Bluepilling the Xen Hypervisor

Joanna Rutkowska & Alexander Tereshkin
Invisible Things Lab

Black Hat USA 2008, August 7th, Las Vegas, NV
Xen Owning Trilogy

Part Three
Previously on Xen 0wning Trilogy...
Part 1: “Subverting the Xen Hypervisor”
by Rafal Wojtczuk (Invisible Things Lab)

- Hypervisor attacks via DMA
- TG3 network card “manual” attack
- Generic attack using disk controller
- “Xen Loadable Modules” framework :)”
- Hypervisor backdooring
  - “DR” backdoor
  - “Foreign” backdoor
Part II: “Detecting and Preventing the Xen Hypervisor Subversions”
by Rafal Wojtczuk & Joanna Rutkowska

✓ Latest Xen security features
✓ How they fail: Q35 exploit
✓ How they fail: FLASK exploit
✓ The need for hypervisor integrity checks!
✓ Introducing HyperGuard!
Now, in this part...
1. **Nested virtualization** ("Matrix inside Matrix")
2. **BluePillBoot**
3. **XenBP**: Bluepilling the Xen hypervisor **on the fly**!
4. Bluepilled Xen **detection**
Nested Virtualization
Idea of how to handle this situation...
Now, let's look at the actual details :)
Let's start with AMD-V...
Looks convincing but we also need to take care about some technical details, that are not trivial...
• Hypervisors expect to have GIF=1 when VMEXIT occurs...
• They might not be prepared to handle interrupts just after VMEXIT from guests!
• ... but when we resume the nested hypervisor CPU sets GIF=1, because we do this via VMRUN, not VMEXIT...
Getting around the “GIF Problem”

- We need to “emulate” that GIF is 0 for the nested hypervisor
- We stop this emulation when:
  - The nested hypervisor executes STGI
  - The nested hypervisor executes VMRUN
- How do we emulate it?
GIF0 emulation

- `VMCBi'.V_INTR_MASKING = 1`
- Host’s `RFLAGS.IF = 0`
- Intercept NMI, SMI, INIT, #DB and held (i.e. record and reinject) or discard until we stop the emulation
Additional details

- Need to also intercept VMLOAD/VMSAVE
- Need to virtualize VM_HSAVE_PA
- ASID conflicts
Hypervisor: ASID = 0

Nested Hypervisor: \textcolor{red}{\textbf{ASID = 1}}
(but thinks that has ASID = 0)

Nested Guest: \textcolor{red}{\textbf{ASID = 1}}
(assigned by the nested hypervisor)

Conflicting ASIDs!
But we can always reassign the ASID in the VMCB “prim” that we use to run the nested guest.
Performance Impact

- One additional `#VMEXIT` on every `#VMEXIT` that would occur in a non-nested scenario.
- One additional `#VMEXIT` when the nested hypervisor executes: STGI, CLGI, VMLOAD, VMSAVE.
- Lots of space for optimization though.
Intel VT-x
Nested virtualization on VT-x

- No GIF bit - no need to emulate “GIF0” for the nested hypervisor :)  
- No Tagged TLB - No ASID conflicts :)  

However:

- VMX instructions can take memory operands - need to use complex operand parser  
- No tagged TLB - potentially bigger performance impact
Nested VT-x: Status

• We have that working!
• The VT-x nesting code cannot be published though :(}
Who else does Nested (hardware-based) Virtualization?
IBM z/VM hypervisor on IBM System z™ mainframe

“Running z/VM in a virtual machine (that is, z/VM as a guest of z/VM, also known as “second-level” z/VM) is functionally supported but is intended only for testing purposes for the second-level z/VM system and its guests (called “third-level” guests).”


IBM System z10, source: ibm.com
Dom0 modifies the MBR
Blue Pill Boot =
  MBR infector +
  Blue Pill loader +
  Blue Pill that supports nested virtualization
Memory

BluePill (stage 1)

BPB’s MBR

BluePill (stage 2)

Original MBR

0x10000

0x0

0x7c00

0x0

Start hardware VM executing from 0x7c00 in Real Mode

Enable Long Mode
Int 15h/e820h queries are intercepted by BluePill

MBR starts Xen which now runs in a hardware virtual machine controlled by the BluePill
Demo: BluePillBooting the Xen

(please excuse the recording quality)
Ensure hypervisor integrity via SRTM or DRTM
Xen Blue Pill
SRTM/DRTM do not protect the already loaded hypervisor!
SRTM/DRTM do not protect the already loaded hypervisor!
The details
Loading using Rafal’s XLM framework...
The diagram illustrates a hypervisor structure with BluePill at the top, containing XBP, xenpgalloc, and xenrunproc. Below the hypervisor are domains, with Dom0 highlighted.
We allocate a block of memory for XBP inside Xen hypervisor -- this memory is used for both the XBP’s code and data and heap.
Demo: Bluepilling the Xen on the fly...
On Xen 3.3 we need to use Q35 exploit instead of direct hdd (see the talk #2)
Bluepilled Xen: Detection
Detecting a VMM is now not enough...
... as we know there is already one VMM in the system already (i.e. the Xen)...
We can only try direct timing analysis to see if #VMEXITs will take longer time to execute... (then on “non-bluepilled” Xen)
Impact on PV domains
We don’t need to intercept anything besides VMRUN (and optionally VMLOAD, VMSAVE, STGI, CLGI) -- all those instructions cause #GP when executed in PV guests (including Dom0)
On AMD!

On Intel we have obligatory intercepts (CPUID, INVD, MOV CR3).
Impact on HVM domains
HVM domains: impact on \#vmexit time (RDMSR intercept on AMD)

- Full Nested Virtualization
- VMCB rewriting
- CLGI/STGI interception
- VMLOAD/VMSAVE interception
- Native Xen (baseline)
5k cycles (Native Xen) vs. 7k cycles (Bluepilled Xen)
2000 cycles from the Holy Grail ;}
But that you can observe only in a HVM domain; on PV domains it is: 0 cycles (on AMD)!
HyperGuard vs. BluePill?
Summary
(of the whole trilogy)
Talk #1 (Rafal)
Modifying Xen via DMA attacks
“Xen Loadable Modules” Framework
Hypervisor Rootkits/Backdoors for Xen
(don’t confuse with virtualization-based rootkits!)
Talk #2 (Joanna & Rafal)
DMA protections (IOMMU/VT-d) on recent Xens
Getting around VT-d Xen protection

(BONUS: on the fly SMM modification, despite D_LCK set)
Other Xen protection mechanisms...
... and how they sometimes might be bypassed...

Exploiting a heap overflow in Xen hypervisor
HyperGuard - integrity scanner for a hypervisor
Talk #3 (Alex & Joanna)
Hardware Nested Virtualization
Blue Pill Boot
Xen Blue Pill: Bluepilling the Xen on the fly
Discussed the XBP detection
Slides available at:
http://invisiblethingslab.com/bh08

Demos and code will be available from the same address after Intel releases the patch.
Thank you!