Br Bromium®

Hypervisor Security : Lessons Learned

Evolving hypervisor design in the quest for better security

Ian Pratt

Bromium

Hypervisor Genealogy

- 2001 **Xen** / XenServer
- 2008 **XenClient** / OpenXT
- 2011 hXen / **µXen** / Type-1.5
- 2016 **AX**

Xen and the Art of Virtualization

- Developed to support XenoServers project
  - Run arbitrary apps for fee on shared infrastructure – VMs enable containerization
- Requirements
  - Robust spatial and temporal isolation of VMs
- Design
  - Avoid complex and slow binary emulation or translation (no VT-x/AMDV)
  - Port kernel to a paravirtual API supported by hypervisor (Linux, XP/2003, *BSD)
- Use x86 segmentation to protect hypervisor from guest
  - Problem when x64 appeared without segment limits in 2003/4
    - Necessary to use pagetable switching, tricks to preserve TLB entries

Xen and Virtualization Extensions

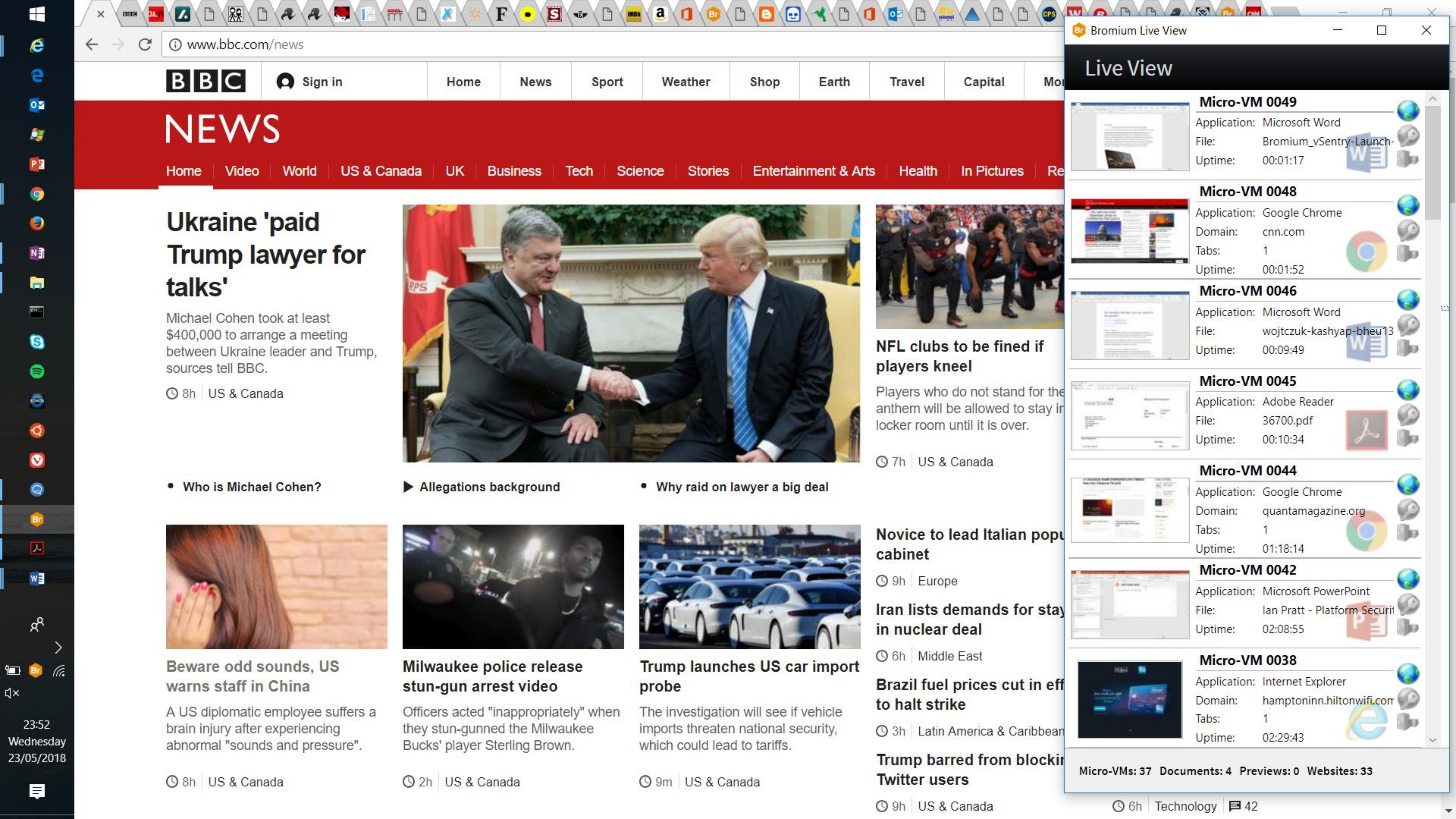
- VT-x / AMDV arrived in 2005/6, Xen was ready with support
- VM entry/exit initially very slow, started getting quite good in 2008
  - Avoid enter/exit roundtrips by looking ahead in instruction stream
    - Scary complex x86 emulation
- Shadow pagetables required to handle composite memory translation until EPT/NPT in 2008
  - Required considerably complexity to make perform well
  - Performed better than EPT/NPT until 2009
- By 2009 it was clear that using virtualization extensions was just better in every way, especially in reducing hypervisor complexity and hence improving security
- Having a large deployed base of legacy VMs on legacy hardware makes it hard to move forward
  - 10 years later still not dead

XenClient

- Opportunity to create a showcase for how Xen should be configured for security
- Guest VMs use VT-x/AMDV
- Dissagregation
  - Qemu stub domains
  - Restartable driver domains e.g. for network/WiFi ; USB Storage
  - Service VMs e.g. VPN VMs
- Mandatory Access Control : SELinux dom0, XSM
- Required PCI passthrough for driver domains, GPU etc using VT-d/IOMMU
  - Worrying complexity; Need to really understand device BARs, config space etc
- DRoT with TXT, enable attestation, use sealed storage for encryption keys
  - Challenges making TXT work on vendor platforms, very limited STM BIOS availability

Bromium vSentry Requirements

- Create a VM for every user-centric task
  - Every web page, every document, every email etc
  - Support many concurrent VMs on laptop/desktop hardware
    - VM cloning, Copy-on-Write memory
- Transparent to the end user
  - Create clone VMs in tens of milliseconds
  - Great interactive performance, battery life
  - Support multi touch screens and trackpads etc
- Cross platform: Windows, MacOS, Linux/Android
- Must provide very robust spatial isolation
- Introspection into VMs for forensic purposes



Ukraine 'paid Trump lawyer for talks'

Michael Cohen took at least $400,000 to arrange a meeting between Ukraine leader and Trump, sources tell BBC.

8h | US & Canada

- Who is Michael Cohen?
- ▶ Allegations background
- Why raid on lawyer a big deal

NFL clubs to be fined if players kneel

Players who do not stand for the anthem will be allowed to stay in locker room until it is over.

7h | US & Canada

Novice to lead Italian popu cabinet

9h | Europe

Iran lists demands for stay in nuclear deal

6h | Middle East

Brazil fuel prices cut in eff to halt strike

3h | Latin America & Caribbean

Trump barred from blockin Twitter users

9h | US & Canada

Beware odd sounds, US warns staff in China

A US diplomatic employee suffers a brain injury after experiencing abnormal "sounds and pressure".

8h | US & Canada

Milwaukee police release stun-gun arrest video

Officers acted "inappropriately" when they stun-gunned the Milwaukee Bucks' player Sterling Brown.

2h | US & Canada

Trump launches US car import probe

The investigation will see if vehicle imports threaten national security, which could lead to tariffs.

9m | US & Canada

6h | Technology | 42

Micro-VM 0049
Application: Microsoft Word
File: Bromium_vSentry-Launch-
Uptime: 00:01:17

Micro-VM 0048
Application: Google Chrome
Domain: cnn.com
Tabs: 1
Uptime: 00:01:52

Micro-VM 0046
Application: Microsoft Word
File: wojtczuk-kashyap-bheu13
Uptime: 00:09:49

Micro-VM 0045
Application: Adobe Reader
File: 36700.pdf
Uptime: 00:10:34

Micro-VM 0044
Application: Google Chrome
Domain: quantamagazine.org
Tabs: 1
Uptime: 01:18:14

Micro-VM 0042
Application: Microsoft PowerPoint
File: Ian Pratt - Platform Securit
Uptime: 02:08:55

Micro-VM 0038
Application: Internet Explorer
Domain: hamptoninn.hiltonwifi.com
Tabs: 1
Uptime: 02:29:43

Micro-VMs: 37 Documents: 4 Previews: 0 Websites: 33

µXen

- Package hypervisor as a platform independent module that can be loaded by Host kernel
- Set of in/out interfaces linked at module load time
- Host thread calls in to uXen module to run VCPU
  - Return when need IO assistance, or when pre-emption possible
  - Call out from module to host kernel for memory allocation, cross-CPU synchronization
- Use host OS scheduler

µXen Architecture

- Require VT-x/AMDV, EPT/NPT
  - No legacy hardware support
  - No legacy guest support (guest automatically recreated)
- PV device interfaces all built on simple hypervisor copy-based primitive
  - No memory sharing (grant tables); copying
  - No xenstore, though still allow device reconnection
  - Simple, narrow interfaces
- Emulated devices irrevocably disabled post-boot (prior to exposure to anything untrusted)
  - Simple Viridian synthetic devices for LAPIC, timers
  - Only very simple instruction decoder required
- Only 3 of the many potential guest escape XSA’s have ever been relevant to uXen

µXen VM Monitoring

- Collect threat intelligence by monitoring guest execution
  - Black box flight recorder trace of execution, held in host to prevent tamper by guest
  - Introspection of key data structures, network, storage; plus guest instrumentation
- Since application is known, look for divergence from expected behavior
  - State machine generates trigger when something interesting happens
  - Most users just allow execution to continue and collect full kill chain
    - Nothing to steal; no way to move laterally; no way to persist
    - Attacker thinks they have succeeded
  - Preserve flight recorder trace, and stream to collection server for analysis

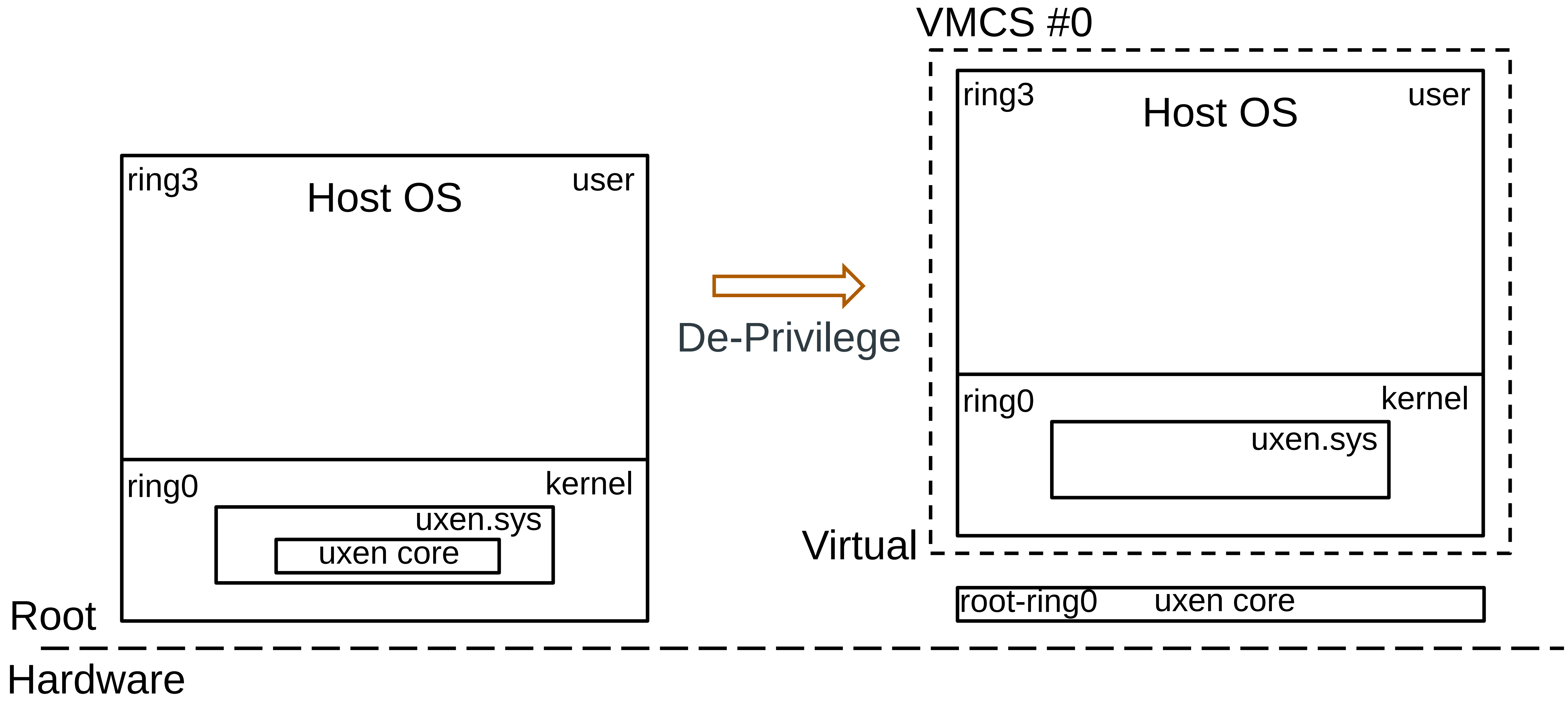
µXen Experience

- Installed on a lot of systems, HP Sure Click
  - Billions of VMs created
- Internal and external review
- Code auditing
- Grey box pen testing
- Bug bounties
- Fuzzing, fault injection on hypervisor entry path
- Exploit mitigation techniques e.g. separate heap for any user-controlled data
  - Never memory map more than you need
- Use all the help from the compiler and tools you can get

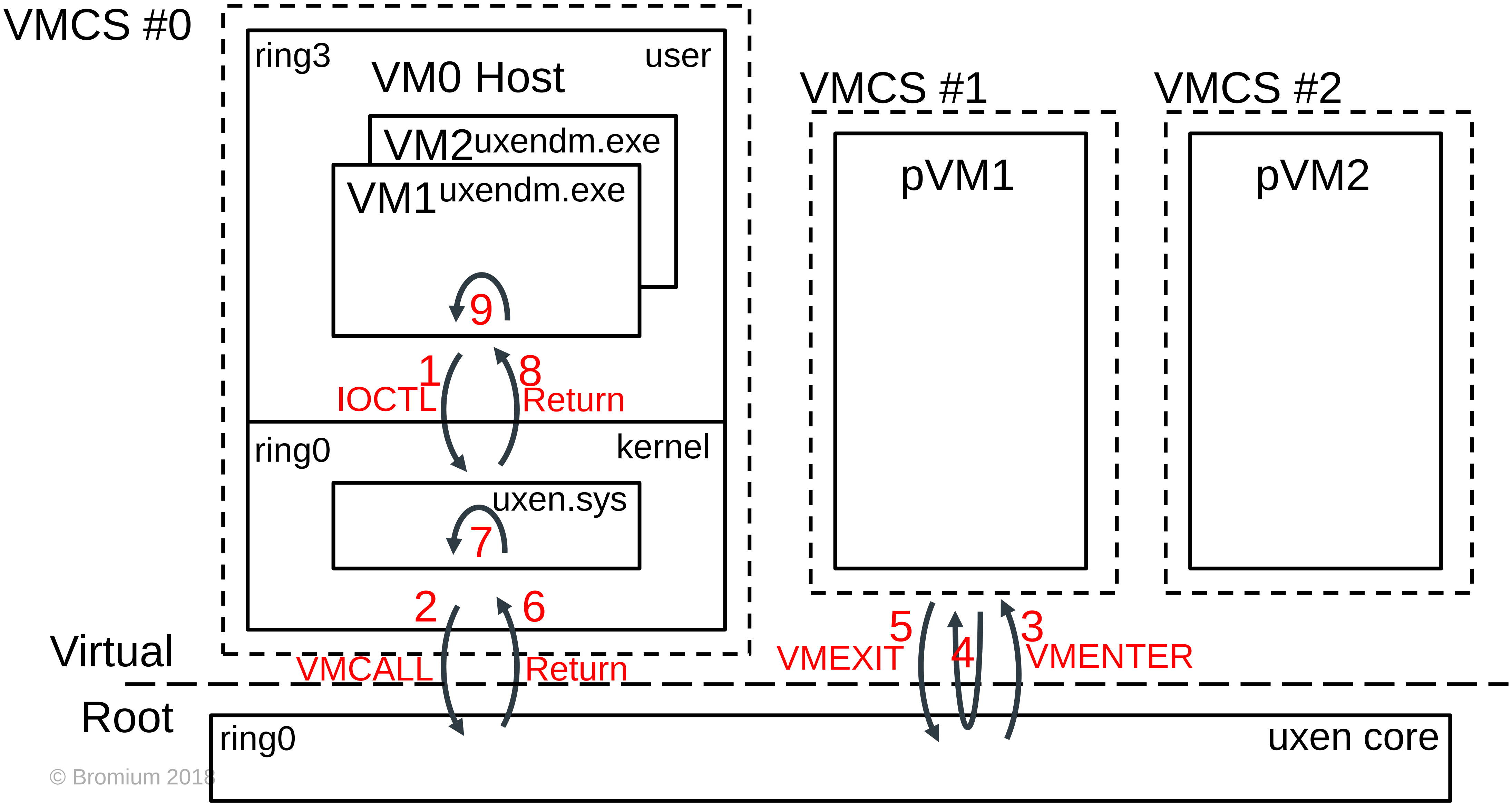
uXen “Type-1.5” Extensions

- Design goal: Allow some VMs that are more trusted than the host, Protected VMs (pVMs)
  - Protected from the host from a Confidentiality and Integrity point of view
  - Use pVMs for running high-value applications and their OS
    - Not just small sensitive parts of applications as per SGX
- Runtime de-privileging of the running host into a VM
  - Establish DRoT with TXT
  - Create Host VMCS and EPT/VT-d tables to allow access to all resources except those used by hypervisor module and pVMs
  - pVMs use host for IO, should ensure encrypted and authenticated (VPN, dm-verity etc)
  - Measure and attest to initial state of each pVM
- Able to kill pVMs, scrub memory, re-privilege the host

Host CPU De-privilege



Execution Model



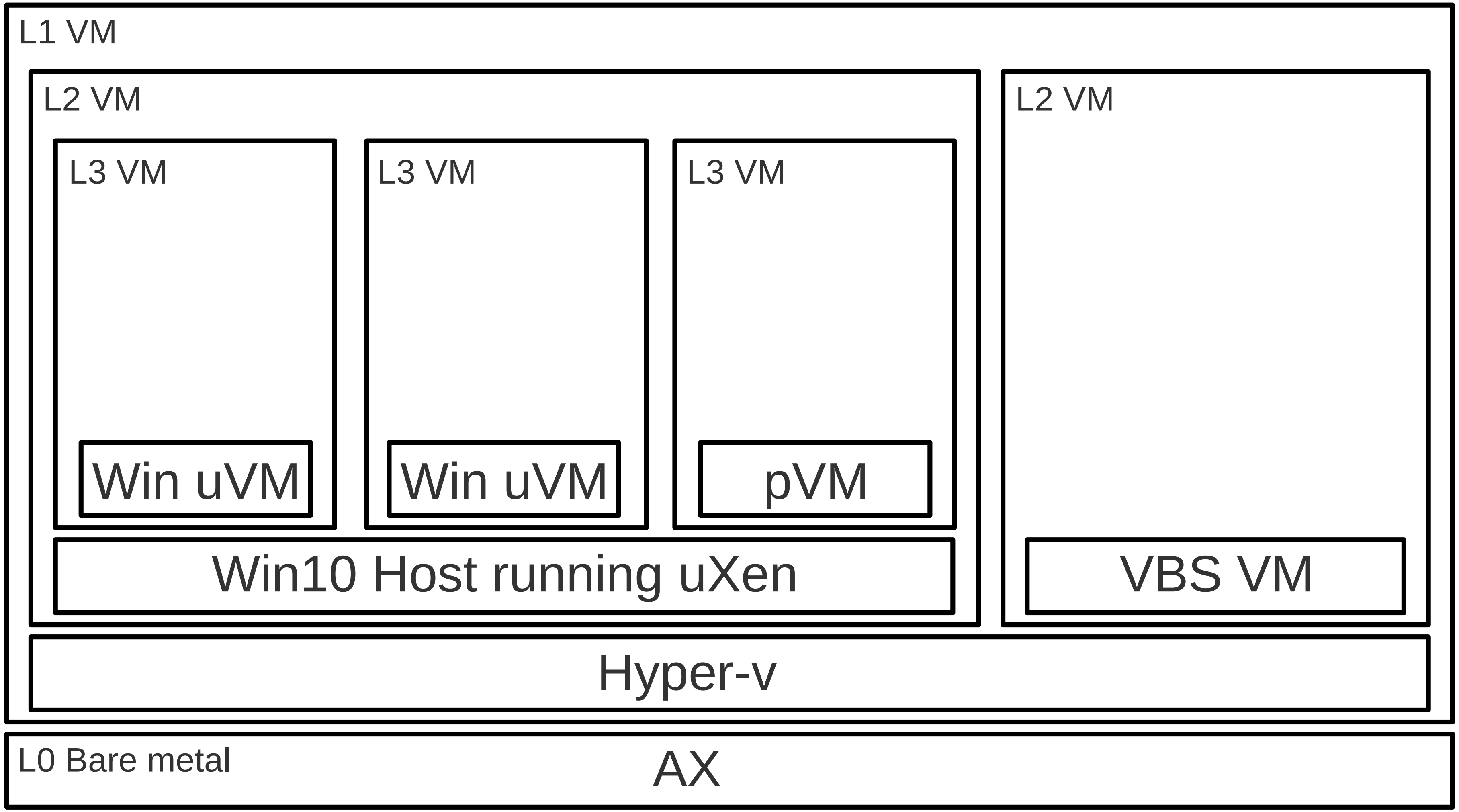
AX Design Goals

- Build on ideas from uXen-T1.5
  - Protected VMs concepts important in Client and Cloud
    - Reduce trust in Cloud Providers
    - Run high-value applications on hosts of unknown state (e.g. BYOD)
- Focus on minimal TCB
  - SRoT and DRoT
- Embrace nested virtualization
  - Common in Cloud; Client Hyper-v

AX Architecture

- UEFI module, de-privilege running system ahead of host OS
  - Can load from system disk or package as DXE/PEI firmware module
- Minimal TCB, just a few KLOC of guest-facing code
- Implement minimum for spatial isolation of resources, Confidentiality and Integrity
  - Scheduling is outside TCB – can use trusted RTOS / SE Linux if you care
  - Hierarchical subdivision of resources
    - Enforce resource Subset rules, Enforce Exclusion rules for Protected VMs
- Enables arbitrary nesting of VMs, even if nested hypervisors don’t support nesting
  - Enables uXen to run on top of Win10 hyper-v even though hyper-v doesn’t support nesting
  - System performance excellent on modern hardware

Win10 VBS virtualization stack with AX



AX Isolation Enforcement

- AX ensures nested VMs are contained and can not exceed the resources of their parent VMs or impact their privacy or integrity
  - Thus VBS CG/DG isolation design goals are maintained
    - Enhanced through additional introspection of hyper-v
- When Protected VMs are created, enforcement of spatial protection is made symmetric
  - Confidentiality and integrity of the child VM is ensured from the parent as well as vice versa
  - Ensure EPT/VT-d memory regions of VMs are disjoint
    - Remove pages referenced in child VM’s EPT from all parent VMs’ EPT
    - When child VM terminates scrub and return pages
    - Use AMD Secure Encrypted Virtualization (SEV-ES) features for additional protection
  - Keep child VM register state and VMCS/VMCB state in AX, parent VM sees and manipulates shadow state only

Protected VM IO/MMIO/DMA operations

- Outer hypervisor can not see pVM’s registers or memory state
  - Hence traditional instruction emulation or virtual DMA not possible
  - Use proven uXen communication primitive, guest drivers and backends
- Use specially configured Linux kernel to use PV drivers, LAPIC, timers
- For Windows use reflective injection of IO/MMIO events back into an instruction emulator running in the context of the pVM that will then use the existing PV interface
  - Allows register and memory access since in context of pVM
  - De-privileges complex emulation code keeping AX small and simple
  - Fits with AMD SEV-ES

Protected VM Input/Output paths

- Confidentiality and integrity provided within pVM, but data passes through host/drivers
  - Net : Use TLS/IPSec connections terminated in pVM
  - Block : Use Authenticated Encryption / Merkle hash trees for integrity
- Take ownership of device in a Service VM, virtualize to other VMs
  - Keyboard : Route input to the currently focussed pVM thus preventing snooping or injection
    - Easier with laptop keyboards (PS2), harder with USB – use “shadow URBs” to parse traffic and extract HID events
  - Enables restartable driver domain model as per XenClient
- Secure video path immune to screen scraping or injection remains challenging if host OS is allowed to use GPU
  - Not all use cases require secure video path
  - Use s/w rendering; GPU stealing; or separate GPUs
  - Ongoing work with h/w vendors to support safe sharing or secure overlays

Measurement and Attestation

- Populate memory image of S3 suspended VM (no device or CPU state), measure on launch
- Only launch PKI signed pVMs with a certificate chain that can be validated against a list of CA certs to prevent abuse
- Allow pVM to get TPM quotes of boot state and VM launch state, attest to 3rd parties
    - Use vTPM or moderated pass-through of hardware TPM
- Currently, destroy protected VMs on host S3/S4 sleep
- Future option to allow save/restore of VMs using authenticated encryption

AX Experience

- Easy to get broad platform support for a client hypervisor
  - Tested on all HP Systems
- Exploit mitigation techniques have proved useful
  - AX contains no indirect branches – proved helpful with Spectre
    - Absolute branches due to CFG; Extreme ASLR
- Introspection capabilities have proved very helpful
  - Monitoring integrity of Hyper-v, Windows
- Scales well to very large systems, down to small IoT systems
  - Very useful security properties for IoT, Client and Cloud
- Architecture has a huge influence on Security. Keep it Simple and Small.