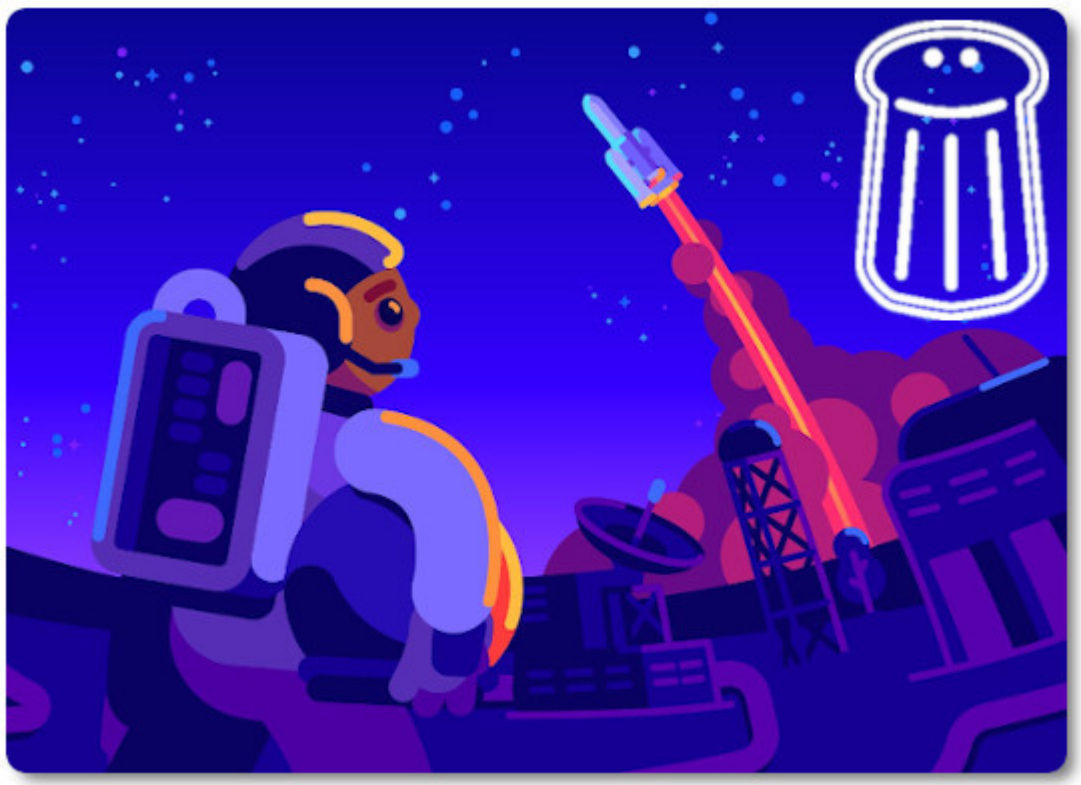
Tackling security issues in virtualization



© Kurzgesagt

About me: Olivier Lambert

- Vates CEO/co-founder
- Former sysadmin
- *Xen* user (since 2006)
- *Xen Orchestra* (2013)
- *XCP-ng*, a Citrix *XenServer* fork (2018)
    - now hosted in *Xen* Project (Linux Foundation)!

Slides available at https://vates.fr/pts

Virtualization, still everywhere?

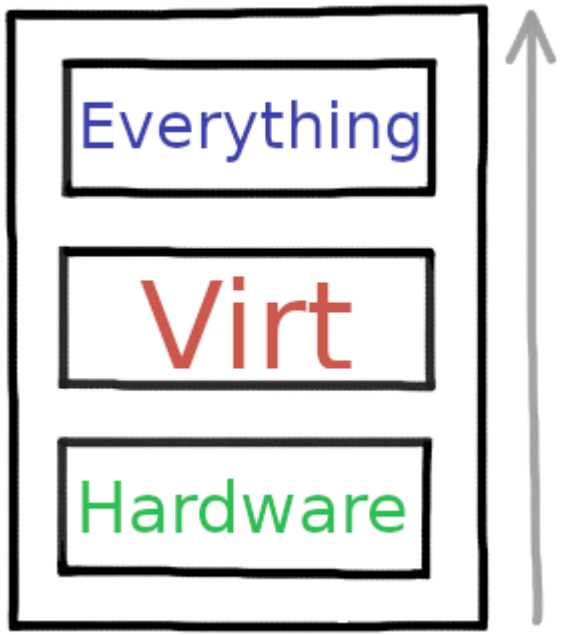
I've heard:

- "It's a thing from the past" (IBM S/360 to be fair)
- "Everybody is in the Cloud"
- Magic words: "Kubernetes", "Serverless", "SaaS/PaaS/IaaS"

This is a good sign.

It means virtualization is ubiquitous.

Also meaning we'll face new challenges.



New challenges?

> "Whole stack" security awareness

> Specific CPU security mitigations

> Security in new virtualization use cases

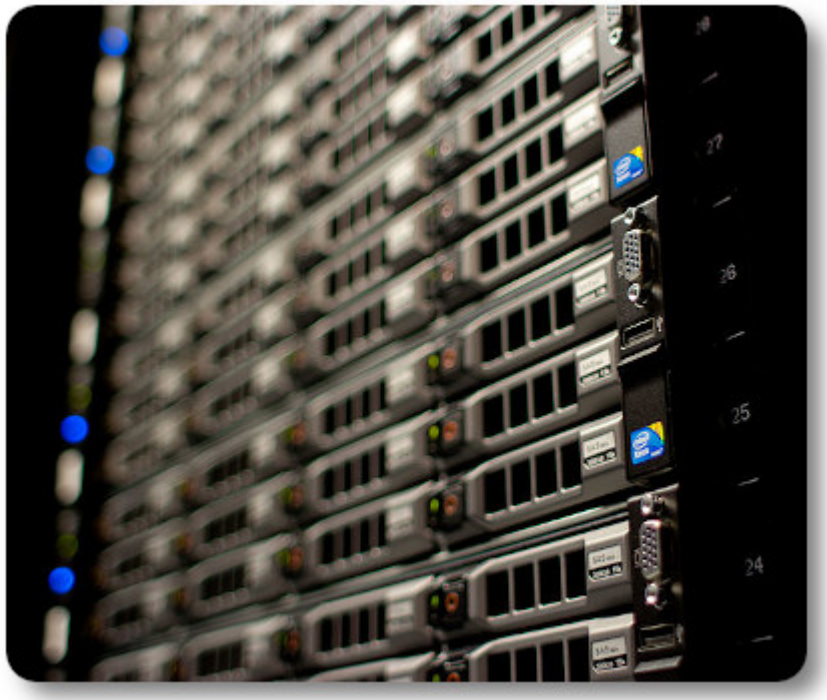
Whole stack security in question

We forgot "layer 1"

Years of trends further away from the metal:

1. Virtualization
2. First public cloud
3. Virt. orchestration (PaaS/IaaS)
4. Containers
5. Container orchestration (k8s)
6. Serverless
7. #Developerless ("No-code")

What's next?



Whole stack security in question

In the mean time...

A big shift for hardware vendors, software is growing in their offer:

- more firmware
- BMCs with more features (Baseboard Management Controller)

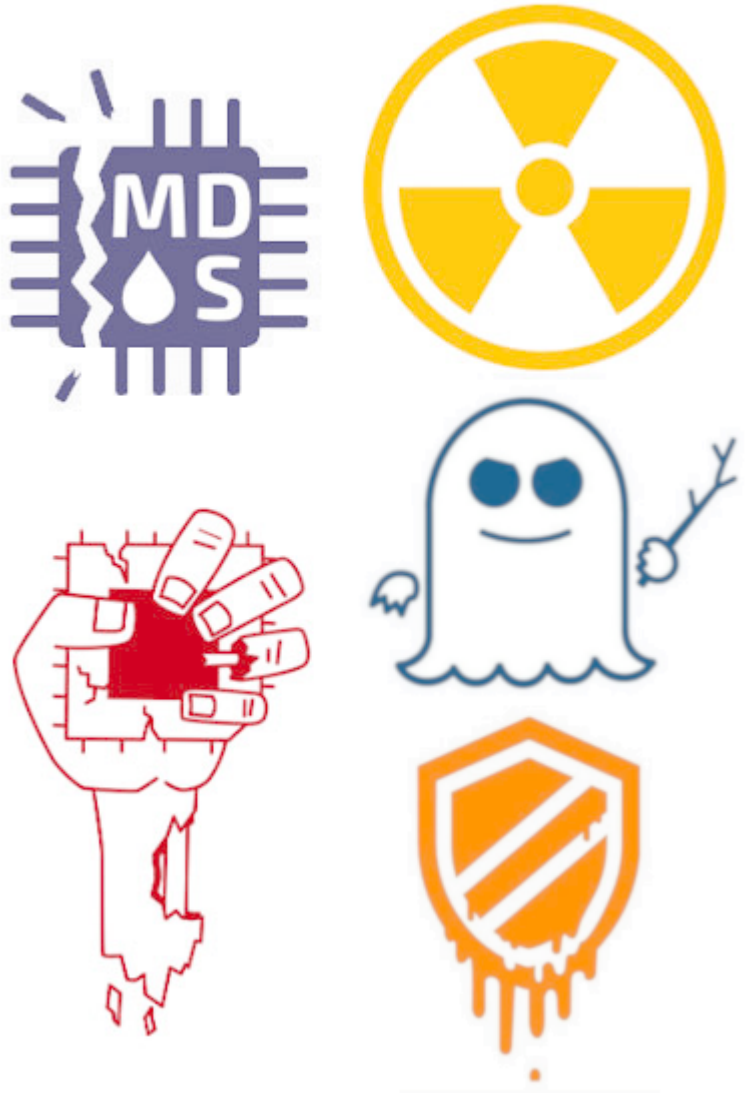
From teams with other priorities than security.

CPU vulnerabilities

Right in the silicon, due to branch-prediction leak and injection:

- Meltdown
- Spectre
- Foreshadow
- ZombieLoad
- CacheOut
- LVI

And all their respective variants.



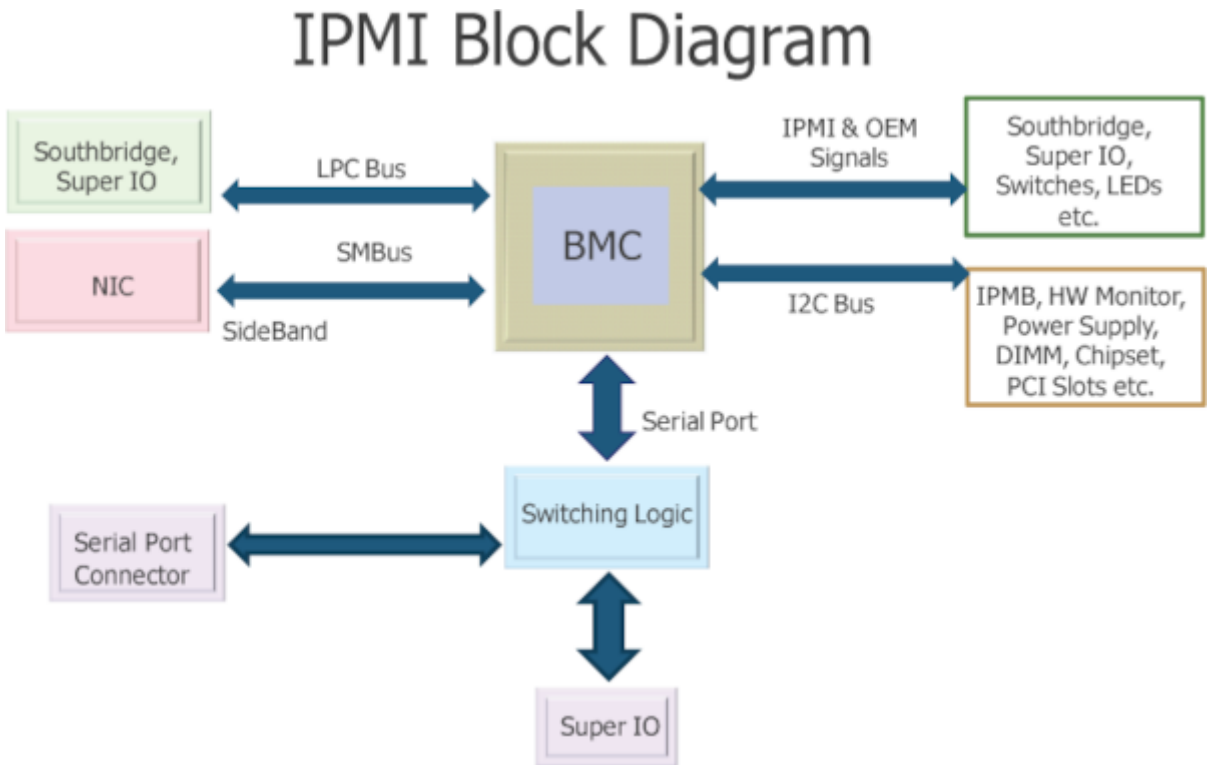
IPMI/BMC vulnerabilities

Extra chip with its own CPU/net, allowing:

- power cycle
- update firmware
- mount virtual devices
- UEFI changes…

Vulnerabilities found:

- escalation of privilege
- denial of service
- information disclosure



So what?

We "realized" that:

- "Low-level" software must be updated often (microcode, firmware, BMCs)
- We might disable HT and/or enable mitigations that can hurt perfs
- Side channels and hardware bugs are here to stay
- Security as a whole: "all layers" matters

What's the impact on virtualization?

Security as a whole

What about security and perf together?

- Mitigations cost is high (SMT/HT disabled, etc.)
- Perf/risk ratio (leave the choice)
- Keep virt overhead low

Compute side

- Rethink/refactor parts of hypervisor regarding:
  - CPUs isolation/sec features (SME/SEV,SGX)
  - "Think big" for massive CPUs (EPYC Rome, NUMA impact…)
  - How to expose this complexity to the end user?

New use cases

- Embedded world
  - Eg: Automotive projects, Aerospace, etc.
  - Compliance and security challenge
- Edge computing: survive in hostile environment ("on the field")
- New architectures (RISC-V, Power9)
- More ARM deployments (eg: Graviton 2)

Security challenges for virtualization

3 ways to face these security challenges:

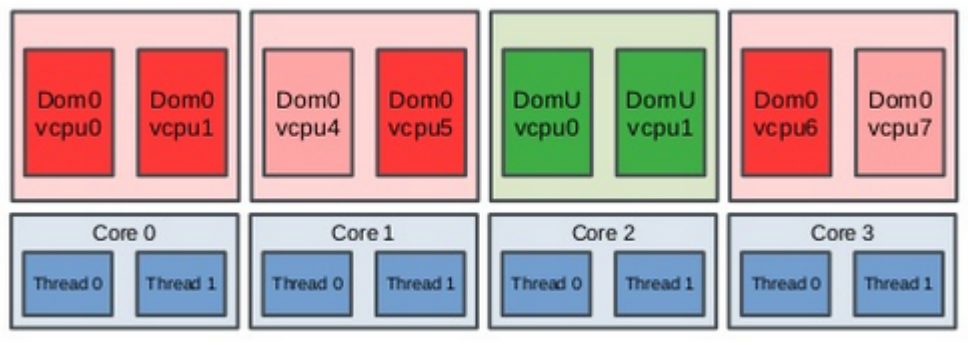
1. Technical solutions
2. Organizational solutions
3. Integration solutions

Technical solutions: flexibility and painless updates

- Modularize the code:
  - mitigations enabled or not
  - extra features or not
  - stripped down version for maximum security (eg: OpenXT, QubeOS)
  - allow a full spectrum of security/perf
  - guide people (document it!)
- Easier updates means more security:
  - apply µcode updates without reboots (*late µCode loading* in Xen)
  - hypervisor live patching
  - hypervisor live upgrade

Technical solutions: R&D

- More research efforts, new ideas:
  - branch hardening
  - core scheduling (avoid side channel attacks)
  - better isolation
  - disaggregation
  - secret-free hypervisor (no global stack, no direct map…)
  - use latest hardware sec features



Organizational: process improvements

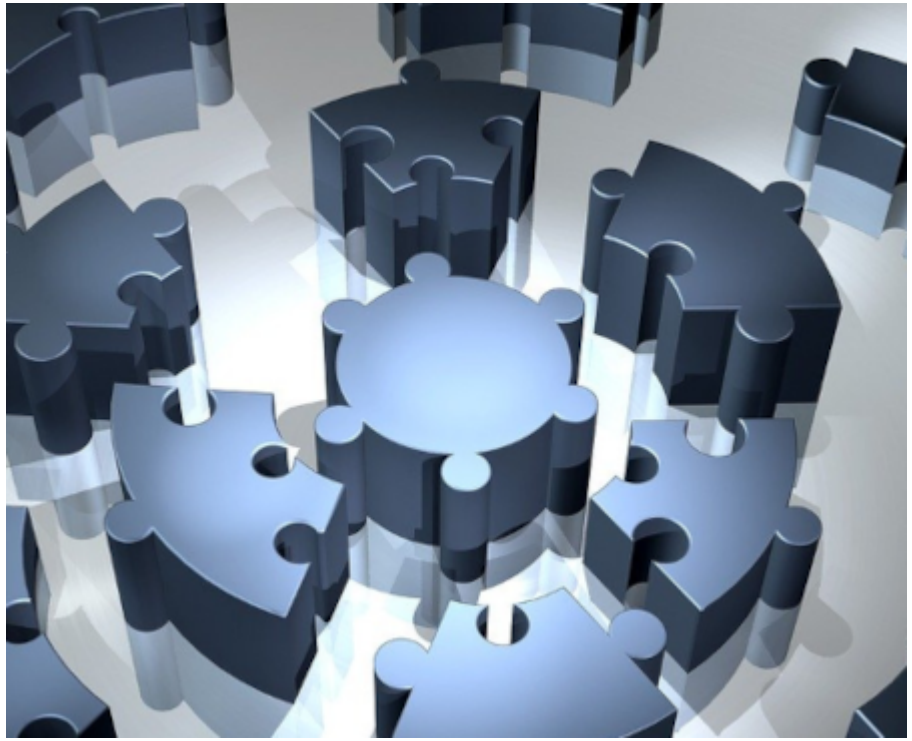
React fast, and adapt to cope with unpredictable vulnerabilities:

- A solid security workflow (eg Xen Security Process)
- Using community superpowers:
  - meetings/summit with devs
  - discuss with people on the field (end-users)
  - exchange with similar projects (eg KVM/Xen/Linux kernel)
  - include "external people" (eg coming from sec world)

Integration is key

Building an integrated product matters:

- Federate projects
  - community & ecosystem
  - organizational solution
- Secure by default
  - default platform choice but flexible
  - easy config/painless updates
- Strong R&D focus on sec & perf
  - find new technical solutions
  - deliver them "turnkey"



Integration example

XCP-ng:

- Xen distro working "out-of-the-box"
- Bundled with various projects (Xen/Linux/OVS…)
- Powerful API (compute/net/storage)
- Fully Open Source

Xen Orchestra:

- Web/CLI capable tool to manage XCP-ng
- Expose sec options but also backup etc

Our work

- Community
  - Build bridges between various communities (devs/end-users)
  - Federating various projects (inside/outside)
- Secure by default
  - Full distro control to make choices
  - Exposing features in GUI
- Intense R&D
  - guest and host secure boot
  - compute/net/storage perfs
  - exploring other arch

Conclusion

Virtualization will:

- see more challenges than ever (security and complexity on physical layer)
- need to keep the pace, requiring large collaborations AND research efforts
- still be relevant as long as the overhead is contained
- deliver integrated solutions to face those challenges
- see a new cycle of trends
  - innovation and competitors will push in the CPUs/hardware/low-level software
  - great opportunities for virt projects by working on hardware/software interface

Questions?