HMX:

Hypervisor-**M**ediated data e**X**change

Primitives for Authentic Communication

Christopher Clark
Edgeform

Platform Security Summit, 23-24th May, 2018

Presenter: Christopher Clark

Software Engineer

Contributor to: **OpenEmbedded**, the **Xen Project** and **OpenXT**.

All opinions expressed are those of the speaker and not previous or current clients.

Xen affiliations:

- **OpenXT Project** - since January 2016.
- Citrix Systems
- XenSource
- Intel
- Cambridge University Computer Laboratory

Xen Projects: Xen hypervisor - XenServer - XenClient - XenClient XT - OpenXT

Refresh from prior Xen Summit material

Concepts presented by John McDermott of **US NRL** about **Xenon** at the **Xen Summit 2007**,
with acknowledgement to their **origin at the NSA.**
Also presented by **Daniel Smith** for **TrenchBoot** at Platform Security Summit 2018.

Robustness = (Strength of Mechanism, Implementation Assurance)

Strength of Mechanism:

"What flaws would be present even if we had a perfect implementation?"

Implementation Assurance: "How well did we build it?"

"It is pointless to build a high-assurance implementation of a low-strength feature." -- Xenon presentation.

The Xen Project Direction in 2018

The Xen Community and its downstream projects are working towards:

Xen hypervisor deployed in Security, Safety and Mixed-Criticality systems.

-> great! supports the OpenXT vision and direction.

It is important for these deployments to succeed.
The software needs to be **robust.**

-> We need to provide **strong separation mechanisms.**

With reference to Rushby's Separation Kernels (SOSP, 1981)

Design of a Secure System: construct as a distributed system to separate concerns.

A hypervisor != separation kernel (usually), since a hypervisor typically:

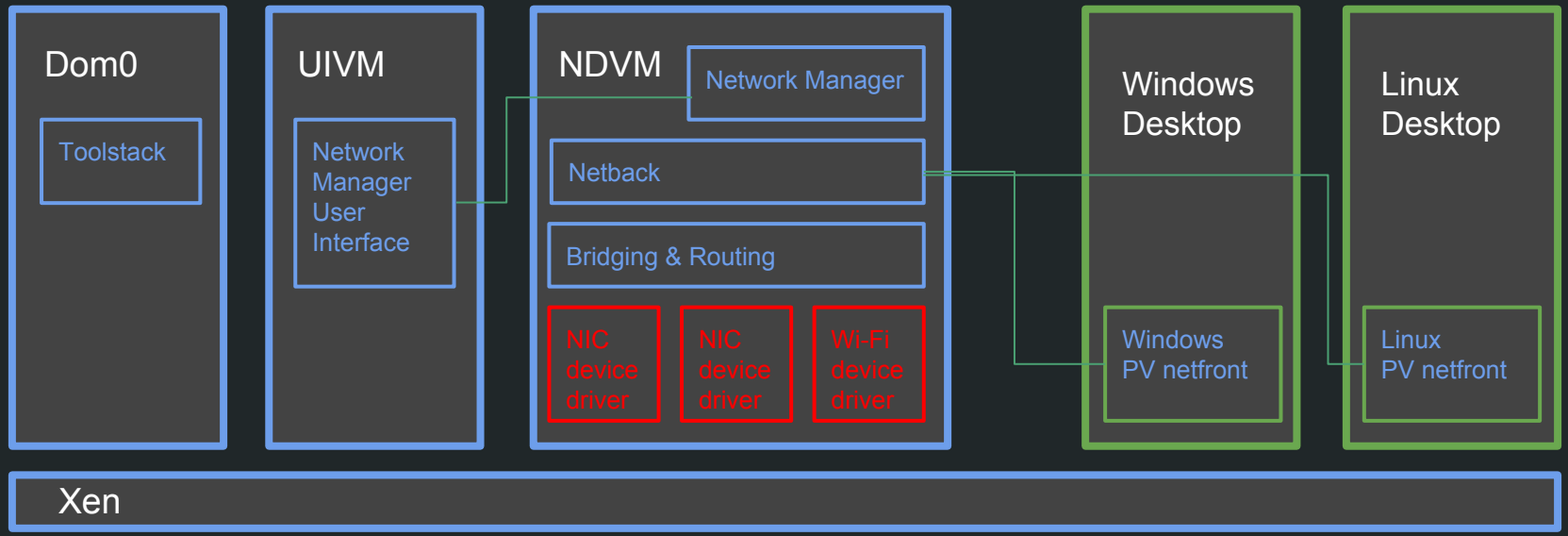
- Aims to provide some simulation of platform hardware
  and its modern features to the hosted guest VMs.

- Lower design priority is given to enforcing absolute separation
  between its execution environments
  versus enabling performance and platform support for advanced workloads.

but: Safety, Security and Mixed-Criticality use cases for Xen are motivating pursuit of properties that a Separation Kernel provides.

OpenXT: Disaggregated system architecture

A key distinguishing feature of the OpenXT system



Default configuration: All of the physical PCI network devices are assigned to a single Network Device Virtual Machine (NDVM) and isolated using VTd.

ref. Separation Kernels cont’d...

Safety, Security and Mixed-Criticality use cases for Xen are motivating pursuit of properties that a Separation Kernel provides - so:

We need to be
evaluating Xen’s mechanisms through the lens of Separation Kernel research,

and the developed wisdom gained in that field of systems, because:
It provides guidance for understanding the architecture and design of the mechanisms in the hypervisor.

MILS Architecture Foundational Security Principals (2005)

- **Data Isolation**
  Information in a partition is accessible only by that partition and private data remains private.

- **Control of Information Flow**
  Information flow between partitions is from an authenticated source to authenticated recipients;
  the source of information is authenticated to the recipient, and information only goes where intended.

- **Periods Processing / Temporal Separation**
  Resources may be used by different components by time-slicing, where the system enforces that the resource is cleaned to remove any trace of its previous use before being reassigned.

- **Fault Isolation**
  Failure within a partition is prevented from cascading to any other partition.
  Failures are detected, contained and recovered locally.

Data Isolation : What happened recently?...

Following Meltdown and Spectre: “**Panopticon Xen**” proposal

Assume that all mapped memory is visible to the running vCPU.
-> ie. supervisor / hypervisor read-access boundaries have been defeated.

New design model:
Assume that the guest can see anything in the hypervisor address space.

If Xen is re-architected to make sensitive data inaccessible,
so you’re still OK even under those conditions,
then Data Isolation and Fault Isolation have been improved.

HMX : **H**ypervisor **M**ediated data e**X**change

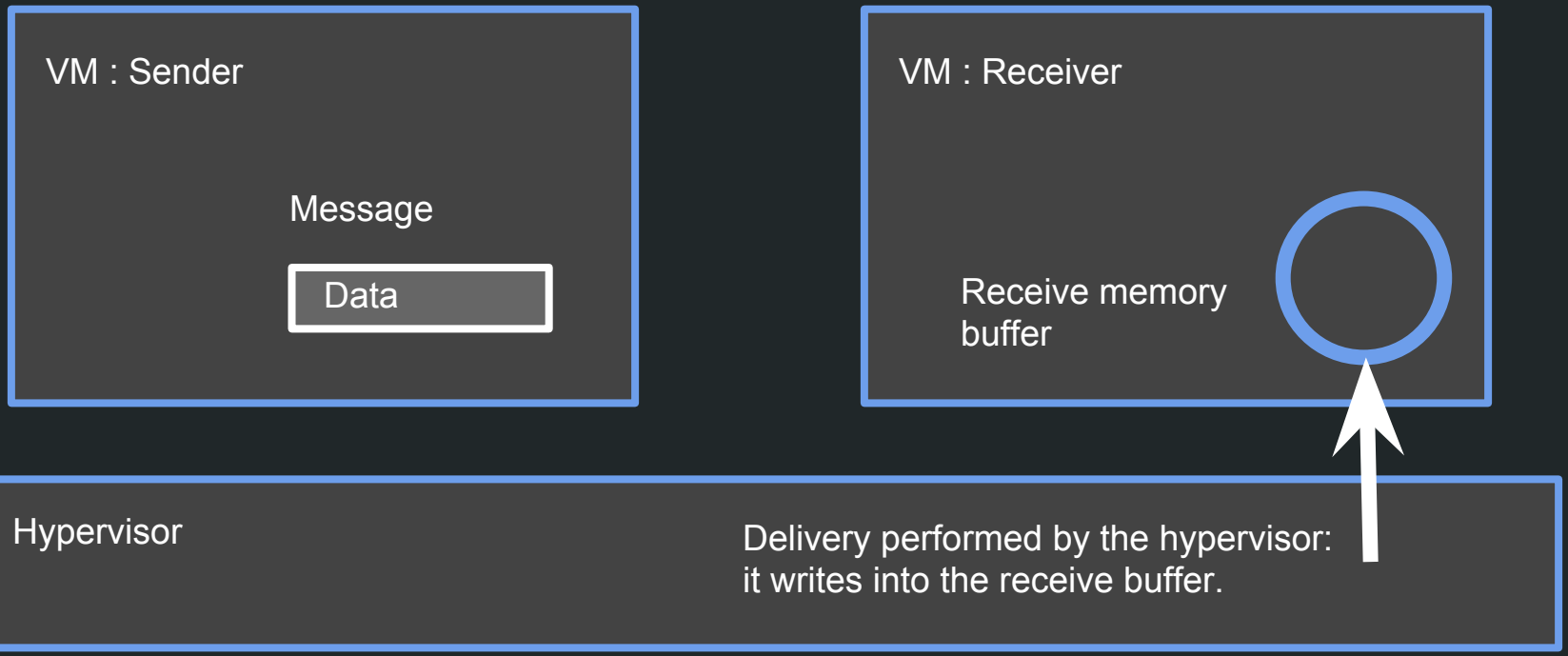
A term to describe:

Asynchronous message passing between VM partitions, performed by the hypervisor.

Channels use no shared memory between the source and the receiver.

Aiming to enable the possibility of enforcing Control of Information Flow between domains via this channel and preserve Data Isolation.

HMX


VM : Sender
Message
Data
VM : Receiver
Receive memory buffer
Hypervisor
Delivery performed by the hypervisor:
it writes into the receive buffer.

Inter-partition Communication: **Xen**

Traditional PV communication channels using primitives: Grants + Events

Grants are most commonly used to share pages between domains.
(Direct page transfers fell out of favour for performance and security concerns.)

=> HMX: no.

Inter-partition Communication: **Hyper-V**

VMBus : HvPostMessage

Messages are copied by the hypervisor into a private per-receiver message buffer in hypervisor-owned private memory.

For delivery, individual messages are then copied by the hypervisor out into a Message Page shared between the hypervisor and the receiver partition, when free slots within the page are available.

=> HMX: **yes.**

since at least 2006.
but: Microsoft does not appear to want this channel used between peer VMs.

Inter-partition Communication: **SEL4**

IPC: True to its microkernel architecture, SEL4 does not buffer IPC data in the kernel address space.

IPC messages between partitions are sent via the kernel and carry only register state. Shared memory regions established between tasks are used for data transport of message contents.

=> HMX : no.

Inter-partition Communication: **Linux-KVM**

Virtio

Virtio rings are established within shared memory regions between front and back ends of split device drivers, to carry payload data.

=> HMX: no.

Virtio also requires that the domain emulating the device has memory mapping privilege over the front end domain, which impacts the isolation between them.

Inter-partition Communication: Xen in OpenXT

v4v

=> HMX : **yes.**

v4v : an interdomain communication transport

- An OpenXT technology, originally developed for XenClient.
- Hypervisor-mediated data copies via private ring buffers with notifications.
- Used by OpenXT and in production in its derivative products,
  and a variant has been in use at Bromium.
- Benefitted from previous reviews by the Xen Community.

v4v : an interdomain communication transport

Motivations for v4v versus any other interdomain communication mechanism:

- Strong isolation between communicating domains
  - No memory is shared between VMs
- Strong enforcement of policy controls on VM communication
  - A firewall within the hypervisor enforces rules that are set externally
- High performance suitable for sustained throughput
- A clean mapping to Linux and Windows native I/O primitives
- Clear separation from guest Operating System networking stacks
- A foundation for the future work that we intend to do

Roadmap

Upstream target: Xen 4.12 (Q4 2018)

Watch Xen development postings for:
“Inter-VM communication primitives for hypervisor mediated data exchange.”