Self-service Cloud Computing

Published in Proceedings of ACM CCS’12

Shakeel Butt
shakeelb@cs.rutgers.edu

Abhinav Srivastava
abhinav@research.att.com

H. Andres Lagar-Cavilla
andres@lagarcavilla.org

Vinod Ganapathy
vinodg@cs.rutgers.edu

• Many new companies & services rely exclusively on the cloud, e.g., Instagram, MIT/Harvard EdX [NYTimes, “Active in Cloud, Amazon Reshapes Computing,” Aug 28, 2012]
Virtualized cloud platforms

Management VM (dom0)

Hypervisor

Hardware

Examples: Amazon EC2, Microsoft Azure, OpenStack, RackSpace Hosting
Embracing the cloud

Lets do Cloud
Embracing the cloud

Trust me with your code & data

You have to trust us as well

Problem #1

Client code & data secrecy and integrity vulnerable to attack

Google Fires Employee Accused Of Spying On Kids
By Phil Villareal on September 16, 2010 9:15 AM
Embracing the cloud

Problem #1

Client code & data secrecy and integrity vulnerable to attack
Embracing the cloud

Problem #2

I need customized malware detection and VM rollback

For now just have checkpointing ...

Clients must rely on provider to deploy customized services
Why do these problems arise?

Management VM (dom0)

Work VM

Work VM

Work VM

Hypervisor

Hardware
Example: Malware detection

[Example: Gibraltar -- Baliga, Ganapathy, Iftode, ACSAC’08]
Clients must rely on provider to deploy customized services
Problem

Client code & data secrecy and integrity vulnerable to attack

Client’s VM

Code

Data

Malicious cloud operator

Process the page

Sec. Policy

Resume guest

Alert user

1

Hypervisor
Problem

Client code & data secrecy and integrity vulnerable to attack

Client’s VM

| Code | Data |

Management VM

Checking daemon

Process pages

Sec. Policy

EXAMPLES:

• CVE-2007-4993. Xen guest root escapes to dom0 via pygrub
• CVE-2007-5497. Integer overflows in libext2fs in e2fsprogs.
• CVE-2008-0923. Directory traversal vulnerability in the shared folders feature for VMWare.
• CVE-2008-1943. Buffer overflow in the backend of XenSource Xen paravirtualized frame buffer.
• CVE-2008-2100. VMWare buffer overflows in VIX API let local users execute arbitrary code in host OS.

.... [AND MANY MORE]
Traditional cloud computing

Management VM

Client’s VMs

Hypervisor

Hardware
SSC: Self-service cloud computing

Management VM

Client’s VMs

Hypervisor

Hardware
Main contributions

• New hypervisor privilege model
• Enables four new cloud abstractions
  – **Udom0**: Per-client management VMs
  – **Sdom0**: System-wide management VM
  – **Service VMs**
  – **Mutually-trusted service VMs**
• Protocols for trustworthy VM startup
• Novel cloud-based services
Duties of the management VM

Manages and multiplexes hardware resources

Manages client virtual machines

Management VM (Dom0)
Main technique used by SSC

Disaggregate the management VM

System-wide Mgmt. VM (SDom0)

- Manages hardware
- No access to clients VMs

Per-Client Mgmt. VM (UDom0)

- Manages client’s VMs
- Allows clients to deploy new services

Solves problem #1

Solves problem #2
An SSC platform

- **SDom0**
- **UDom0**
- **Service VM**
- **Client’s meta-domain**
  - Work VM
  - Work VM

**SSC Hypervisor**

**Hardware**

**TPM**

**Trusted Computing Base**
1. Separation of Privilege
2. Least Privilege

SDom0

UDom0

Service VM

Work VM

Work VM

SSC Hypervisor

Hardware
But providers want some control

- Udom0 and service VMs put clients in control of their VMs
- Sdom0 cannot inspect these VMs
- Malicious clients can misuse privilege
- Mutually-trusted service VMs

NO data leaks or corruption

NO illegal activities or botnet hosting
Trustworthy regulatory compliance

SSC Hypervisor

Hardware

Mutually-trusted Service VM

Work VM

Work VM

UDom0

SDom0
Traditional privilege model

Privileged operation

Hypervisor

is request from Management VM?

YES  NO

ALLOW  DENY
SSC’s privilege model

Privileged operation

Self-service hypervisor

Is the request from client’s Udom0?

YES

Does requestor have privilege (e.g., client’s service VM)

YES

ALLOW

NO

DENY

ALLOW

NO

ALLOW
Bootstrap: the Domain Builder

SSC Hypervisor

Hardware

SDom0

Domain Builder

UDom0

Work VM

Service VM
Bootstrap: the **Domain Builder**

Must establish an encrypted communication channel

**SSC Hypervisor**

**Hardware**
1. Udom0 image, Enc (1, 0)
3 DomB installs key, nonce

Enc (key, nonce)

Domain Builder

UDom0

SSC Hypervisor

Hardware
Client gets TPM hashes
5 Udom0 sends to client
6. Client sends Udom0 SSL key.
SSL handshake and secure channel establishment

7

Hardware

SSC Hypervisor

Domain Builder

UDom0

Hardware
Can boot other VMs securely

SSC Hypervisor

Hardware

UDom0

Domain Builder

Work VM

Service VM

VM image

SSL

8
Client meta-domains

- Mutually-trusted Service VMs
  - Regulatory compliance
  - Trustworthy metering

Service VMs
- Storage services
- Firewall and IDS
- Malware detection

Computation
- Work VM
- Work VM
- Work VM

Udom0

SSC hypervisor

Hardware
Case studies: Service VMs

• Storage services: Encryption, Intrusion detection

• Security services:
  – Kernel-level rootkit detection
  – System-call-based intrusion detection

• Data anonymization service

• Checkpointing service

• Memory deduplication

• And compositions of these!
Evaluation

• Goals
  – Measure overhead of SSC

• Dell PowerEdge R610
  – 24 GB RAM
  – 8 Xeon cores with dual threads (2.3 GHz)
  – Each VM has 2 vCPUs and 2 GB RAM

• Results shown only for 2 service VMs
  – See our CCS’12 paper for more
Storage encryption service VM

<table>
<thead>
<tr>
<th>Platform</th>
<th>Unencrypted (MB/s)</th>
<th>Encrypted (MB/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Xen-legacy</td>
<td>81.72</td>
<td>71.90</td>
</tr>
<tr>
<td>Self-service</td>
<td>75.88</td>
<td>70.64</td>
</tr>
</tbody>
</table>
Checkpointing service VM

<table>
<thead>
<tr>
<th>Platform</th>
<th>Unencrypted (sec)</th>
<th>Encrypted (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Xen-legacy</td>
<td>1.840</td>
<td>11.419</td>
</tr>
<tr>
<td>Self-service</td>
<td>1.936</td>
<td>11.329</td>
</tr>
</tbody>
</table>
## Related projects

<table>
<thead>
<tr>
<th><strong>CloudVisor</strong> [SOSP’11]</th>
<th><strong>Xen-Blanket</strong> [EuroSys’12]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Protect client VM data from Dom0 using a thin, bare-metal hypervisor</td>
<td>Allow clients to have their own Dom0s on commodity clouds using a thin shim</td>
</tr>
</tbody>
</table>

![Diagram](image)

**CloudVisor**

![Nested Hypervisor]

**Client VM**

**Dom0**

**CloudVisor**

**Xen-Blanket**

**Cloud Hypervisor**

**Cloud Dom0**

**Client Dom0**

**Client VM**
Current and future work

• Novel network services, e.g., trustworthy network traffic metering

• VM migration in an SSC-based cloud:
  – Co-location of service VMs and work VMs.
  – Without exposing details of cloud platform to clients
  – Pricing and metering issues

• Cloud market model:
  – Service VMs as *cloud apps*