Hacking: Hardware & People

Applied Information Security
Lecture 3

Recap: Foreknowledge

know your enemy.

- attacker mindset
- attack phases
- attacker tools

with few resources: injection attacks

- Dynamic Evaluation, Insecure Deserialization, XSS
- Command Injection, SQL Injection, Buffer Overflow

easy to perform,
easy to counter.

cheap, low-risk. but, what if injection attacks fail? do attackers just “go home”?

Today’s Topics

“next-level” (sophisticated) attacks.

costly: side-channel attacks

hard to counter.

- hardware
- network
- physical world (air-gap)

risky: social engineering attacks

hard to counter.

- weapons of Influence
- social engineering

attacks: side-channel

Sherlock Holmes

Sharing is a Vulnerability

attackers exploit sharing.

if A & B share E,
A can **observe/affect** B through E.

different attacks depending on what is shared

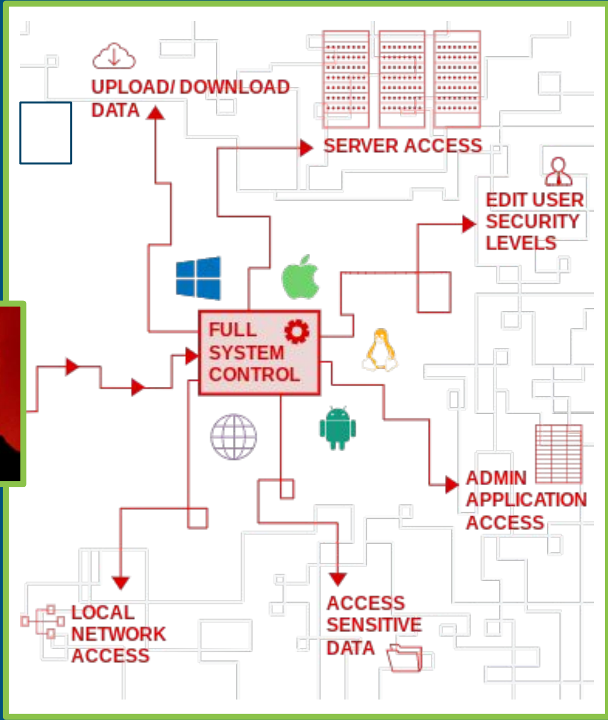
- hardware
- network
- physical world (air-gap)

fundamental tension:
Isolation (security) vs. sharing

"somebody toucha my spaghet!"

sharing:

hardware



Imitating the Ideal

ideal computer: infinite cores, infinite memory.

fake it

- OS multitasking *time-sharing*
- memory hierarchy *space-sharing*

important process requirement: *isolation*.
processes *share* resources.

isolation can be violated! (Unintended communication/interference)

How It Works

World

Sherlock

Can I use that resource?

No.

Why not?

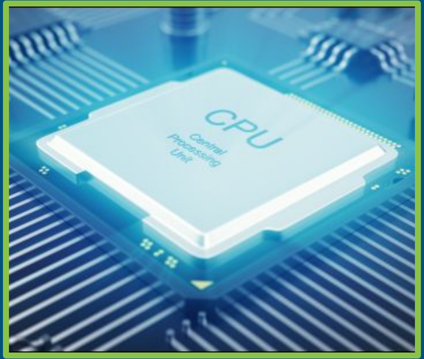
Bob’s process is using it.

Why is Bob’s process using it?

Because Bob’s process took the **then**-branch (not **else**) in procedure-

Aha! From this, I conclude…!

CPU



Processes run on same CPU.

OS manages processes;
implements process API.

- create, destroy
- wait, control (suspend/resume)
- status, …

Process states

- running, ready,
- blocked (e.g. waiting for disk IO)

(OS controlled via. CLI: *shell*)

CPU Timing Attacks

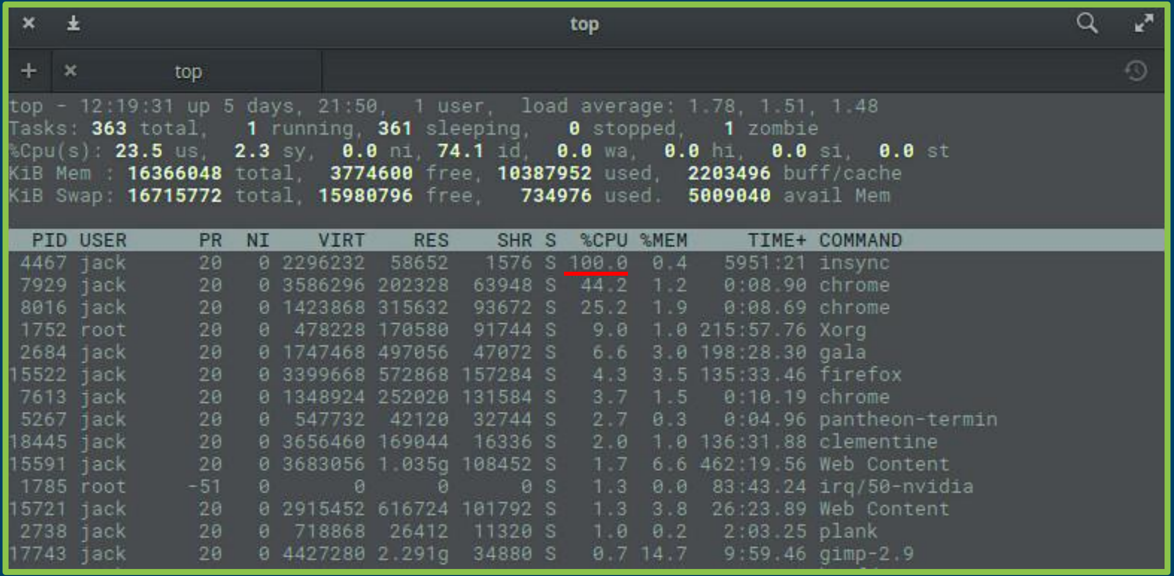
table of processes, task manager

Attack: Process A monitors the CPU load of Process B.

- High CPU load ⇒ 1
- Low CPU load ⇒ 0

Attack: Race conditions.

who writes to storage first



Storage

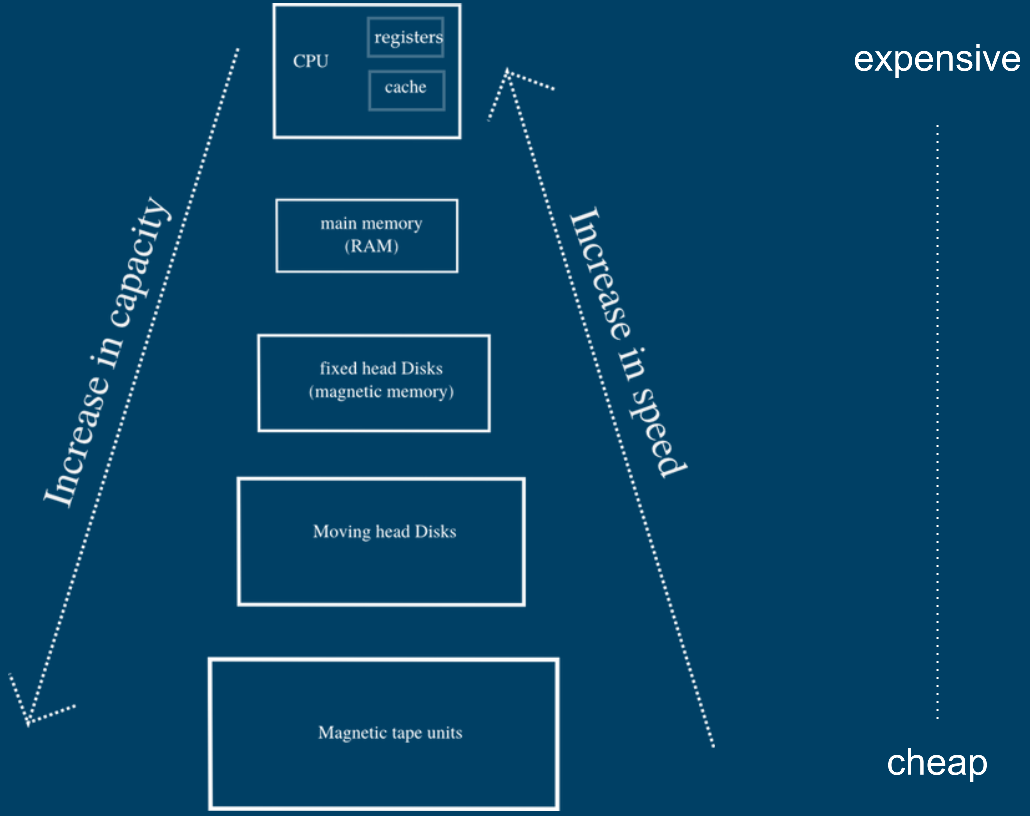
Processes come from same memory.

Attack: Communicate via. disk space.

- Add file (less space left) ⇒ 1
- Delete file ⇒ 0

Attack: Write to a shared log file.

Recap: Memory Hierarchy



Attacks: Side-Channel - Hardware - Storage

Latency Numbers Every Programmer Should Know

latency.txt Raw

```
Latency Comparison Numbers (~2012)
----------------------------------
L1 cache reference                           0.5 ns
Branch mispredict                            5   ns
L2 cache reference                           7   ns                      14x L1 cache
Mutex lock/unlock                           25   ns
Main memory reference                      100   ns                      20x L2 cache, 200x L1 cache
Compress 1K bytes with Zippy             3,000   ns        3 us
Send 1K bytes over 1 Gbps network       10,000   ns       10 us
Read 4K randomly from SSD*             150,000   ns      150 us          ~1GB/sec SSD
Read 1 MB sequentially from memory     250,000   ns      250 us
Round trip within same datacenter      500,000   ns      500 us
Read 1 MB sequentially from SSD*     1,000,000   ns    1,000 us    1 ms  ~1GB/sec SSD, 4X memory
Disk seek                           10,000,000   ns   10,000 us   10 ms  20x datacenter roundtrip
Read 1 MB sequentially from disk    20,000,000   ns   20,000 us   20 ms  80x memory, 20X SSD
Send packet CA->Netherlands->CA    150,000,000   ns  150,000 us  150 ms

Notes
-----
1 ns = 10^-9 seconds
1 us = 10^-6 seconds = 1,000 ns
1 ms = 10^-3 seconds = 1,000 us = 1,000,000 ns
```

orders of
magnitude!

Cache

Stores data “at hand”, so future requests to it are faster.

- Between RAM and CPU registers.

Analogy: You need a wrench.

1. Check your tool box
2. Check your work desk
3. Check your storage room

How a Cache Works

prefetching demo!

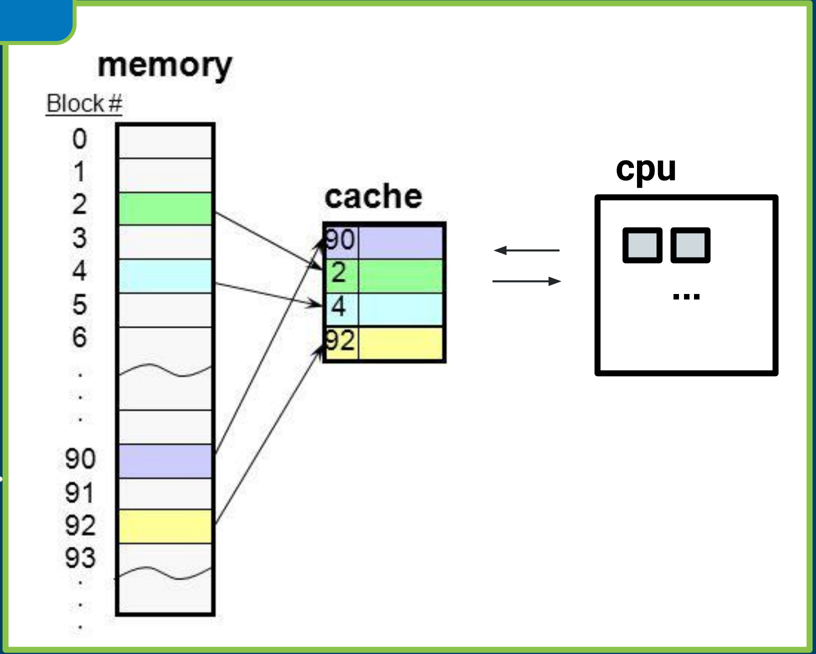
e.g. several array cells

Cache stores **blocks** of memory.

Scenario: CPU reads an address.

- **Hit:** (address in block in cache) 90
  Read from cache
- **Miss:** (address' block not in cache) 91
  Evict a block, Fetch the block (from mem).
  4

What to evict? What to do on write?
Different strategies (LRU, adjacent wr.).



How a Cache Works

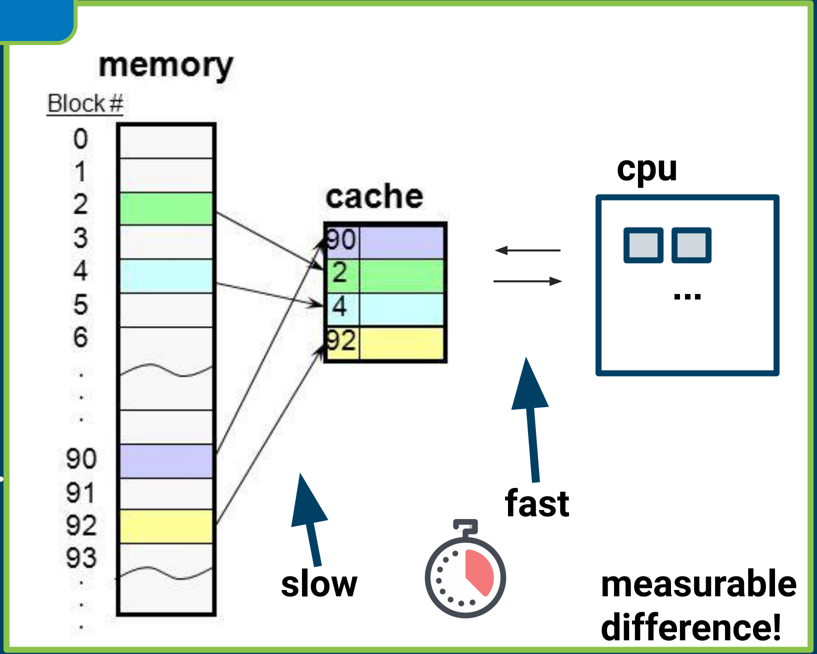
Cache stores blocks of memory.

e.g. several array cells

Scenario: CPU reads an address.

- **Hit:** (address in block in cache) 90
  Read from cache
- **Miss:** (address' block not in cache) 91
  Evict a block, Fetch the block (from mem).
  4

What to evict? What to do on write?
Different strategies (LRU, adjacent wr.).



small decrease hit rate ⇒ **large** increase latency

Processes Share a Cache

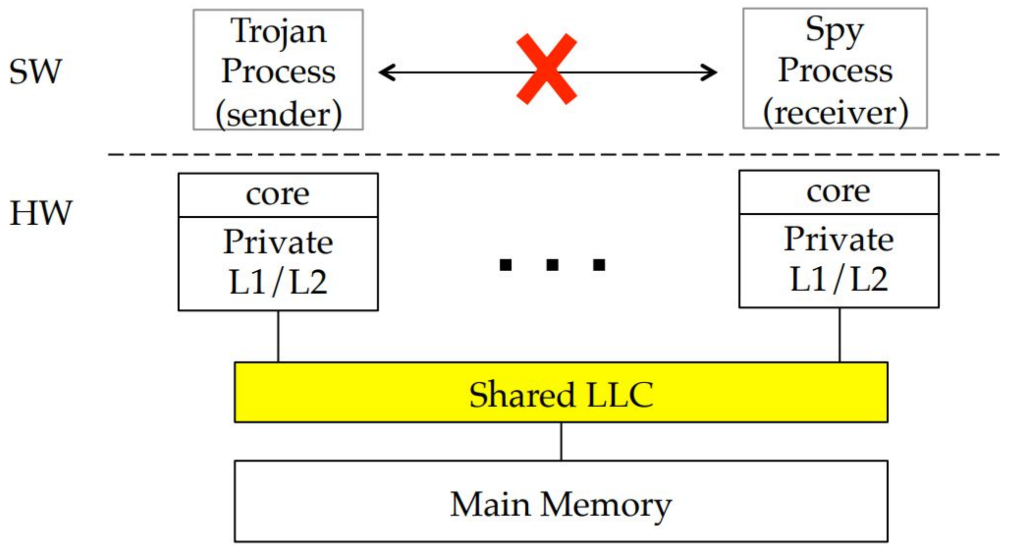


Fig. 3. LLC-based covert channel attack scenario.

different address space, though. How do they communicate?

Attacks: Side-Channel - Hardware- Cache

Cache Timing Attack: Prime+Probe

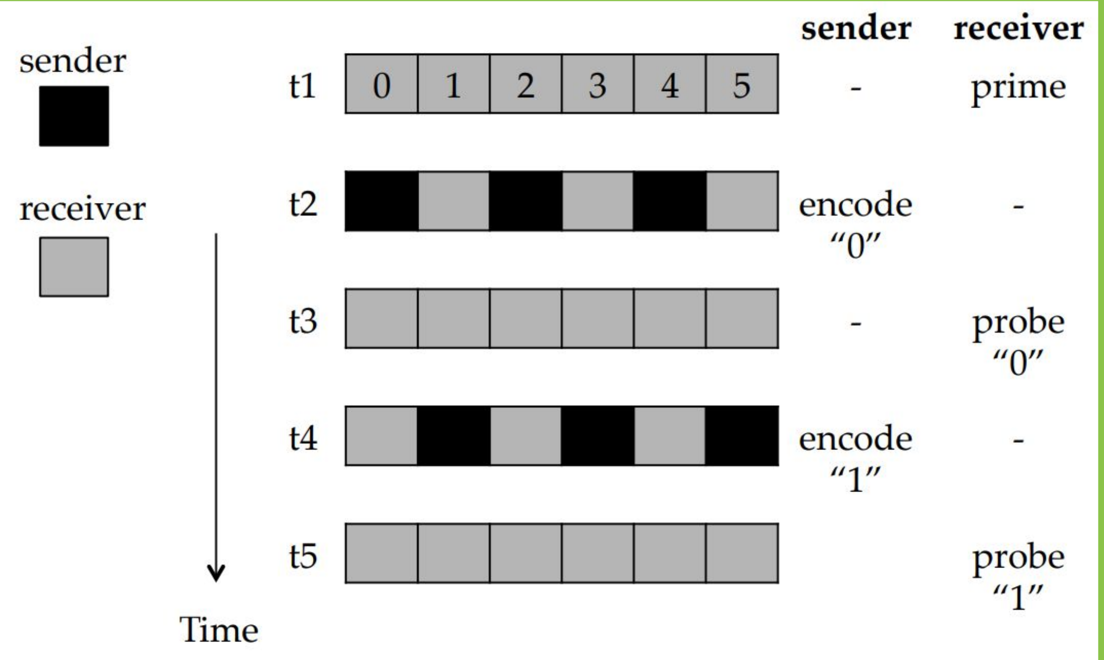


Fig. 1. Example of *Prime+Probe*.

estimage nr. of cache misses w/ a timer.

(remember memory hierarchy)

Cache Timing Attacks are Common

Platform:	Vulnerability:	Information leak (what / how):
Browser	Cache (Browser)	Browsing history [2]
Browser	FPU (CPU, Intel)	Cross-origin (pixel stealing) [3]
Browser	Cache (CPU, Intel)	System-wide mouse / network activity [4]
Android	Cache (CPU, ARM)	Cross-core (e.g. tap & swipe, keystrokes, ...) [5]

Lots more; Meltdown (cache, out-of-order) [6], Spectre (branch prediction) [7], ...

Fix, state-of-the-art: {Patch, Abandon**}** the affected system.
Hope such attacks won't happen again. But they will; **arms race**.

Attacks: Side-Channel - Hardware - Cache

Cache Timing Attacks, References

[2]: Timing Attacks on Web Privacy

[3]: On subnormal floating point and abnormal timing

[4]: The Spy in the Sandbox – Practical Cache Attacks in Javascript

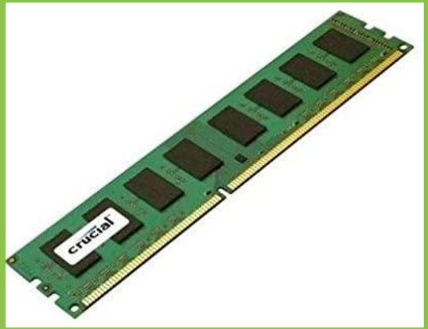
[5]: ARMageddon: Cache Attacks on Mobile Devices

[6]: Meltdown

[7]: Spectre Attacks: Exploiting Speculative Execution

Attacks: Side-Channel - Hardware

RAM



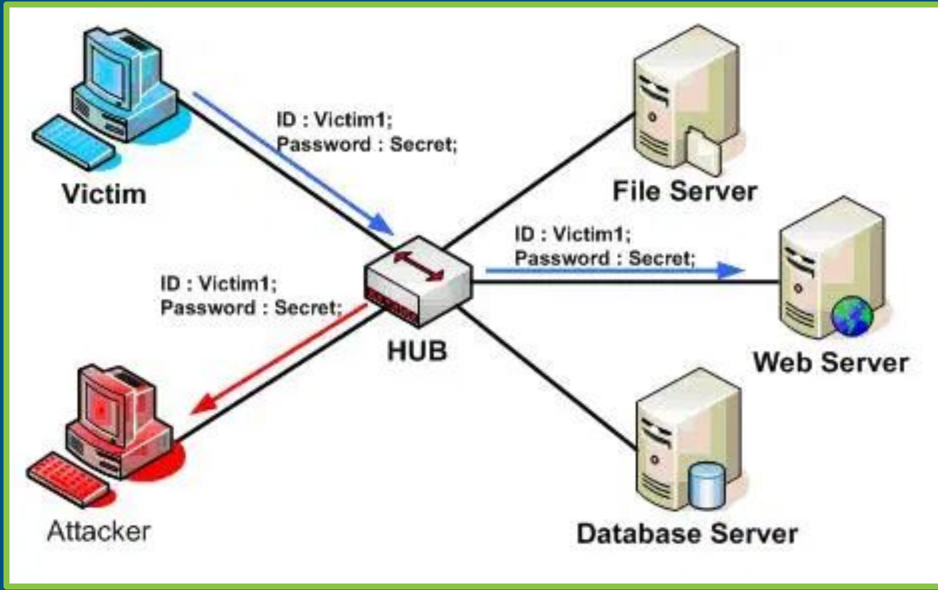
programs & data stored in main memory: RAM

attack (rowhammer): repeatedly accessing (i.e. hammering) a row in memory causes charge from adjacent rows, to leak into hammered row, thus flipping bits.

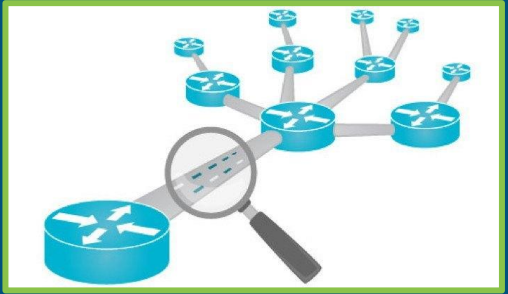
privilege bit ⇒ you are root

sharing:

network



Traffic Analysis



Deduce information from communication pattern (time).

De-anonymize. Attacks exist on:

- SSH ("Morse hand")
  "Timing Analysis of Keystrokes and Timing Attacks on SSH" (Song et al.)
- Tor
  "Low-Cost Traffic Analysis of Tor", S&P 2005
- All countermeasures
  "Why Efficient Traffic Analysis Countermeasures Fail", S&P 2012

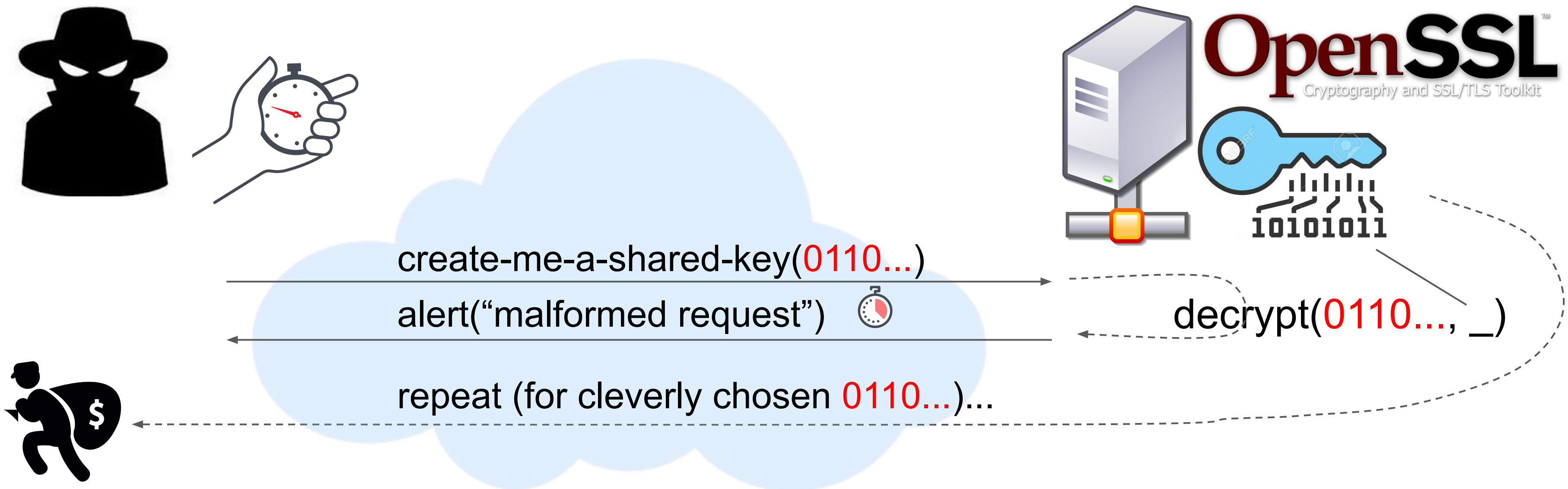
An overview (Willard's slides):

http://www.csc.kth.se/~buc/PPC/Slides/trafficanalysis.pdf

Attacks: Side-Channel - Network - Traffic Analysis

Remote Timing Attack: OpenSSL

[Brumley+05]



Fix: abandoned; use **TLS** instead

What can we do?

When processes **share**, leaks are **unavoidable**.

solution (?) : guarantee no sharing: **isolation**.

- prove that processes don’t share (IFC)
- remove sharing: partition
    - CPU time (round-robin)
    - HDD, RAM, Cache, …

the latter kills performance. uncommon;
choice between security and performance?
typically, performance chosen.

sharing:

real world

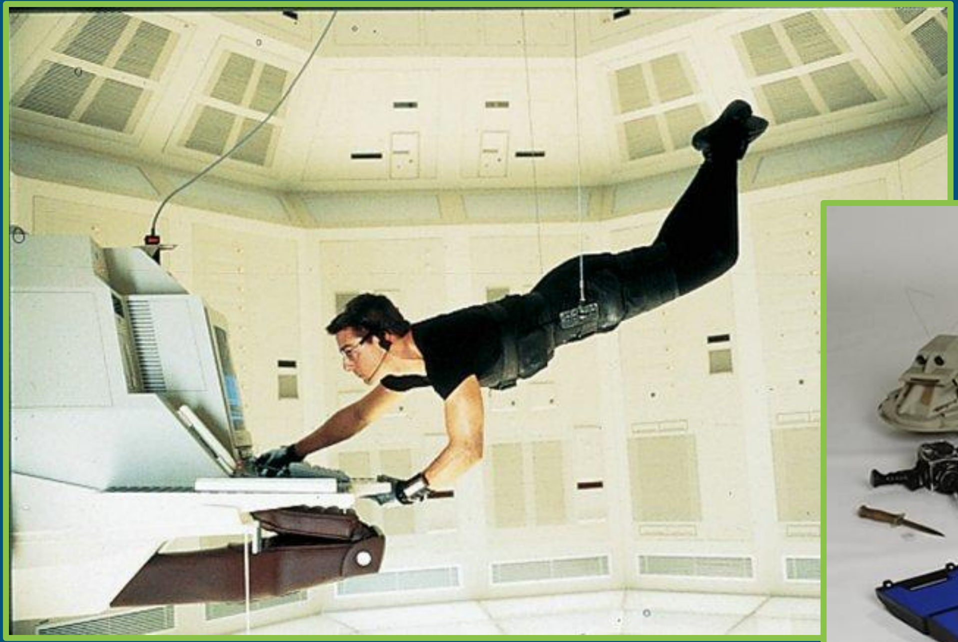
Attacks: Hardware

Attacks: Hardware

Breaching the **air-gap**

air-gap: system physically isolated from network.

how to breach (w/ enough **resources**)? time to get Hollywood-level creative.

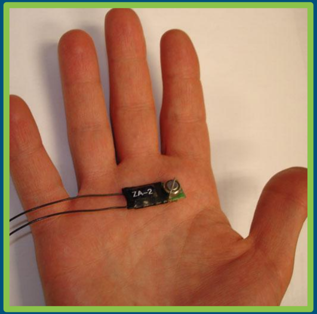


Attacks: Hardware

Can you get close?

sound

Acoustic Analysis



Computers emit sound

- Fans
- HDD
- Coil whine*

Often outside audible spectrum.

All controllable by SW!

Receiver:

- Smartphone, planted bug, …

Range: several m (10m for *).

4096-bit RSA key extraction

Acoustic Analysis Scenarios

Q5: What are some examples of attack scenarios?

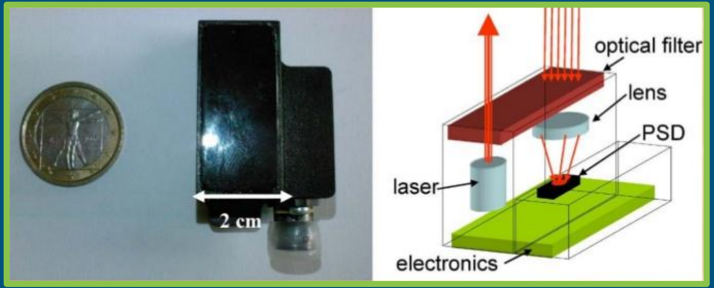
We discuss some prospective attacks in our paper. In a nutshell:

- Install an attack app on your phone. Set up a meeting with the victim and place the phone on the desk next to his laptop (see Q2).
- Break into the victim's phone, install the attack app, and wait until the victim inadvertently places his phone next to the target laptop.
- Construct a web page use the microphone of the computer running the browser (using Flash or HTML Media Capture, under some excuse such as VoIP chat). When the user permits the microphone access, use it to steal the user's secret key.

Adi Shamir et al. https://www.cs.tau.ac.il/~tromer/acoustic/

light

Optical Analysis



Computers emit light.

- HDD indicator
  - Malware on PC precisely controls the indicator. Optically sense the indicator from a distance.
- Transistor state change
  - Emits photons. Can be read through picosecond imaging analysis. Shown to recover AES secret keys.

heat

Thermal Analysis

Computers gives off heat.

Computers have thermostats

- monitor overheating

Two nearby computer communicate by monitoring each other’s heat.

Range: 40 cm (in a demo)

electromagnetic emissions

Van Eck Phreaking

Computers give off EM emission.

- keyboards, monitors, printers, …

Readable from 10s-100s m away (!)

- Affordable (max $2k equipment)
- Known since the 1950s (!)
  - TEMPEST NSA standard 1958
    - WW2, Bell Telephone discovered attack on its SIGCUM machine.
  - Korean war

Dutch gov. banned electronic voting.

Defense: **Faraday Cage**

Attacks: Hardware

Van Eck Phreaking

Reading a display over an air-gap

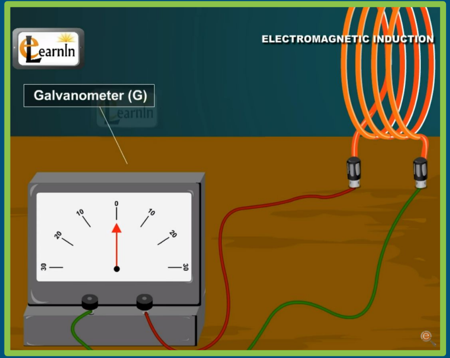
BBC report, February 1985

https://www.youtube.com/watch?v=mcV6izFG3vQ (01:29 - 03:23).

Attacks: Hardware

Can you get *real* close?

Electromagnetic Analysis



Chips give off EM emissions.

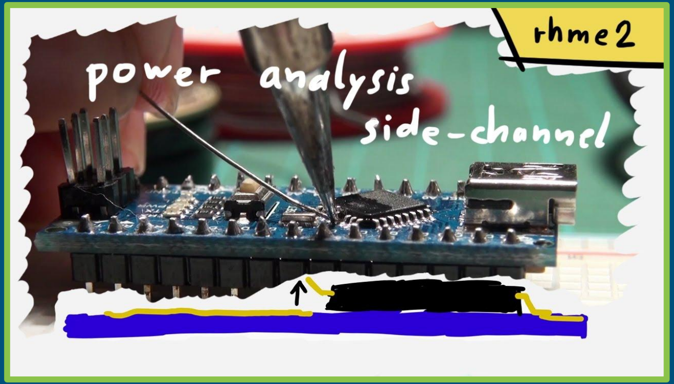
- Smart cards, PCs, smart phones

Readable a few m away.

- User's signing key (OpenSSL)

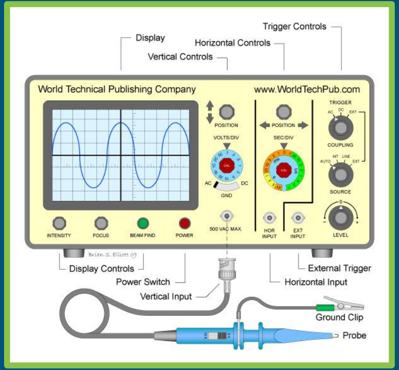
Successful against crypto HW that perform different operations based on the data currently being processed

Attacks: Hardware



Can you *touch* it?

Power Analysis



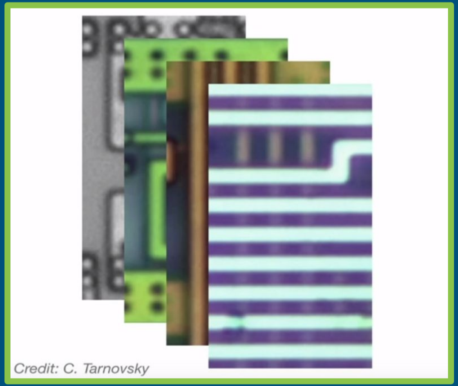
Analyze power consumption of cryptographic hardware (e.g. TPM)

Can infer:

- branching behavior
- operands
  (so even constant-time implementations are vulnerable)

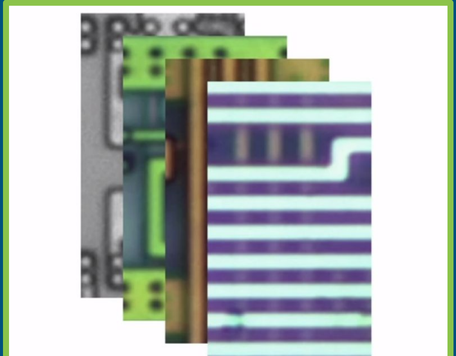
Extract cryptographic keys
(TPM. Exploit on Microsoft BitLocker 2010)

Attacks: Hardware



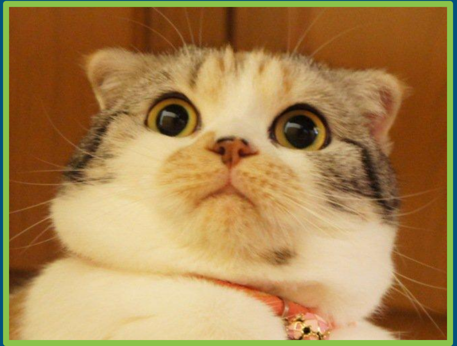
Can you *enter the silicon*?

Attacks: Hardware

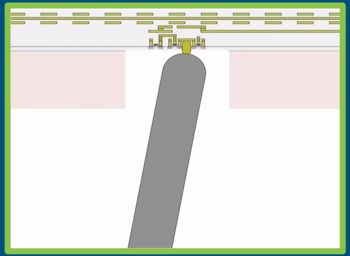


Credit: C. Tarnovsky

Can you *enter the silicon*?



IC Analysis



Hardware reverse-engineering

- Image the chip (5, 8, 15 layers)
- Rebuild logical function of chip
- Probe signals *inside the chip* (milling, probing)

Bypass physical protection.
Read encryption keys in the plain.

"data is in the chip" ⇏ data cannot be extracted

Example: Cloned Canal+ smart cards

Breaking-and-Entering through the Silicon

Dmitry Nedospasov hacks circuits

“Security of the IC Backside”, talk at Chaos Communication Congress 2013

https://www.youtube.com/watch?v=Gt6VyuLZBww (42:00 - 43:06).

Attacks: Hardware

What can we do?

SW cannot secure against physical access.

Physical security to prevent physical access.
Interference to prevent short-range reading.

- Shielding (Faraday Cage)
- Jamming (White noise)
- Security analysis SW
  (check for these vulnerabilities)

… or bury your stuff real deep underground.

Fortunately, very few of us have data that is worth all this effort…

attacks:
social engineering

Did you ever agree to something,
without understanding why?

Attacks: Social

You are a Vulnerability

Attackers exploit **people**. How:

- Weapons of Influence
- Social Engineering

Most commonly exploited vulnerability.

- Easy to exploit
- Hard to counter

Frank Abagnale posed as Pan-Am pilot; used social engineering to con his way to $Ms in luxury.

(movie: “Catch Me If You Can”, starring Leonardo DiCaprio)

Attacks: Social

Frank becomes a Pilot

https://www.youtube.com/watch?v=5IxKXFHSSKA

weapons of influence

- reciprocation
- commitment, consistency
- social proof
- liking
- authority
- scarcity

Attacks: Social - Weapons of Influence

Principle	Description	Exploit
reciprocation	We repay in kind.	Gift in exchange for sensitive information
commitment, consistency	Once we commit, we stick to it.	Pose as auditor. Get him to commit to answer your questions.
social proof	We do as others do.	Social network account, friend her friends, then friend-request her. Friend-of-friends...
liking	We say yes to those we know and like.	Be charming and personable.
authority	We respect authority.	Forge letters of authority.
scarcity	We desire that which is rare.	“Time is of the essence”

social engineering

kinds of attacks:

- physical
  - dumpster diving, theft
- social
  - weapons of influence
  - curiosity
- technical
  - Google search, Maltego
- socio-technical
  - baiting (USB)

Taxonomy

Channel

- e-mail, IM, voice (voice: phone, VoIP)
- social networks (fake ID, hide & harvest)
- cloud (shared directory)
- website (waterhole)

Operator

- human
- software

Attacks

- phishing
- dumpster diving
- shoulder surfing
- reverse social engineering
- waterholing
- baiting
- tailgating

A lot of interaction virtual ⇒ bigger attack surface.
spearphishing (impersonate others, learn behavior), social network, cloud service, blogs & wikis, mobiles

Summary

Summary

Shifting Focus

attackers
perpetuate
attacks
that exploit
vulnerabilities
to inflict
harm

Summary

Shifting Focus

attackers
perpetuate
attacks
that exploit
vulnerabilities
to inflict
harm

focus
so far

Summary

Shifting Focus

rest of course!

countermeasures control

attackers
perpetuate
attacks
that exploit
~~vulnerabilities~~
to inflict
harm

focus so far

Summary

Shifting Focus

rest of course!

countermeasures control

attackers
perpetuate
attacks
that exploit
~~**vulnerabilities**~~
to inflict
harm

focus so far

spoiler: cannot make it “secure”.
what we **can** do

- transform (blind) trust into trustworthiness
- make attacks unaffordable

We understand attacks (threats).
Now, *what are we going to do about them?*