Xen Project 15 Years Down the Line:
Thinking Outside of the Conceived Tech Comfort Zone
Lars Kurth
Community Manager, Xen Project
Chairman, Xen Project Advisory Board
Director, Open Source, Citrix
IRC lars_kurth

Contributors
Andrew Cooper: Committer, Xen Project
Christopher Clark: Contributor, Xen Project
Mihai Donțu: Chief Linux Officer, Bitdefender
George Dunlap: Committer, Xen Project
Happy Birthday!
What is Xen and Xen Project?

Versatile Virtualization Platform
Designed to be a component in a SW stack
Ease of use for end-users was never a design goal

Xen Hypervisor = “Engine”
Taken by integrators to build a product, service, …
Analogy: integrators build a “car” with Xen as “engine”

Xen Project
Development community with several sub projects that develop technologies related to Xen
- Hypervisor
- PV Drivers
- Unikernel related projects: MirageOS, Unikraft
Rewind to the Beginning
Original Use Cases

Server Consolidation
Co-located hosting facilities, Distributed web services
Cloud Computing

Secure Computing Platforms and Application Mobility
High availability, Fault tolerance, Checkpointing
Original Design Goals

Modular architecture
Via disaggregation

Flexible architecture
Via use of system services

High degree of customisability
Initial goal: server and possibility of desktop products
So, how did we get to such a diverse product eco-system?
Strong Security mindset by main Xen Developers
Disaggregation was part of the original Xen conception. Influenced by microkernel thinking.
Disaggregation was part of the original Xen conception. Influenced by microkernel thinking.
Disaggregation was part of the original Xen conception
Influenced by microkernel thinking

The PV protocols were specifically designed for untrusted backends and untrusted native device drivers
Particularly the grant tables

Security technologies were always in maintainers minds
They were up-streamed without resistance
Sometimes with a lot of support from maintainers
Today: Used by ...

- Cloud Computing
  - Unikernels
  - VMI
  - Xenon Hypervisor family, Magrana Server, ...
- Server Virtualization
  - Defense Applications
  - General purpose desktop and mobile Virtualization
  - Qubes OS
  - Embedded Defense / Security Applications
  - Automotive
  - ARLX/Virtuosity OA, Crucible Hypervisor
  - Virtuosity, GlobalLogic Nautilus, EPAM Fusion
- Defense Applications
Tension: Core vs. non-Core Usage

Many security technologies were not made easy to use
Disaggregation is a good example

Product schedule pressure, led to

- Core team members choosing the quick to implement / more performant approach → New features went into Dom0 first
- Maintainers reviewing non-Core contributions: takes effort to understand and guide contributors to the best approach → This did often not happen

E.g. Xen has no build capability for disaggregated domains
Led to duplication amongst vendors
Increased the barrier to adoption for newcomers
Strong Security Mindset
Enabled non-Core Security Contributions
From Military Clouds to Desktop/Tablet Use Cases on x86
Xen Summit: 2007, New York, USA
US DoD employees presented approaches to Apply EAL5 Common Criteria Security Clearance for Xen Xen Security Modules (XSM)

EAL5 CC related research led to the Xenon Separation VMM family

Today: widely used by DoD departments

XSM was subsequently upstreamed
2012: The Xenon Separation VMM
Secure virtualization infrastructure for military clouds

“We reduce the size of the code base by dropping support for selected features that are not needed to run commodity operating systems, e.g. Windows 7 or Linux. Because the internal design and construction of the base Xen VMM is highly modular, removing some of the features does not disable any of the remaining features.

The total effort required was about 3 engineers over a period of 2 years, and includes the effort of keeping up with the Xen community.”
x86: Desktop / Tablet Use Cases

2009: Virtual Computer releases NxTop 1.0
NxTop is the first of 3 Xen based client solutions
Citrix and Neocleus release their first solutions in 2010

2012: Snowden-approved Qubes OS 1.0 releases
Today, Qubes OS 4.0 is the first technology to use PVHv2 in production

2013: Citrix and DoD collaborate to create SecureView
In 2015, XenClient is discontinued and released as OpenXT
SecureView continues to be developed on top of OpenXT
Reflection

Early & continued engagement of Security Researchers
Security feature contributions before being productized
Xen became an attractive platform for security research

Enabler:
The Xen community accepted contributions

Modular Architecture
Enabled specialised and certifiable products for defence use

But: Innovations were not attempted to be up-streamed
Reflection

Attempts to commercialize Xen on Desktop
Enabled a new products leveraging Xen’s security features
They also relied on removing upstream Xen code (on x86)

BUT:
None of these Xen capabilities were used in Server/Cloud

Related Story: GPU Passthrough & Virtualization
Were developed for Desktop Use Cases on Xen

TODAY:
GPU Virtualization is widely used in Server/Cloud
Virtual Machine Introspection: from Research to Commercial Server Products
Georgia Tech: 2007, Atlanta, USA
2007: Virtual Machine Introspection for Xen hatched at Georgia Tech

- 2003: Initial Research by Tal Garfinkel & Mendel Rosenblum
- Bryan Payne created a project called XenAccess
  Later expanded scope to other hypervisors as LibVMI

2009: XenAccess and mem-events APIs were added
Used for security research and specialist security apps

Enabler:
The project accepted these changes with active help from maintainers
Commercial interest in VMI

2013 to 2015: Intel launched its Haswell & Broadwell CPUs Specifically VMFUNC, EPTP and #VE capabilities

New HW capabilities of Haswell & Broadwell generated commercial interest in VMI by multiple vendors

2014 to 2015: The XenAccess and mem-events APIs were re-architected into the VMI subsystem

In parallel: related companion technologies such as alt2pm are being developed
Productization

2015 to 2016: Citrix and Bitdefender collaborated to bring VMI to market through XenServer 7.0 and Bidefender HVI

Today: Similar products are available from AIS and Zentific

Today: ongoing contributions by security vendors to alt2pm indicate future productization
Reflection

Few security features made it into Server/Cloud
For example: VMI & Live Patching

WHY?

HW capabilities were not enough
It was too early or the market wasn’t ready
Example: Dynamic Root of Trust related technologies

Complex and Use Case dependent
Example: Xen Security Modules - requires Specialist expertise; configuration is Use Case dependent
From Desktop / Mobile Use Cases to Embedded / Automotive
2008, Seoul, Korea
2008, Seoul, Korea

Samsung releases Secure Xen on Arm port
This Xen port used paravirtualization and was kept as a Xen fork hosted by the Xen Project as requested by Samsung

2008 to 2014: Samsung regularly shows Xen demos
Dual mobile operating systems running on Samsung devices and later Nexus 10

Demos and presentations covered
  – Innovations in real-time capabilities based on Xen
  – GPU sharing technologies based on Xen
This showed what is possible
Inspiring others to investigate and research along similar lines
Xen on Arm reboot

Patches for upstream Xen on Arm support are posted
Armv7 and v8 are fully supported upstream by March 2014

The primary motivation for this work was to target Arm based servers

Developers learned from the earlier Arm port:
- Clean and simple architecture
- Perfect match between Xen and Arm ISA
- Much smaller code size compared to Xen on x86
Upstream support and simplicity of Arm port enabled embedded Use Cases on Arm
The missing Evolutionary Link
From x86 to Arm

2010: Dornerworks contributes the ARINC653 Scheduler
Launches ARLX (initially for x86) taking a similar approach
to Xenon

2014: Dornerworks pioneers safety certification for Xen
A new Evolutionary Wave

2014: Xen moves into embedded and automotive
The Xen Project launched an automotive initiative
Xen based distros and products are being developed

2015: First automotive Xen based platform
GlobalLogic introduces the Nautilus platform at CES 2015

2015: First embedded Arm-only Xen Distro
Xen Zynq Distribution for XILINX

The last few years: R&D
Similar order of magnitude to earlier waves
Sedimentation: from Software to Hardware
Sedimentation:
Functionality implemented in Software being fully or partly implemented in Hardware

Example: PV Interrupt and Timers ➔ using IO APIC and Posted Interrupts

Silicon vendors have always worked closely with Xen
Xen’s had a significant impact on virtualization related technology in Hardware

I wanted to tell this story, but:
It’s very difficult to explain
And much of the work was performed under NDAs
What have we learned?
What can you learn from Xen?
Culture enables Innovation

Accept changes for non-Core Use Cases
In our case this was not planned: we inherited the culture
But: do not sacrifice due diligence and future-proofing

Help people trying to adopt Xen for non-Core Use Cases
We were bad at this until about 3 years ago

Work with new ideas, forks and distributions
Developer Events: allow air-time for new ideas
Forks are not always bad: Samsung fork hosted by us
OpenXT (distro): close collaboration is key
Process enables Innovation

Feature/Configuration Classification
Experimental ➔ Tech Preview ➔ Supported
Marked non-Core functionality as Experimental or Preview

Maintainership
Contributors need to step up for larger non-Core Features
Proves longer term commitment by the contributor

Deprecation of Features
Intent to deprecate is announced on list: identify objections
Remove non-core Features, if stale or not used
Tension need to be Managed

**Vendors have different Priorities**
These can lead to conflicts
Non-Core vs. Core arguments can be ugly
Active community management is required

**New Processes can lead to Tensions**
Security Vulnerability Management led to problems
Who deals with a Vulnerability in non-Core code?
We didn’t understand this ➔ created tension and rejection of non-Core features until we modified the process
Process enables Innovation

Security Support
The security team does not handle non-Core Vulnerabilities
- Supported Feature without Security Support
- Delegated Security Support (e.g. ARINC653 scheduler)

KCONFIG based Configuration Management
Larger non-core features are build or run-time disabled

Subprojects as incubators for New Ideas
Can be used to foster innovation on the periphery
Examples: MirageOS, Automotive Project, Unikraft
Questions

lars.kurth@xenproject.org
References

https://wiki.xenproject.org/wiki/15_Anniversary_References