Live Patching, Virtual Machine Introspection and Vulnerability Management

A Primer and Practical Guide

Lars Kurth
Community Manager, Xen Project
Chairman, Xen Project Advisory Board
Director, Open Source, Citrix

Presentations on www.slideshare.net/xen_com_mgr/presentations
Terminology

Xen Project / XenServer

Security
Xen Project is the Engine
XenServer is the Car
Vulnerability
A weakness in the computational logic of software that, when exploited, results in a negative impact to confidentiality, integrity, OR availability of that software.

Patch / Live Patch
A fix for a security vulnerability.
A live patch can be applied to a running system
Malware
Software designed to disrupt, damage, or gain authorized access to a computer system.

Exploit
Malware designed to take advantage of a vulnerability
This includes file-less/memory based attack techniques (around 14% at end of 2016)

0-Day Exploit
An exploit of an undisclosed/previously unknown vulnerability that is/has been exploited

Rootkit
Software tools that enable unauthorized users to gain control of a computer system without being detected, some for for a long period of time

Some malware relies on social engineering: e.g. phishing, trojans, adware, ...
Story 1: Virtual Machine Introspection

A new way to protect against malware

Developed by Zentific, Citrix, Bitdefender, Intel and others
Traditional Cloud Security

Installed in-guest agents, e.g. anti-virus software, VM disk & memory scanner, network monitor, etc. Can be disabled by rootkits.

Dom0
- Dom0 Kernel
  - Drivers

VM2
- App
- Guest OS
- Agent(s)

VM3
- App
- Guest OS
- Agent(s)

VMn
- App
- Guest OS
- Agent(s)
A new model for Cloud Security?

Uses HW extensions to monitor memory (e.g. Intel EPT) ➔ Low Intrusion
Register rules with Xen to trap on and inspect suspicious activities
(e.g. execution of memory on the dynamic heap)

- Dom0
- Security Appliance VM
- Introspection Engine
- VM
- App
- Guest OS

Protected area

Authentication mechanism to protect the IF
Protection against attack techniques

All malware need an **attack technique** to gain a foothold.
Attack techniques exploit specific software bugs/vulnerability.

Most exploits use one of a **small set of attack techniques**:
Buffer Overflows, Heap Sprays, Code Injection, API Hooking, …

Because VMI protects against attack techniques:
It can protect **against entirely new malware**.

Verified to block these advanced attacks in **real-time**:
APT28, Energetic Bear, DarkHotel, Epic Turla, Regin, ZeuS, Dyreza, *EternalBlue*¹
… solely by relying on VMI.

¹ [businessinsights.bitdefender.com/hypervisor-introspection-defeated-enternalblue-a-priori](businessinsights.bitdefender.com/hypervisor-introspection-defeated-enternalblue-a-priori)
Protection against rootkits & APTs

Rootkits
Exploit 0-days in Operating Systems/System Software
Can disable agent based security solutions (mask their own existence)

VMI solutions operate from outside the VM
Thus, it cannot be disabled using traditional attack vectors

BUT:
VMI is not a replacement, for traditional security solutions
It is an extra tool that can be used to increase protection
If you want to know more ...

Documentation & Demos
wiki.xenproject.org/wiki/Virtual_Machine_Introspection
youtube.com/watch?v=qpQPBvOniUU (shows Bitdefender HVI)

Products

**Bitdefender HVI**
XenServer
www.bitdefender.com
Protection & Remedial
Monitoring & Admin
Citrix Ready

**AIS Introvirt**
XenServer
www.ainfosec.com

**Zentific Zazen**
Xen & XenServer & …
www.zentific.com
Protection & Remedial
Monitoring & Admin
Forensics & Data gathering
Malware analysis
Story 2: Vulnerability Management in Xen Project

Result of several community consultations
Software bugs happen
Some will be security vulnerabilities
Xen Project: Responsible Disclosure
xenproject.org/security-policy.html

Fixing Security Bugs:
Dedicated security team =
security experts from within
the Xen Project Community

Security Team:
Triage
Creation of fix/patches
Validation of fix/patches
Assignment of CVE
Issue description and risk analysis

R: Vulnerability reported to security@xenproject.org
P: Vulnerability pre-disclosed on xen-security-issues@lists.xenproject.org
Xen Project: Responsible Disclosure
xenproject.org/security-policy.html

Fix their systems/software:
Eligible Xen Project Users
are informed under embargo
of the vulnerability

Eligible Users = Pre-disclosure list members:
Product Companies, Open Source & Commercial Distros (e.g. Huawei, Debian)
Service/Cloud Providers (e.g. Alibaba)
Large Private Downstream (e.g. Google)

Allowed to share information via
xen-security-issues-discuss@lists.xenproject.org

R: Vulnerability reported to security@xenproject.org
P: Vulnerability pre-disclosed on xen-security-issues@lists.xenproject.org
A: Vulnerability announced on xen-announce@lists.xenproject.org & xenbits.xen.org/xsa
Xen Project: Responsible Disclosure

xenproject.org/security-policy.html

General Publication:
Information about vulnerability is made public

Everyone else:
Patches their systems either through security updates from distros/products or builds them from source.

Users of service/cloud providers will not be impacted

R: Vulnerability reported to security@xenproject.org
P: Vulnerability pre-disclosed on xen-security-issues@lists.xenproject.org
A: Vulnerability announced on xen-announce@lists.xenproject.org & xenbits.xen.org/xsa
Xen Project: Responsible Disclosure
A note on Live Patching

Product Vendors:
Create and test live patches

Product customers:
Apply live patches here

Security Team:
Can a Livepatch can be created?
No? If possible, re-write fix/patches

Service Providers:
Create, test and deploy live patches

Users of service/cloud providers will not be impacted

R: Vulnerability reported to security@xenproject.org
P: Vulnerability pre-disclosed on xen-security-issues@lists.xenproject.org
A: Vulnerability announced on xen-announce@lists.xenproject.org & xenbits.xen.org/xsa
Story 3: Live Patching in Xen Project and XenServer

A tale of close collaboration within the Xen Project Community
What is Live Patching?

Replacing compiled functions with new code, encoded in an ELF file called payload, while the hypervisor is running without impacting running guests.

Design: xenbits.xenproject.org/docs/unstable/misc/livepatch.html
Building Live Patches in Xen
Through livepatch-build-tools (based on kpatch-build)

The exact source tree used to build the running Xen instance. The .config from the original build of Xen. A build-id onto which the livepatch will be applied. A source patch.

livepatch-build-tools

The exact same compilation toolchain used to build the running Xen.

Livepatch payload
Applying Live Patches in Xen
Through xen-livepatch

Supports **stacking** of different payloads; payloads depend on **build-id**

**Functionality:**

* list: lists loaded and applied live patches
* upload: load & verify a live patch
* unload: unload a live patch
* apply: apply a live patch
* revert: un-apply a live patch
## Live Patching in Xen & XenServer

<table>
<thead>
<tr>
<th>Target</th>
<th>Technology</th>
<th>Function + Data</th>
<th>Data Structures</th>
<th>Inline f() patching</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dom0 &amp; Guest Linux Kernel</td>
<td>Kernel Live Patching</td>
<td>✓</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td></td>
<td>kGraft (SUSE)</td>
<td>✓</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td></td>
<td>kPatch (RedHat)</td>
<td>✓</td>
<td>✓ via hooks</td>
<td>✗</td>
</tr>
<tr>
<td></td>
<td>kSplice (Oracle)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Hypervisor</td>
<td>Xen LivePatch</td>
<td>✓ Xen 4.7</td>
<td>✓ Xen 4.8 via hooks</td>
<td>✗ Future</td>
</tr>
</tbody>
</table>

XenServer LivePatch

For Dom0 (CentOS)

For Xen

Integrates different solutions into a single user experience
Live Patches in XenServer

---

**Hot Fixes contain**
Per **valid patch** level: a Xen or Dom0 Live Patch
Matching RPMs for **most recent** patch level
In case of a reboot or for Xen/Dom0 not capable of Live Patching

**Extensive Verification and Validation:**
The process of patching a live hypervisor or kernel is not an easy task. What happens is a little bit like open heart surgery. The patient is the hypervisor and/or Dom0 itself, and precision and care are needed to get things right.

One wrong move and it is game over.

---
Live Patches in XenServer

Update process:
- Download Hot Fix RPM & LPs
- XAPI Toolstack initiates host update
- XenCenter or xe
- Initiates host update
- Works out correct LP & updates
- Running System instance that supports live patching
- Hypervisor
- Dom0
- Dom0 Kernel (CentOS)
- Updates (such that after reboot the patches are applied)

Disk updates (such that after reboot the patches are applied)
**DEMO: Apply Live Patch in XenServer**
xenbits.xenproject.org/people/larsk/LCC17 - Apply LivePatch.mov

**Shows:** a) test that XSA-213 is present (via XTF), b) upload/application of Hotpatch while the hypervisor runs, c) re-run test in (a)
Using Live Patching with XenServer makes Live Patching easy!
If you want to know more ... 

**Demos XenServer**

xenbits.xenproject.org/people/larsk/LCC17 - Build LivePatch.mp4  
(xenbits.xenproject.org/people/larsk/LCC17 - Apply LivePatch.mov)

**Xen Project LivePatch Specification & Status**

xenbits.xenproject.org/docs/unstable/misc/livepatch.html  
wiki.xenproject.org/wiki/LivePatch

**Xen Project LivePatch Presentations & Videos**

xenbits.xenproject.org/people/larsk/FOSDEM17-LivePatch.pdf (Short)  
people/larsk/XPDS16-LivePatch.pdf (Long)

**Xen Project LivePatch Videos**

fosdem.org/2017/schedule/event/iaas_livepatxen/

**XenServer**

xenserver.org
Assessing a FOSS Project’s Security Record

Security Process
Number of Vulnerabilities
Media Coverage
Other Considerations
Vulnerability Disclosure Models

**Responsible Disclosure:** fix critical systems/software **before** publication

- **R:** Vulnerability reported to security@...
- **P:** Vulnerability pre-disclosed to eligible users
- **A:** Vulnerability announced publicly
- **F:** Fix available

**Full Disclosure, post-fix:** public disclosure **with** a fix

**Full Disclosure, immediate (no-fix):** public disclosure **without** a fix
## Vulnerability Process Comparison

<table>
<thead>
<tr>
<th>FOSS Project</th>
<th>Bug Severity</th>
<th>Process Type</th>
<th>Days</th>
<th>Who?</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Linux Kernel</strong> via OSS-security distros</td>
<td>≥ Medium – Critical, ≤ Low</td>
<td>Responsible Disclosure</td>
<td>14-19</td>
<td>D 4</td>
</tr>
<tr>
<td><strong>QEMU (KVM) via QEMU Security Process</strong></td>
<td>≥ Medium – Critical, ≤ Low</td>
<td>Responsible Disclosure</td>
<td>14-19</td>
<td>D 4</td>
</tr>
<tr>
<td><strong>OpenStack OSSA</strong> <strong>OpenStack OSSN</strong></td>
<td>≥ Medium – Critical, ≤ Low</td>
<td>Responsible Disclosure</td>
<td>3-5</td>
<td>D, S, P</td>
</tr>
<tr>
<td><strong>Xen Hypervisor</strong></td>
<td>Low – Critical</td>
<td>Responsible Disclosure</td>
<td>14</td>
<td>D, S, P</td>
</tr>
</tbody>
</table>

1) Is the CVE severity used to handle vulnerabilities differently?
2) Days embargoed (information is classified)
3) D = Distros/Products, S = Public Service, P = Private Downstream
5) [http://www.openwall.com/lists/oss-security or devel list](http://www.openwall.com/lists/oss-security)

Only handles x86 KVM bugs (no ARM or other bugs)
Impacts how long a user (aka you) is at risk
Is my distro/vendor on a pre-disclosure list?
A surprisingly large number of distros are not:
including a few on Linux.com’s Best Linux Distros of 2016 list

Impacts cloud / service providers
As a user, are security issues fixed before public disclosure?

Low Severity vulnerabilities can still be High Risk
Temporal and Environmental CVSS scores are not covered by CVE databases
(neither cvedetails.com or nvd.nist.gov)
Vulnerabilities can be chained together, making the combo High Severity
(e.g. Hot Potato used 3 old unpatched vulnerabilities to gain root access)
Comparing Projects and Products

cvedetails.com (bit.do/guide-cvedetails)
Easy to use interface for vulnerability data
Data from several sources
Browsable by vendor, product, version, type, date…
Vulnerability statistics, trends, reports
BUT: rigid ➔ getting data outside pre-defined vendor/product categories is near-impossible

vulners.com (good guides on slideshare.net – search for vulners)
In many ways more accurate and flexible than cvedetails

```
type:cve AND (description:kvm OR description:qemu) AND published: [2012 TO *] ➔ 307
type:cve AND (description:xen) AND published: [2012 TO *] ➔ 245
```

Works best when used through its API (in particular if you want to visualize the data)
Example: CVE’s per Project

* Data up to Sept 4th, 2017
Vulnerability data from vulners.com
Using Media Coverage to compare Projects: a very bad idea

Data covering September 2016 – September 2017: from vulners.com, mention.com and theregister.com
Other Considerations

Does the Project Look for Vulnerabilities?
Approach to Quality and Testing: e.g. Fuzzing, Audits of components
Does the project award Bug Bounties (e.g. NetBSD)?
Do vendors supporting the project offer Bug Bounties?

Infrastructure related to Vulnerabilities
Transparency: How well are processes documented
Vulnerability Testing: XTF (in Xen)
Vulnerability Tooling: XSATool, XSAMatch (in Xen)
Summary ...
Xen Project:
Only Hypervisor with VMI
Protection from new classes of malware
Several security companies working with XenServer

Live Patching
Disruption free application of vulnerabilities
Used by several cloud providers
Used best in commercial products, e.g. XenServer

Industry Leading Vulnerability Process
Includes QEMU and Kernel XSAs

General:
Tools do assess project’s track records
Not an easy task
Harder for proprietary products due to lack of information
Questions

www.slideshare.net/xen_com_mgr/presentations

Picture by Lars Kurth