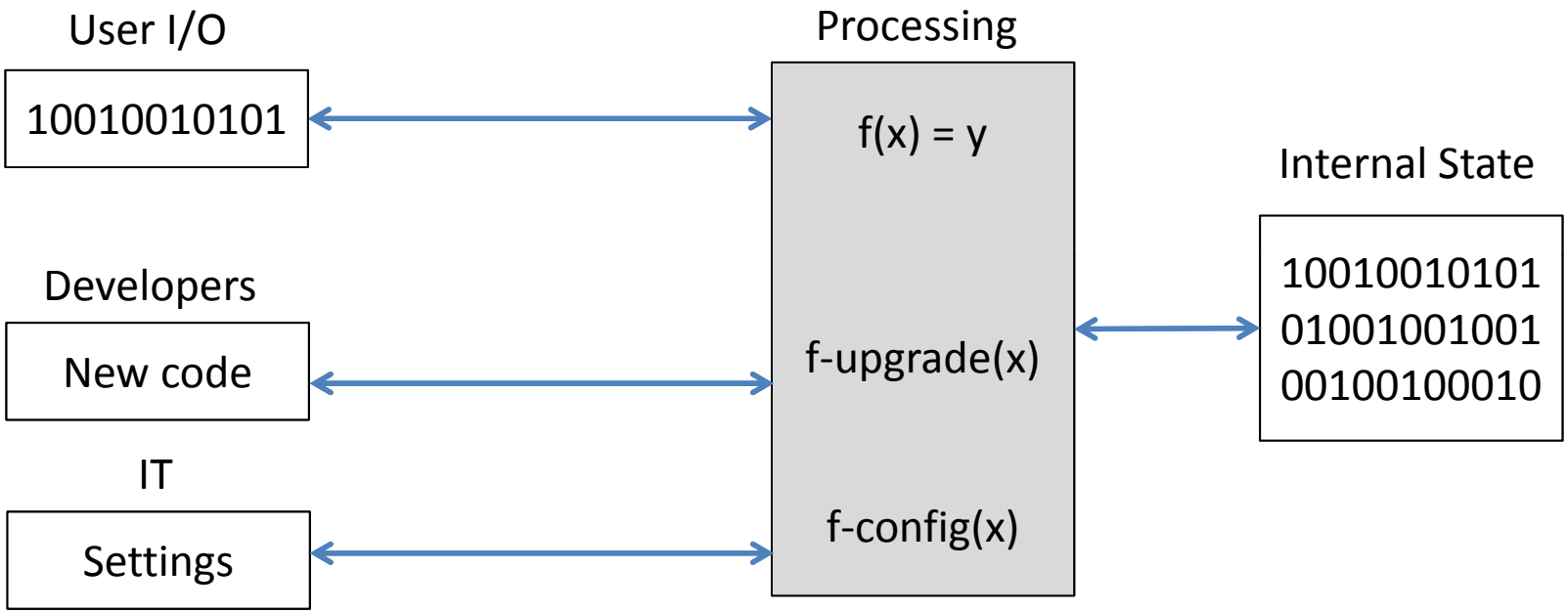
Hacking, exploits existing vulnerabilities



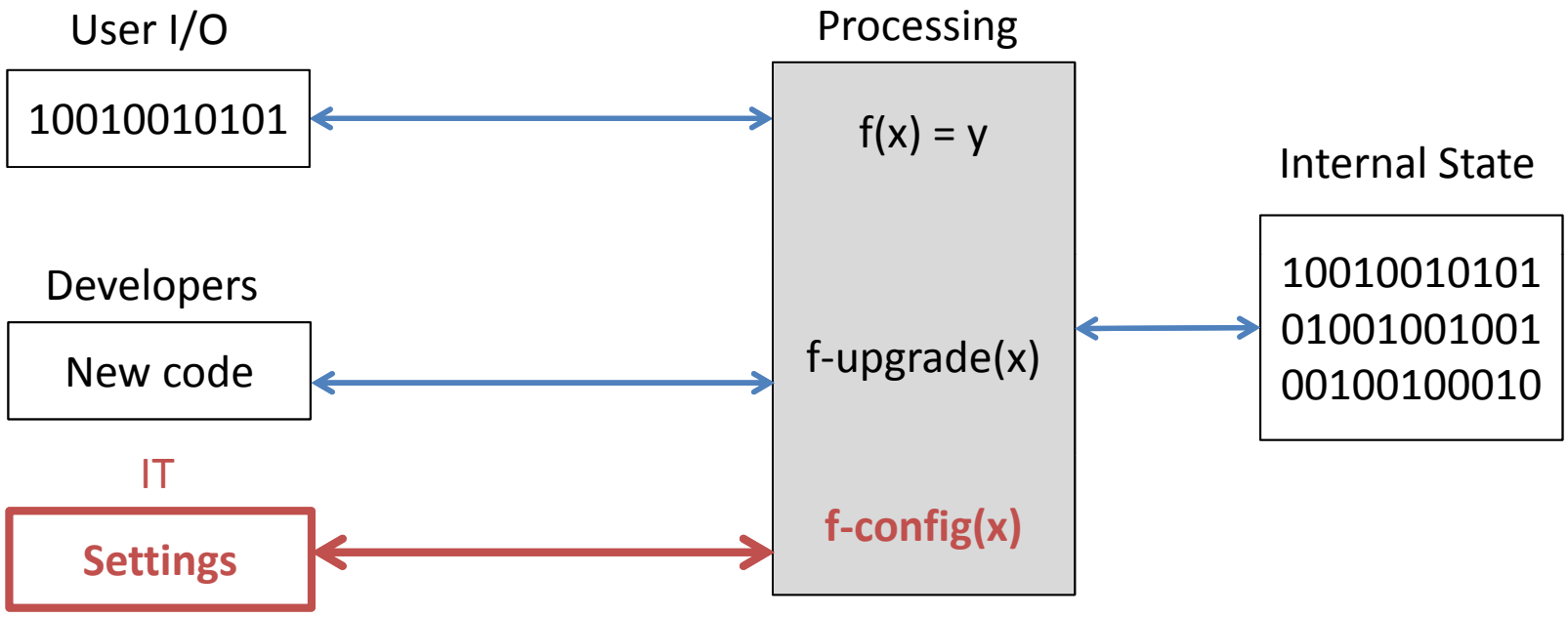
Physical attack, walking with the data



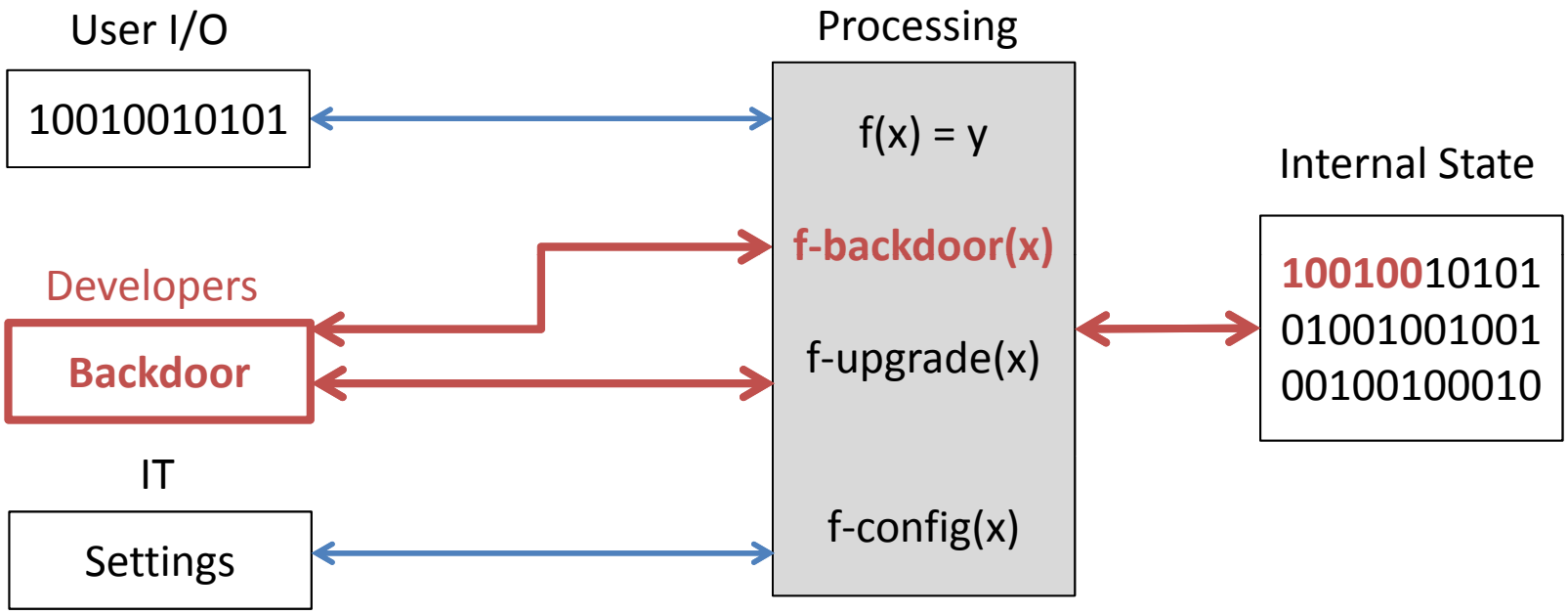
Lets add Operations



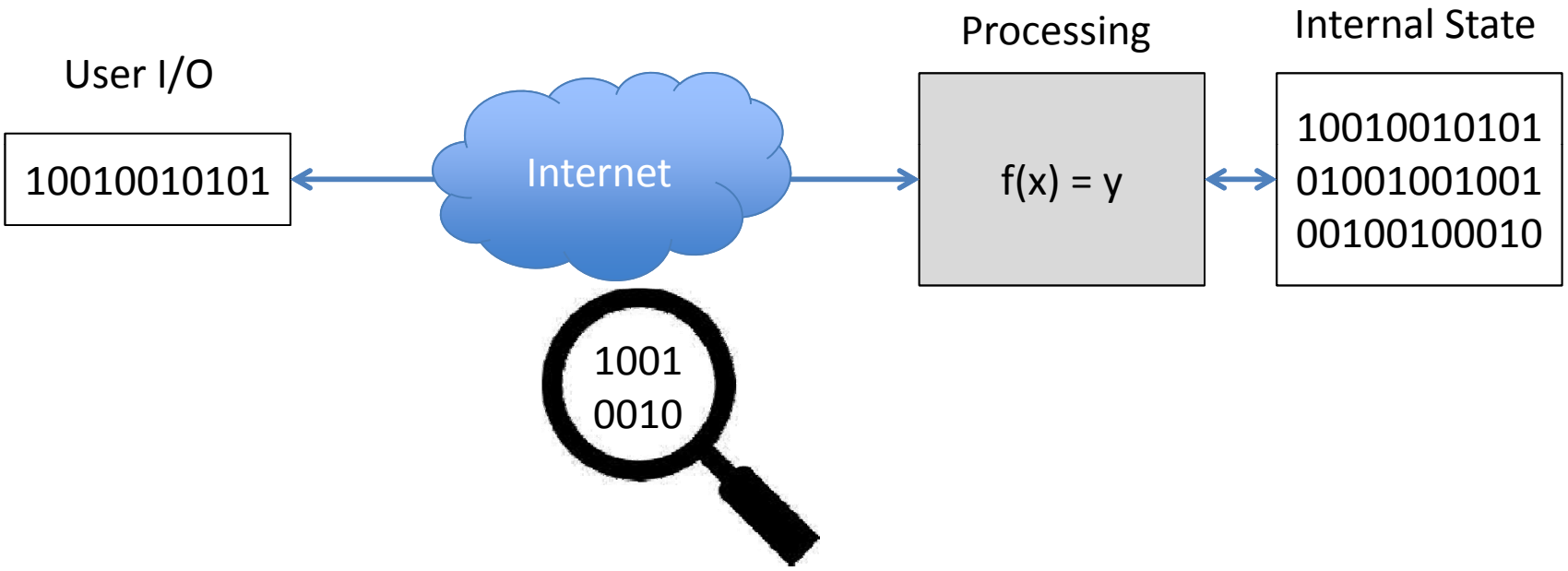
Admin hack - self provision access



Developer hack – Introducing backdoors

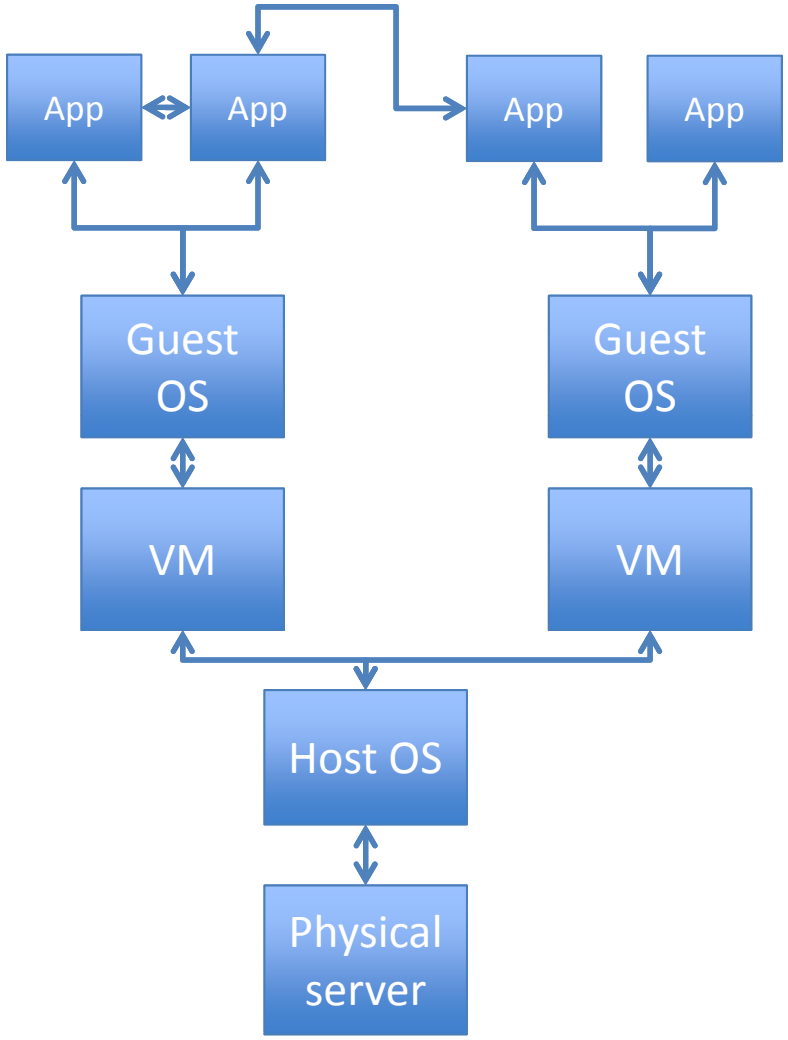


Add risk of public communications



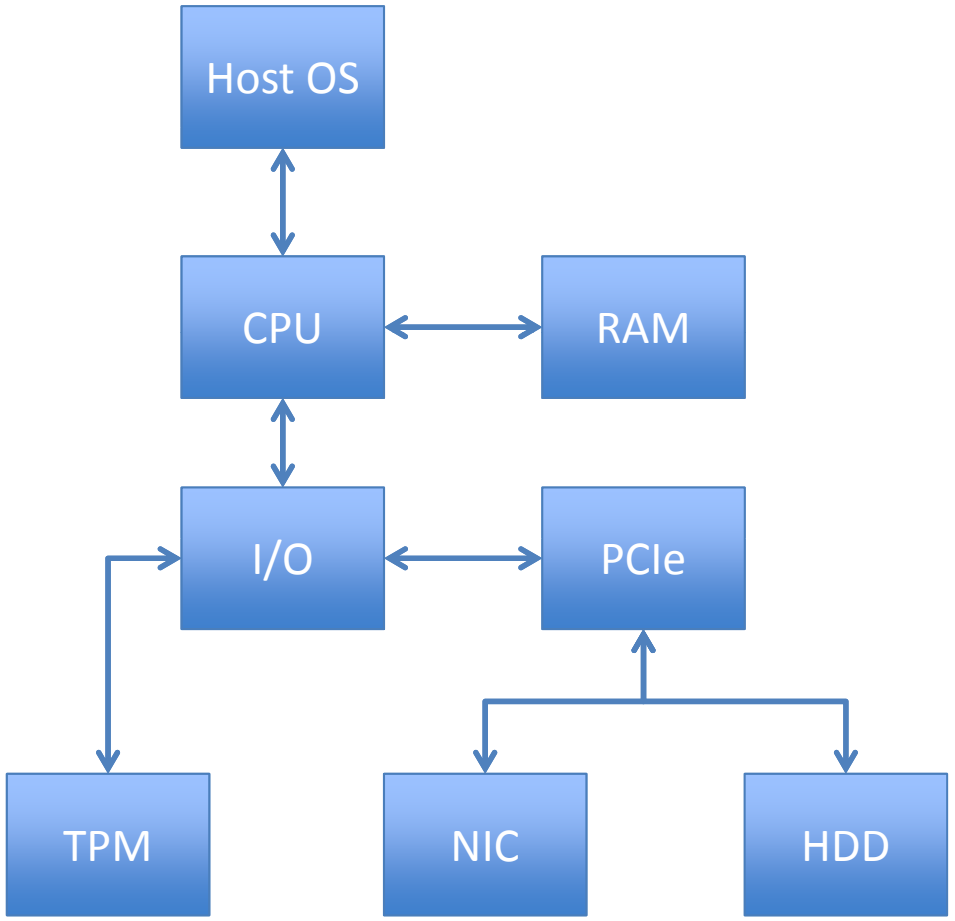
Also, real systems show complex composition

Up the stack!



Still, real systems show complex composition

Down the stack!



Sample attacks at the IaaS level

Integrity attacks

SMM infection

HDD firmware infection

injected kernel arguments

Physical attacks

Grabbing clear private SSH keys

Cold-boot

Logical access attacks

Inception

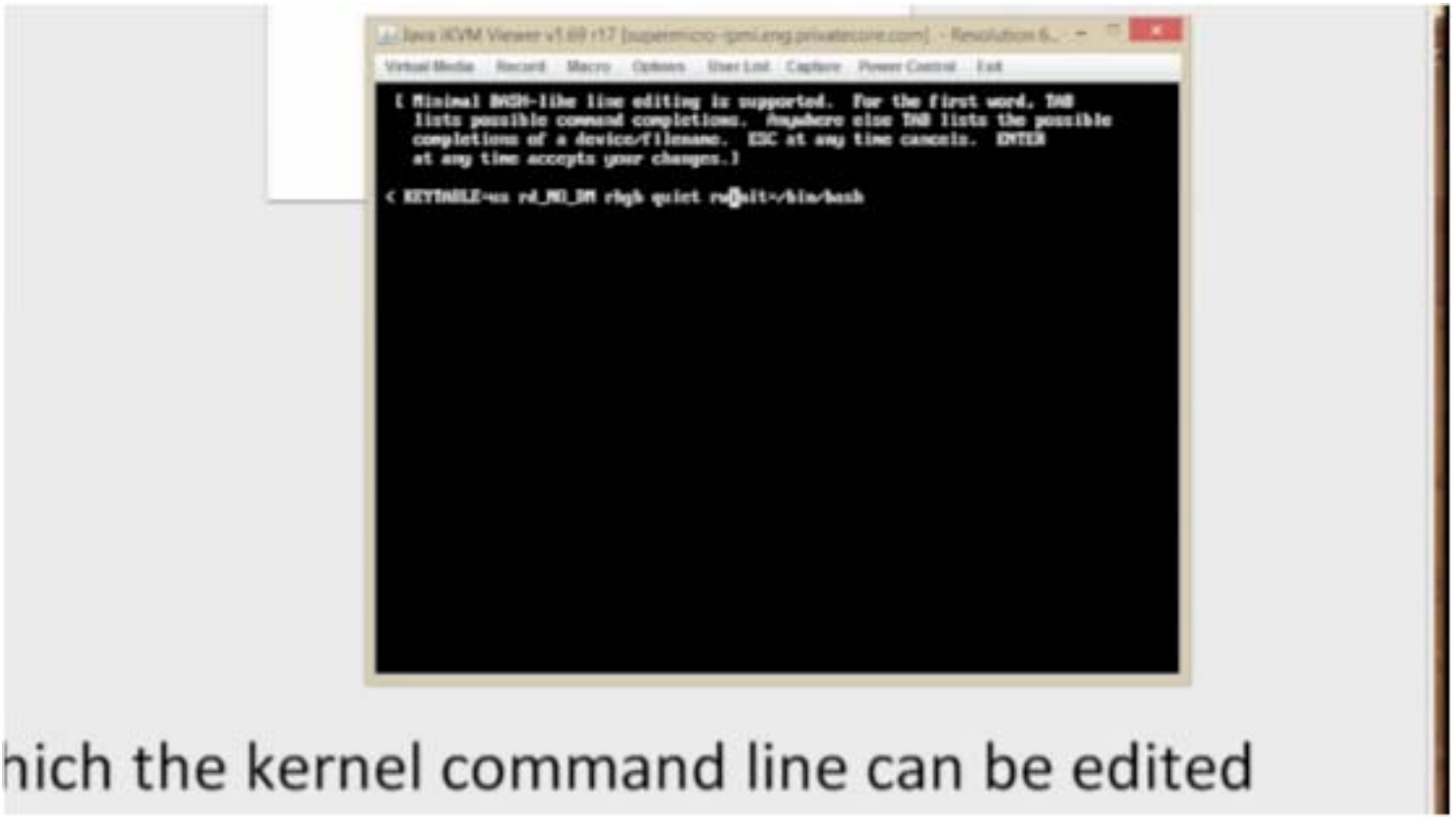
DMA capture of mysql records

Malicious device I/O

SMM Infection, execution integrity forever lost

HDD firmware infection, WYSINWYG

Injected kernel argument & SSH key grab



http://youtu.be/6C0b3nMXeGU

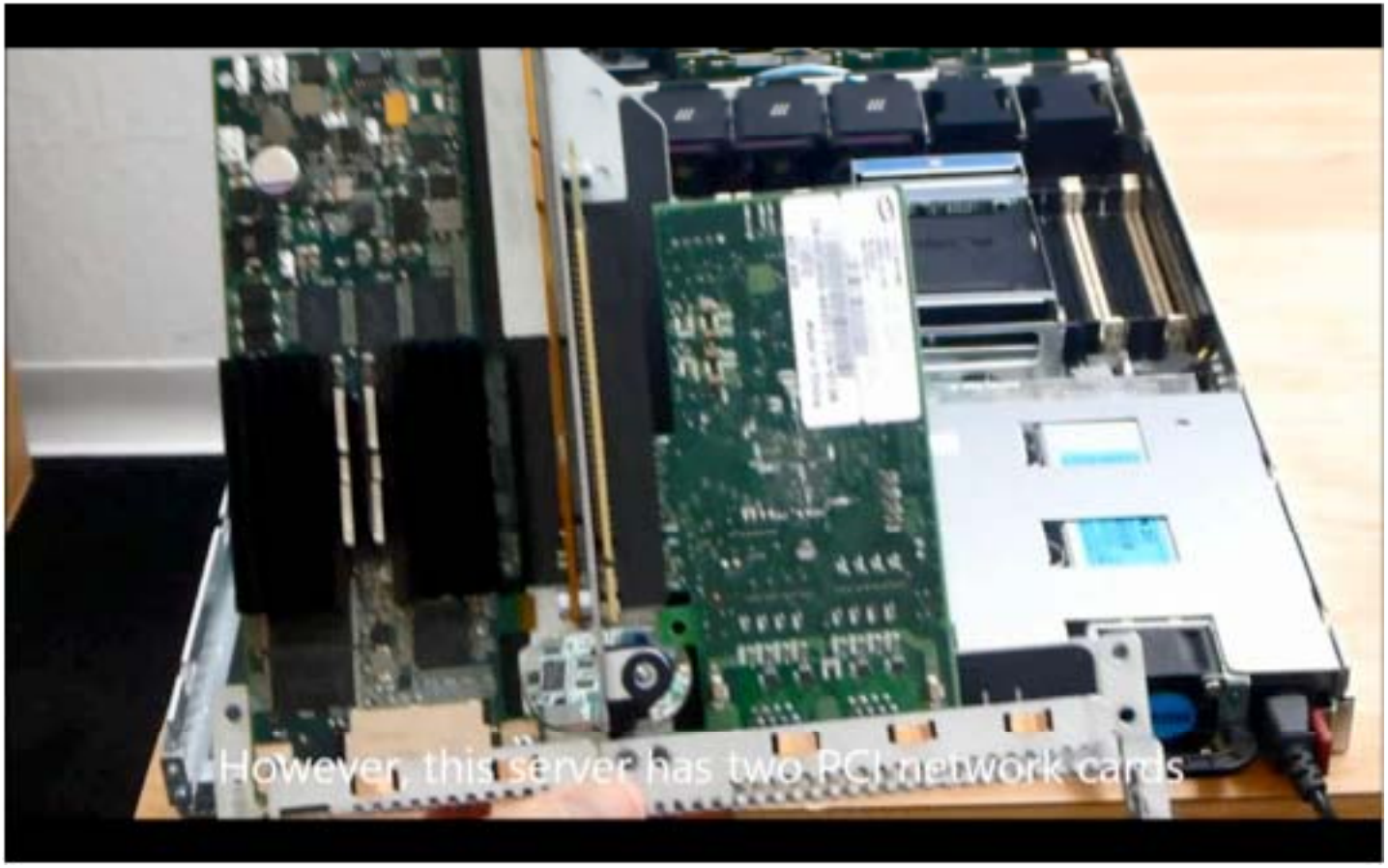
Cold-boot attack, grabbing memory

http://youtu.be/5SKq9o0Luyo

Inception rewriting your memory

http://youtu.be/wki66w1iJHA

DMA IaaS (Inception-as-a-Service)



http://youtu.be/AI-XbzKO7HM

Malicious device I/O

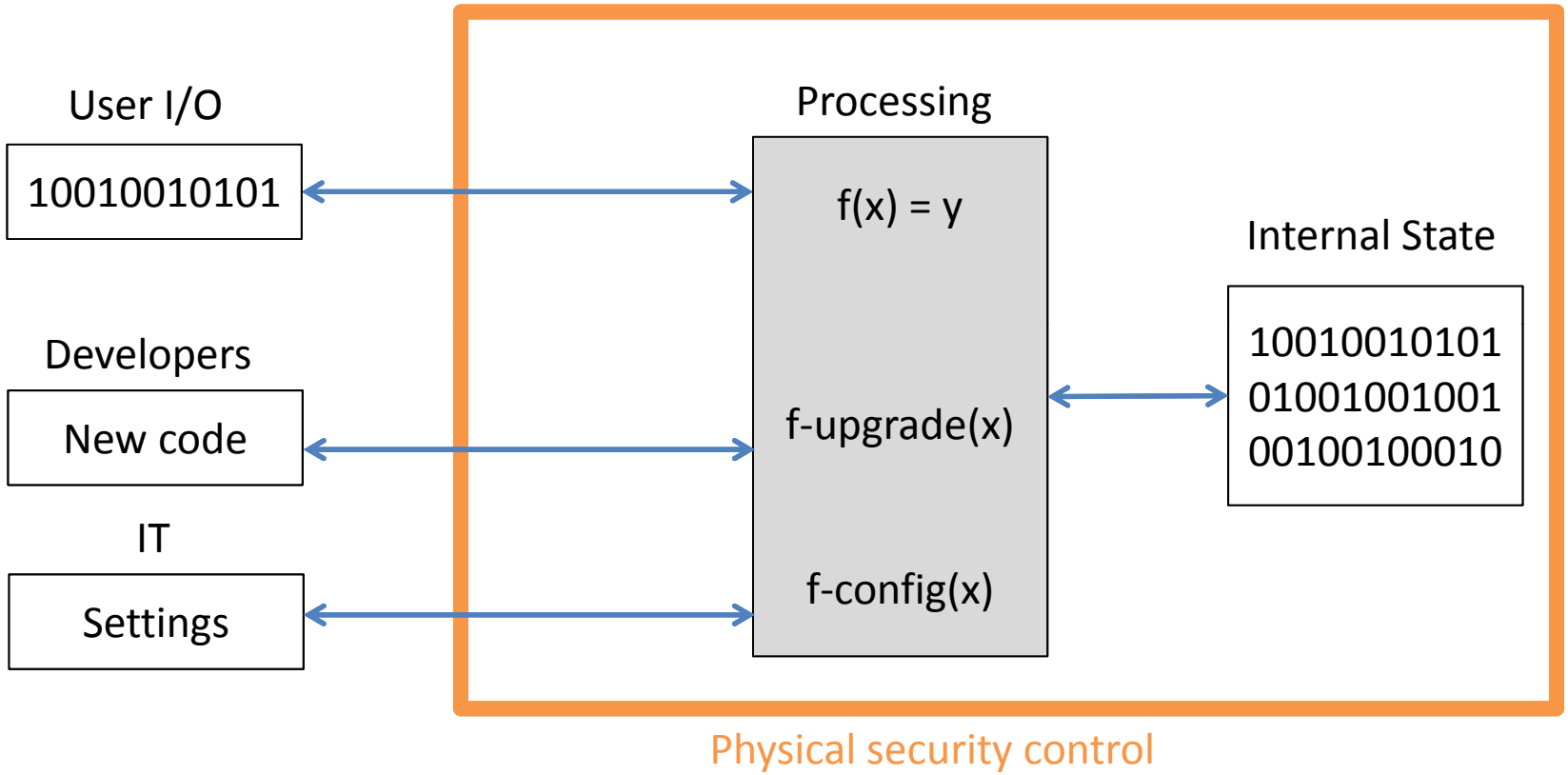
OS Developers are not writing defensive device drivers…

In response for our submitted drivers vulnerabilities:

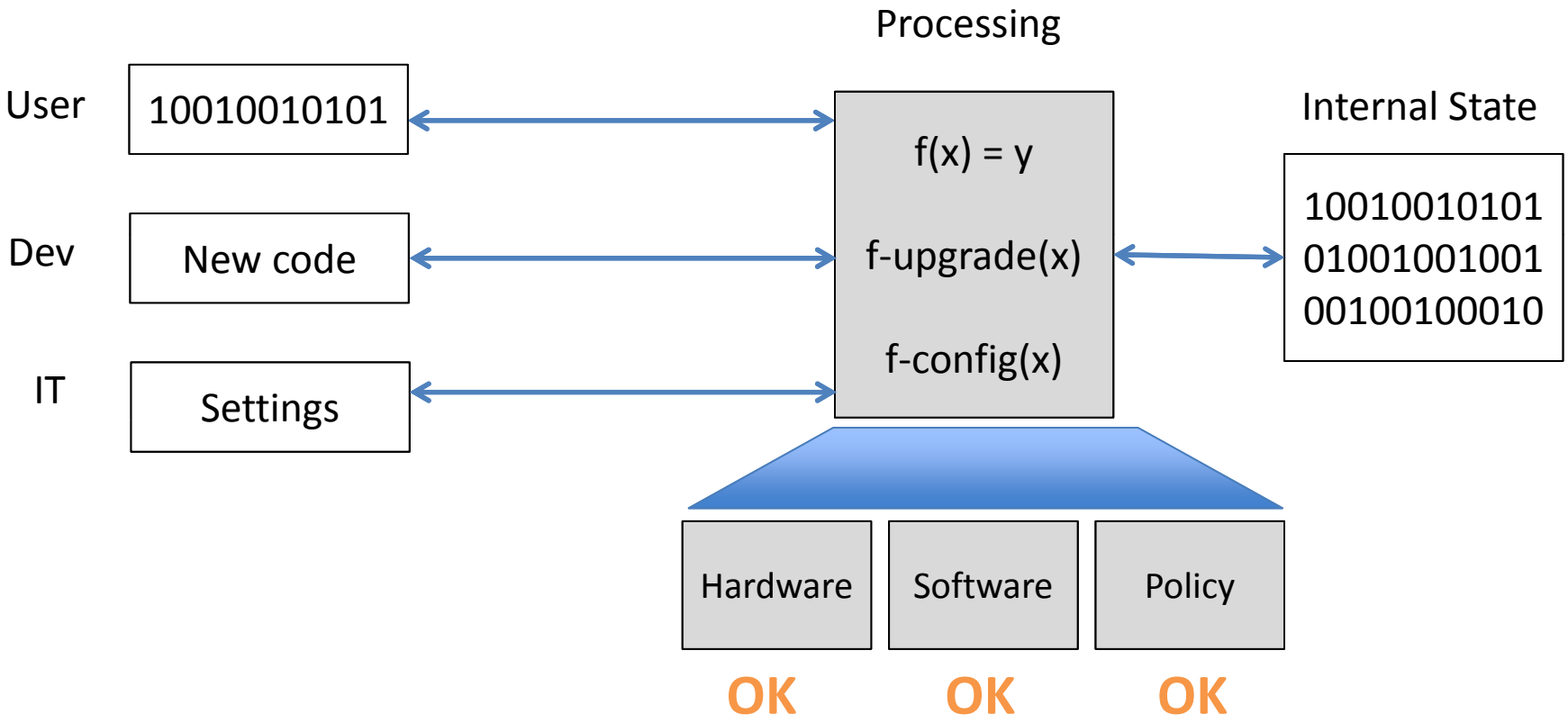
"These are lengths written by hardware, so will only be wrong if the hardware is broken. If the hardware is broken (or replaced by something malicious) then it can do anything it likes. Invalid values in ring entries are the least of your worries."

So how do we protect against such attacks?

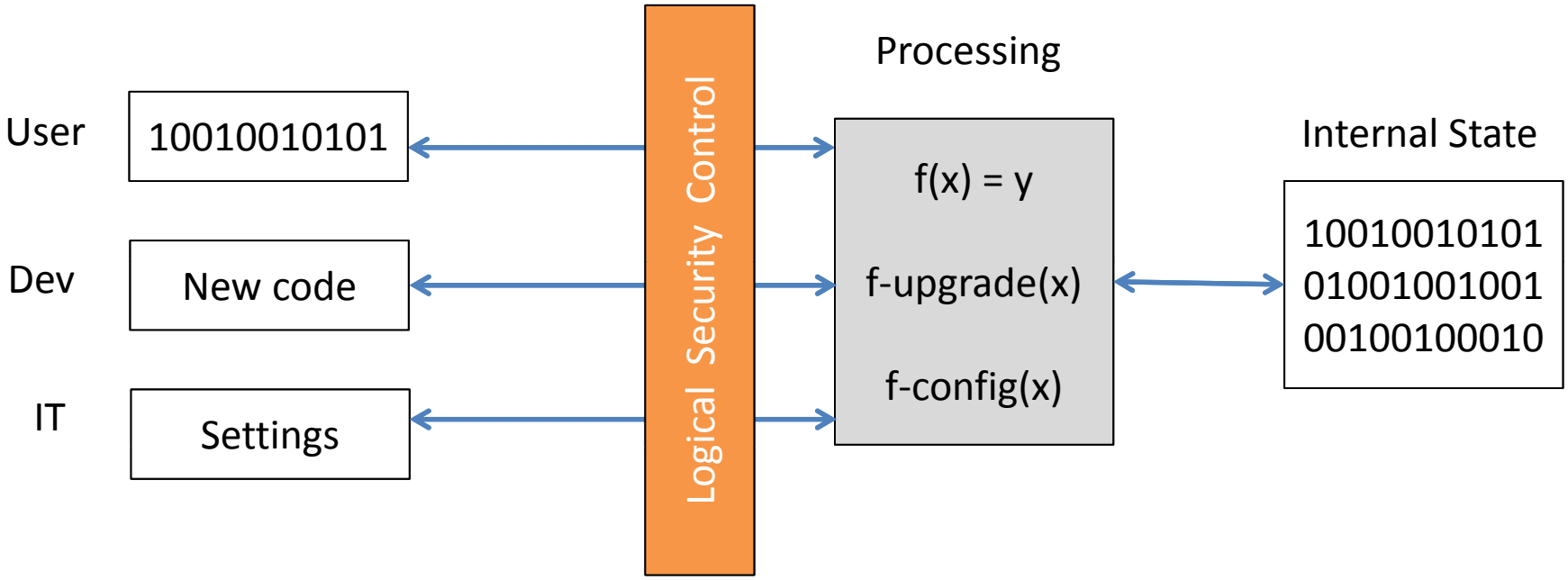
IT Security Job I: Prevent physical grab



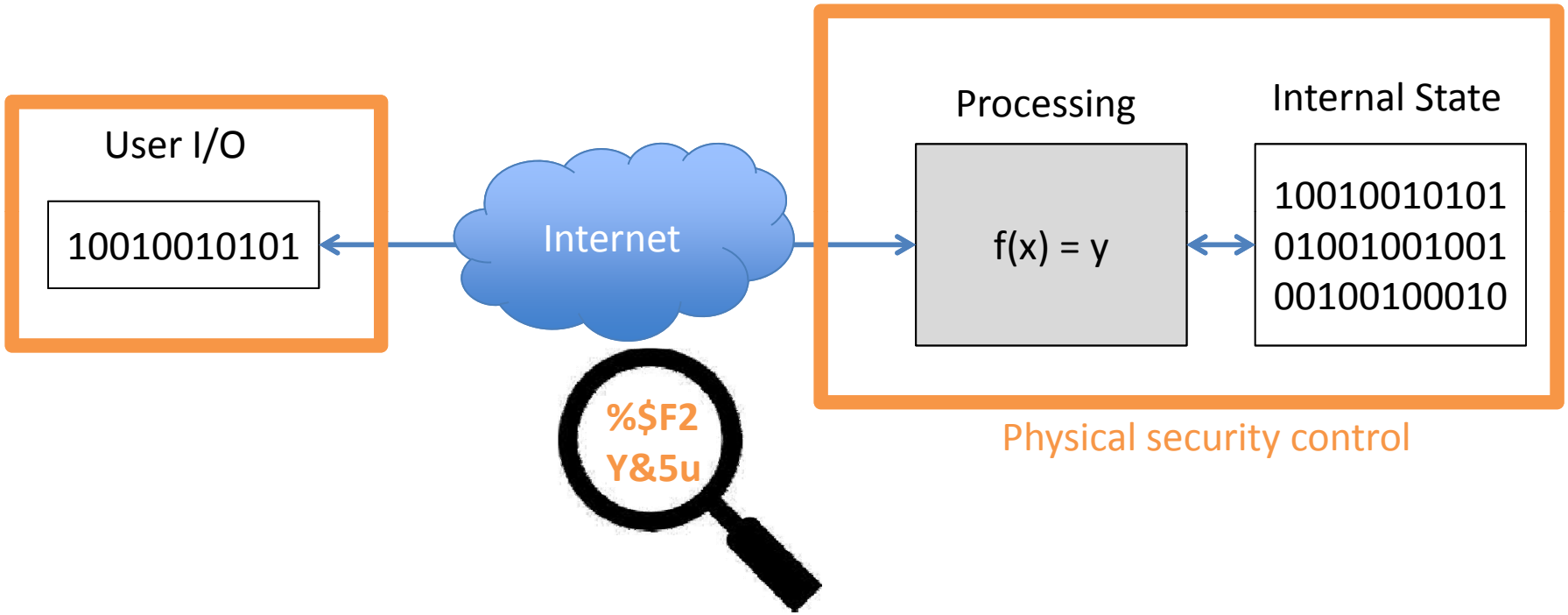
IT Security Job II: Check system integrity & lockdown



IT Security Job III: Secure logical access



IT Security Job IV: Encrypt public I/O



So how do we protect against such attacks?

Integrity attacks

SMM infection

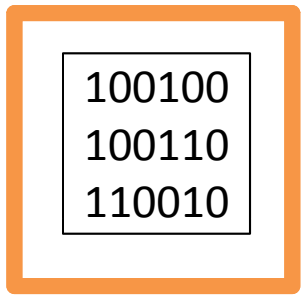
HDD firmware infection

injected kernel arguments

Physical attacks

Grabbing clear private SSH keys

Cold-boot



Physical security control

Encrypt elsewhere

Logical access attacks

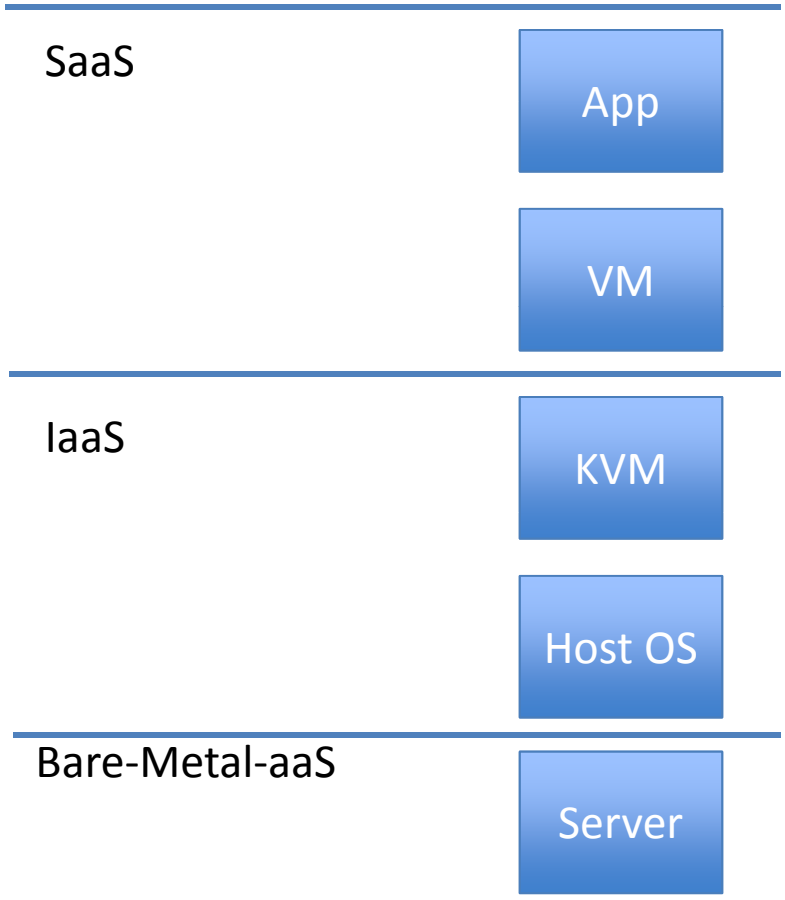
Inception

DMA capture of mysql records

Malicious device I/O

(IO-MMU)

The Cloud Challenge



How can a tenant verify integrity?

Who defines an “OK” stack?

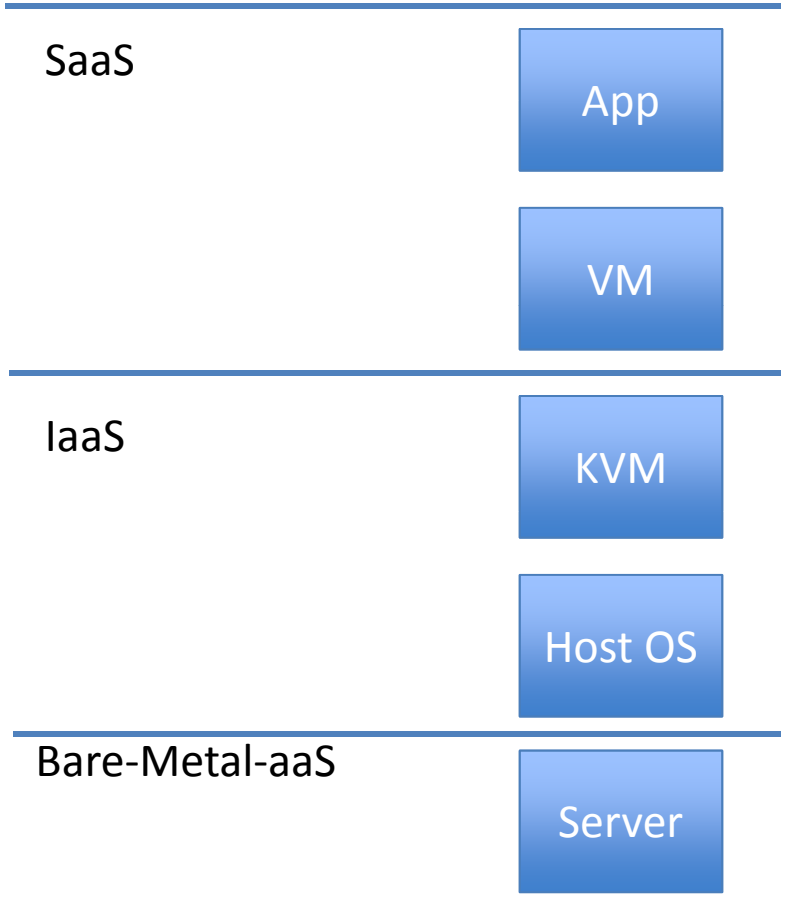
What’s a good physical perimeter?

- The data-center?
- Cage?
- Server?
- CPU?

(Encryption depends on the above question)

Should IaaS CSPs take more responsibility? Or give more control to customer?

Our mantra for secure IaaS (in x86 world)



1. Enable TPM & TXT

2. Choose a policy for hypervisor (i.e. “below the VM”) secure configuration. Tip: Consider stateless hypervisors.

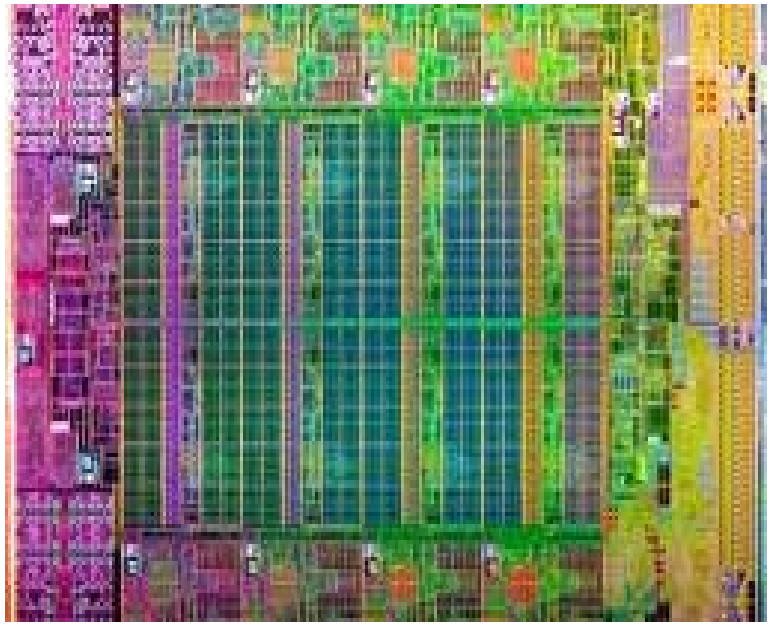
3. Verify than trust. Give no secrets to unverified systems

4. Decide on physical perimeter
   - Best – CPU
   - Good – The server
   - Risky – Data-center

5. Encrypt outside your chosen perimeter! (storage & network)

PrivateCore vCage Host

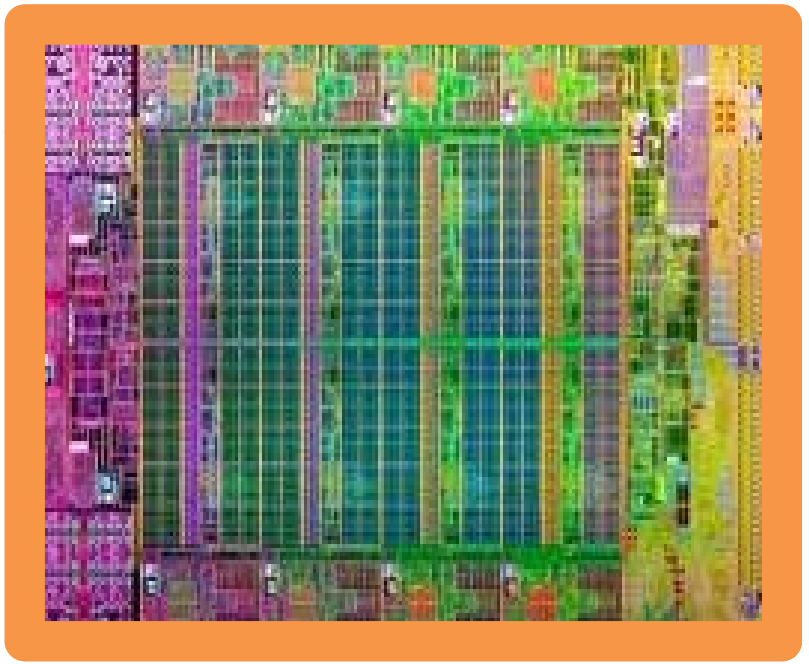
The CPU as the perimeter of computation



PrivateCore vCage Host

The CPU as the perimeter of computation

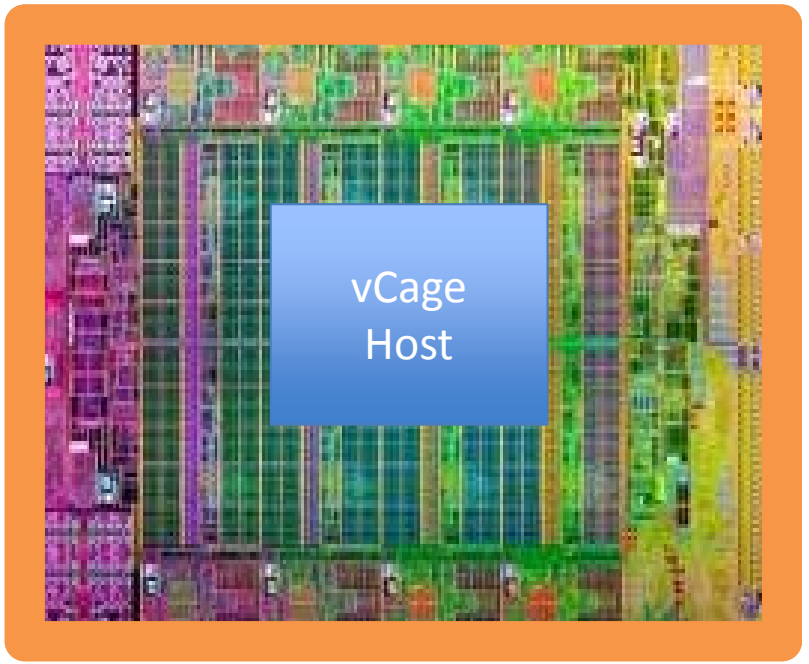
- Physical security is the CPU package itself



PrivateCore vCage Host

The CPU as the perimeter of computation

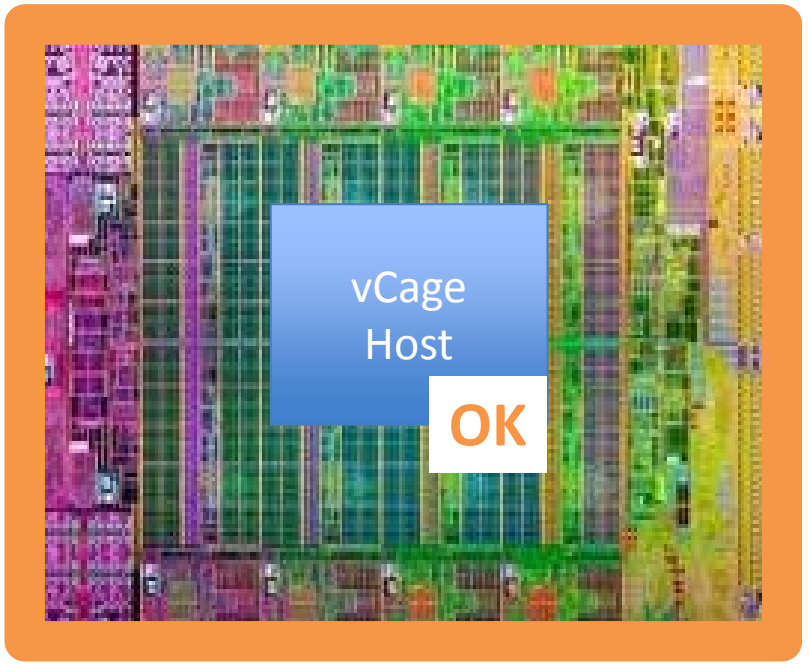
- Physical security is the CPU package itself
- Loading stateless image into CPU cache



PrivateCore vCage Host

The CPU as the perimeter of computation

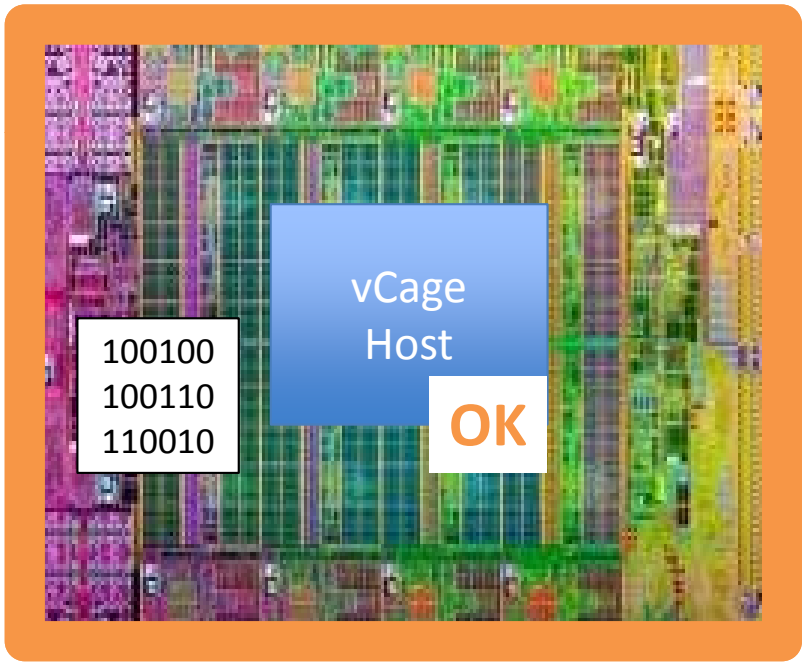
- Physical security is the CPU package itself
- Loading stateless image into CPU cache
- Test system integrity via Intel TXT



PrivateCore vCage Host

The CPU as the perimeter of computation

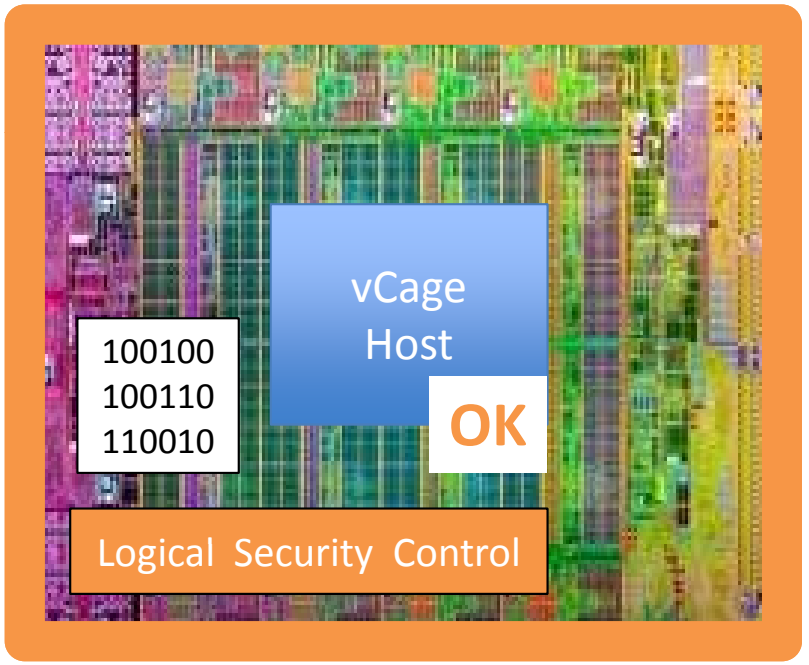
- Physical security is the CPU package itself
- Loading stateless image into CPU cache
- Test system integrity via Intel TXT
- Provision secrets (keys)



PrivateCore vCage Host

The CPU as the perimeter of computation

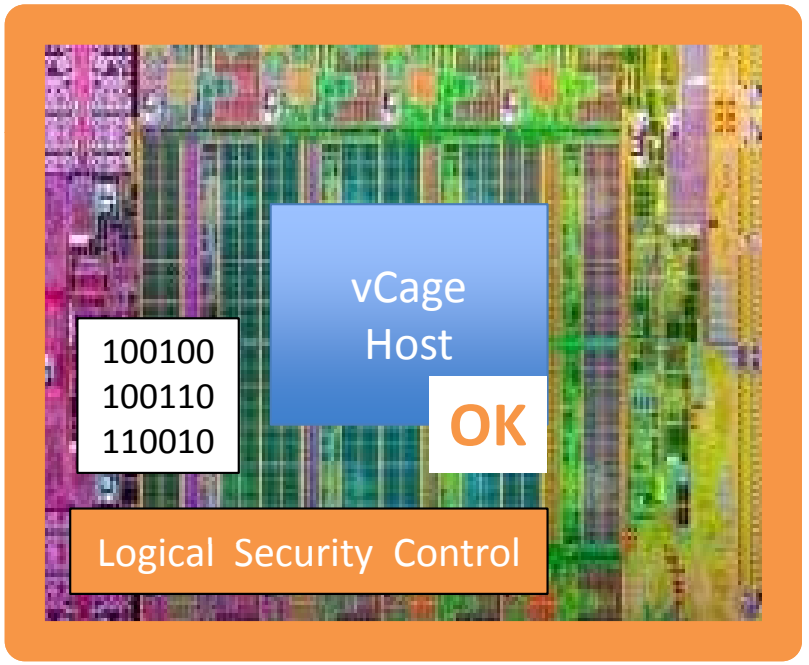
- Physical security is the CPU package itself
- Loading stateless image into CPU cache
- Test system integrity via Intel TXT
- Provision secrets (keys)
- Add logical security
  - DMA protection
  - Filter device IO

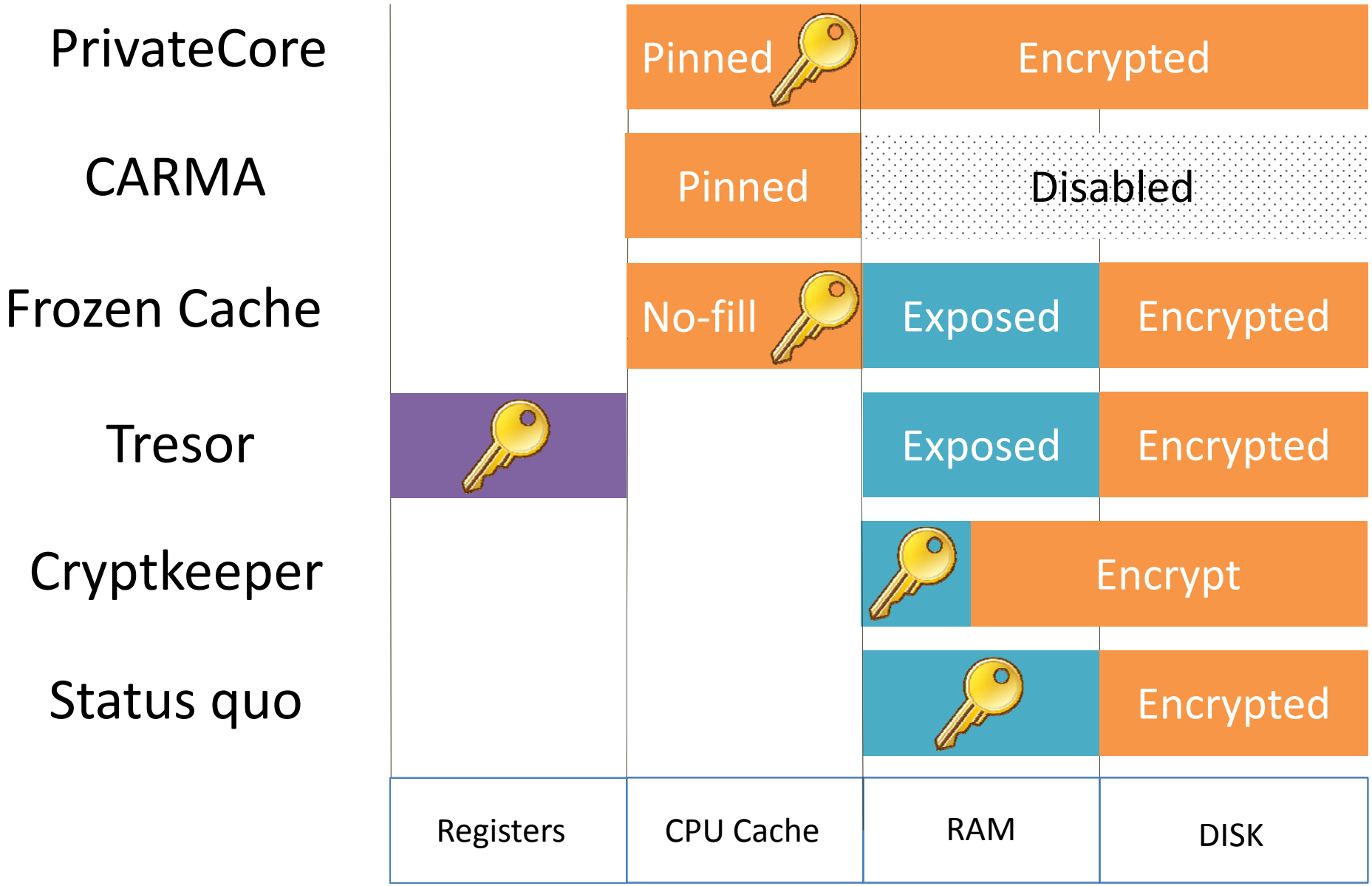


PrivateCore vCage Host

The CPU as the perimeter of computation

- Physical security is the CPU package itself
- Loading stateless image into CPU cache
- Test system integrity via Intel TXT
- Provision secrets (keys)
- Add logical security
  - DMA protection
  - Filter device IO
- Encrypt anything outside the CPU

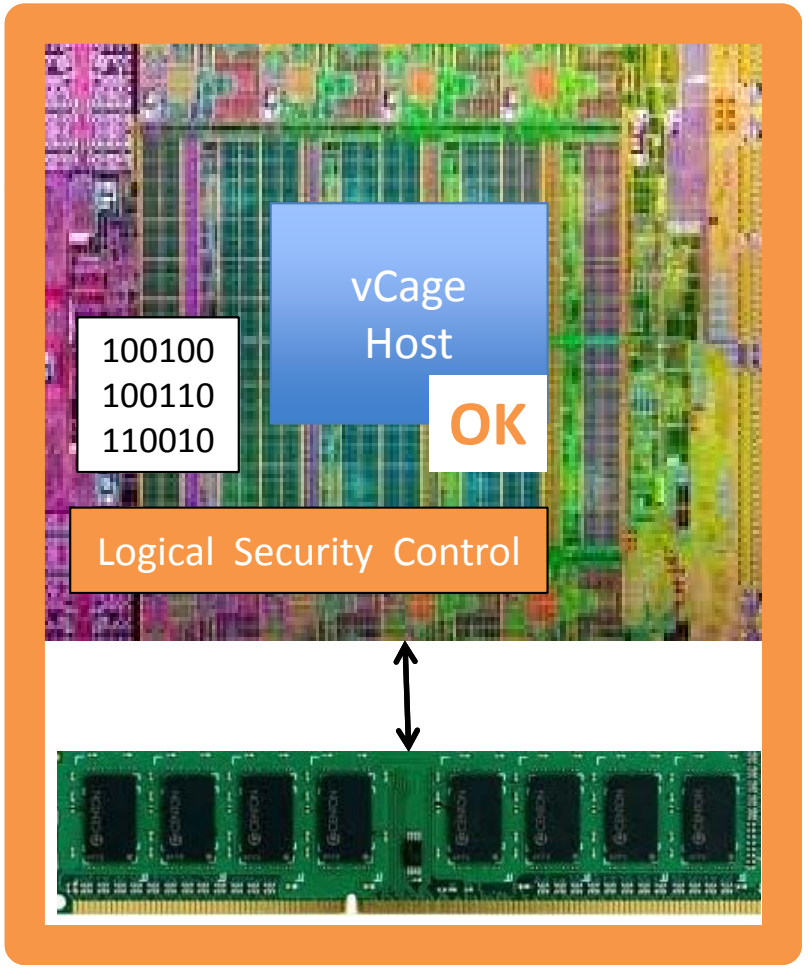



PrivateCore
CARMA
Frozen Cache
Tresor
Cryptkeeper
Status quo
Pinned
Encrypted
Pinned
Disabled
No-fill
Exposed
Encrypted
Exposed
Encrypted
Encrypt
Encrypted
Registers
CPU Cache
RAM
DISK

A reasonable performance tradeoff

The CPU **& DRAM** as the perimeter of computation

- Encrypt anything outside the CPU **& DRAM**
- Cons: Vulnerable to “cold-boot”, “malicious DIMM” & bus analysis
- Pro: High integrity without the performance penalties
- Ideal for public cloud environments

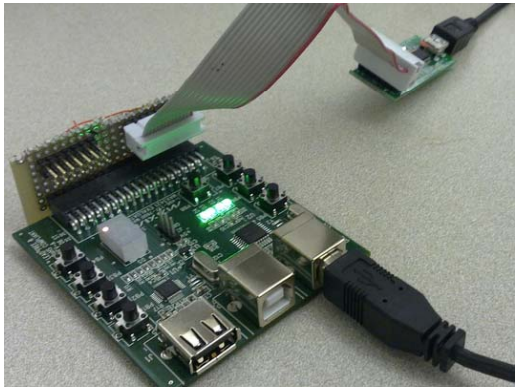


Biggest challenges

- Squeeze the Linux kernel into < 10MB while
  - Keeping all virtualization features
  - Keeping it stable (No OOM allowed)
- Keep CPU cache under our control
- Performance work
  - Squeeze different data structure to reduce working set
  - Identify new hot-paths in the kernel
  - Utilize AESNI capabilities

What's coming?

Offensive



Deeper down the stack we go!

Sniffing and MITM any bus

facedancer – USB hacking in python! 55$

Defensive

Intel SGX – A huge step toward CPU as physical perimeter

More Open Source software & hardware

Q & A

Oded Horovitz
oded@privatecore.com