On the Control Plane of a Self-service Cloud Platform

**Shakeel Butt** - Rutgers University & NVidia

**Vinod Ganapathy** - Rutgers University

**Abhinav Srivastava** - AT&T Labs Research
Client security on cloud platforms

Cloud providers and administrators are all powerful. Clients have little choice but to trust them.
Implications – Attacks

Client data exposed to attack

- Malicious attacks perpetrated by employees:
  - “Insider” attacks by cloud provider’s employees
  - Cited as important concern in [Gartner 2008]
- Exploits against cloud admin interfaces:
Implications – Deploying services

**Clients have limited flexibility**

- Cloud clients largely restricted to in-VM security tools
- Deployment and configuration of powerful security tools entrusted to cloud provider:
  - VM introspection tools
  - Network-level middleboxes
Our Solution – SSC [ACM CCS 2012]

Self-service Cloud Computing

• De-privilege cloud admins
• Transfer privilege to clients
• Main ideas:
  – Privilege separation
  – Least privilege
• Implemented via hypervisor modifications
Contributions of this paper

Control plane for a cloud platform consisting of SSC hypervisors

<table>
<thead>
<tr>
<th>Client’s Perspective</th>
<th>Provider’s Perspective</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deploying custom network middleboxes</td>
<td>Unified administrative interface</td>
</tr>
<tr>
<td>Specifying VM dependencies</td>
<td>VM dependency-aware migration</td>
</tr>
</tbody>
</table>

Client’s Perspective

- Deploying custom network middleboxes
- Specifying VM dependencies

Provider’s Perspective

- Unified administrative interface
- VM dependency-aware migration
Traditional cloud platforms

Provider’s Management VM (dom0)

Client VM
Client VM
Client VM

Hypervisor

Hardware
Privilege allocation

Provider’s Management VM (dom0)  
Hypervisor  
Hardware

Client VM  
Client VM  
Client VM
Example: Malware detection

Client’s VM

Management VM

Checking daemon

Process the page

Sec. Policy

Hypervisor

Code

Data

1

2

3

Resume guest

Alert user

Process the page

Alert user

Resume guest

Example: Malware detection
Flexibility Problem

Clients rely on provider/admins to deploy the service

1. Hypervisor

Client’s VM

- Code
- Data

Management VM

Checking daemon

- Process the page

- 2
- 3

Sec. Policy

Process

- Resume guest
- Alert user
Security Problem

Client code & data secrecy and integrity vulnerable to attack

Client’s VM

- Code
- Data

Checking daemon

Malicious cloud operator

Hypervisor

- Resume guest
- Alert user

Sec. Policy

Managed VM

Process the page

Malicious cloud operator

Problem

Client code & data secrecy and integrity vulnerable to attack
Privilege allocation in SSC

Provider’s Management VM

Client Mgmt. VM

Client VM

Client VM

SSC Hypervisor

Hardware
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

**SDom0**
- Manages hardware
- No access to client VMs

**UDom0**
- Manages client’s VMs
- Allows clients to deploy new services

System-wide Mgmt. VM (one per physical host)

Per-Client Mgmt. VM
An SSC platform

SSC Hypervisor

Hardware

Trusted Computing Base

SDom0

UDom0

Security VM

Work VM
Cloud control plane

Cloud Controller

Node Controller
Dom0

Hypervisor

Client's interface to the cloud

Client VM

Client VM

Client

VM images

Client's interface to the cloud
Cloud control plane

• From the client’s perspective:
  – Interfaces with the client to get VM images
  – Is the client’s administrative interface

• From the cloud provider’s perspective:
  – Manages VM placement and migration
  – Abstracts away platform details, hiding them from the client’s view
Need for an SSC control plane

Traditional control plane software unaware of SSC abstractions

Two implications:

1. **Poor flexibility:**
   - Client cannot specify VM dependencies
   - Client cannot specify middlebox placement

2. **Poor security:**
   - Udom0s on individual platforms may expose cloud provider topology to malicious clients
SSC-aware control plane

• Enhanced dashboard interface to abstract details of individual Udom0s

• Allows specification of:
  – VM dependency constraints
  – Middlebox placement topologies

• Transparently handles VM migration and placement
  – Please see paper for details on our VM migration protocol
SSC-aware control plane

Cloud Controller

Node Controller

Sdom0

Client switches

UDom0

Client VM

SSC Hypervisor

Client’s interface to the cloud

VM images

Client
Client interface

Cloud Controller

Node Controller

Sdom0

SSC Hypervisor

Client switches

UDom0

Client VM

Client’s interface to the cloud

VM images

Client
Example scenario

Web Server
(web_vm)
Example scenario

Web Server  
(web_vm)  

VMI tool  
(vmi_vm)  

MUST_COLOCATE
Example scenario

SSL Proxy (ssl_vm) \rightarrow \text{Web Server (web_vm)} \rightarrow \text{VMI tool (vmi_vm)}

\text{MUST_COLOCATE} \quad \text{MUST_COLOCATE}
Example scenario

- SSL Proxy (ssl_vm)
- Web Server (web_vm)
- VMI tool (vmi_vm)

Colocation requirements:
- MUST COLOCATE: SSL Proxy (ssl_vm) and Web Server (web_vm)
- MUST COLOCATE: Web Server (web_vm) and VMI tool (vmi_vm)
- MAY COLOCATE: Firewall (firewall_vm)
VM dependency constraints

VM web_vm; // Client’s Web server
VM vmi_vm; // VMI-based Memory introspection tool
VM ssl_vm; // SSL proxy for the Web server
VM firewall_vm; // VM running the Snort NIDS

web_vm.name = “MyWeb”; web_vm.image = Apache.img;
vmi_vm.name = ...; vmi_vm.image = ...;
ssl_vm.name = ...; ssl_vm.image = ...;
firewall_vm.name = ...; firewall_vm.image = ...;

Grant_Privilege (vmi_vm, web_vm, Kern_Mem);
Set_Backend (ssl_vm, web_vm, NET, MUST_COLOCATE);
Set_Backend (firewall_vm, ssl_vm, NET, MAY_COLOCATE);
Cloud controller

Client’s interface to the cloud

Node Controller
Sdom0

Client switches
UDom0
Client VM

SSC Hypervisor

Client
VM images
Cloud controller’s tasks

- Solves a constraint-satisfaction problem
  - All **MUST_COLOCATE** constraints satisfied
  - Output is a set of VM placements
- Communicates VM placements to individual node controllers
  - Sends network switch configurations for backend VMs (**Set_Backend**)
  - Also sends permission requirements for VMs (**Grant_Privilege**)
Udom0 and switches

Cloud Controller

Node Controller

Sdom0

SSC Hypervisor

Client switches

UDom0

Client VM

Client’s interface to the cloud

VM images

Client
Udom0 and switches

- Each physical host that runs a client VM has a Udom0 and software switches
  - We use Open vSwitch for switches
- Udom0 handles Grant_Privilege requests, and enables system services
- Software switches configured to handle Set_Backend requests and accommodate middleboxes
Example of middlebox placement

Host A

Open vSwitch VM

firewall_vm

Inbound traffic

Traffic scanned by firewall_vm

Host B

Open vSwitch VM

ssl_vm

web_vm
Evaluation

• Goals
  – Measure overhead of control plane components

• Dell PowerEdge R610 running Xen-4.3
  – 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 one case study
  – See our paper for more
Baseline overhead for middleboxes

SAMEHOST

Measurement host

Ping requests

Middlebox

Client VM

DIFFHOST

Measurement host

Open vSwitch

Open vSwitch

Middlebox

Client VM
Baseline overhead for middleboxes

### SAMEHOST

<table>
<thead>
<tr>
<th>Setup</th>
<th>Throughput (Mbps)</th>
<th>RTT (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Traditional</td>
<td>925.4 ± 0.5</td>
<td>0.38</td>
</tr>
<tr>
<td>SSC</td>
<td>924 ± 1.2</td>
<td>0.62 (1.6x)</td>
</tr>
</tbody>
</table>

### DIFFHOST

<table>
<thead>
<tr>
<th>Setup</th>
<th>Throughput (Mbps)</th>
<th>RTT (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Traditional</td>
<td>848.4 ± 11.2</td>
<td>0.69</td>
</tr>
<tr>
<td>SSC</td>
<td>425.8 ± 5.5</td>
<td>1.6 (2.3x)</td>
</tr>
</tbody>
</table>
Example: Network metering
## Network metering overhead

<table>
<thead>
<tr>
<th>Setup</th>
<th>Throughput (Mbps)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Traditional</td>
<td>924.8 ± 1.1</td>
</tr>
<tr>
<td>SSC</td>
<td>924 ± 0.4</td>
</tr>
</tbody>
</table>

### SAMEHOST

<table>
<thead>
<tr>
<th>Setup</th>
<th>Throughput (Mbps)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Traditional</td>
<td>845.4 ± 11.1</td>
</tr>
<tr>
<td>SSC</td>
<td>424.3 ± 3.1</td>
</tr>
</tbody>
</table>

### DIFFHOST

---

36
See paper for more

• Network intrusion detection
• Network access control
• Host+network (hybrid) intrusion detection
• Evaluation of VM migration overheads
Related work

• Client security:
  – CloudVisor [SOSP’11], Xoar [SOSP’11], Intel SGX, Haven [OSDI’14], Overshadow [ASPLOS’08]

• Client flexibility with nested VMs:
  – XenBlanket [EuroSys’12]

• Client-controlled middleboxes with SDN:
  – SIMPLE [SIGCOMM’13], FlowTags [NSDI’14], CloudNaaS [SOCC’11]
Thank You.

Shakeel Butt - shakeelb@cs.rutgers.edu
Vinod Ganapathy - vinodg@cs.rutgers.edu
Abhinav Srivastava - abhinav@research.att.com
BACKUP SLIDES
SSC versus Haven/Intel SGX

• SGX allows clients to create *enclaves*, which are opaque to cloud providers

• Benefits of Intel SGX over SSC:
  – Cloud provider is untrusted
  – Ability to defend against memory snooping
  – Strong, cryptographic security guarantees

• Benefits of SSC over Intel SGX:
  – Mutually-trusted domains allow provider to monitor client
  – Mimics cloud setting of VMs over hypervisors
# Cloudvisor and XenBlanket

<table>
<thead>
<tr>
<th><strong>CloudVisor</strong> [<em>SOSP’11]</em></th>
<th><strong>Xen-Blanket</strong> [<em>EuroSys’12]</em></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 showing Cloudvisor and XenBlanket](image_url)
Providers want some control

- **NO** data leaks or corruption
- **NO** illegal activities or botnet hosting

- 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**
Traditional privilege model

Privileged operation

Hypervisor

is request from Management VM?

YES

ALLOW

NO

DENY
SSC’s privilege model

Privileged operation

Self-service hypervisor

Is the request from client’s Udom0?

YES

ALLOW

NO

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

YES

ALLOW

NO

DENY