
Trent Jaeger
Penn State University
Can customers move their services and validate that they still protect data security?
Reasons to Doubt

- History has shown they are **vulnerable to attack**
  - SLAs, audits, and armed guards offer few guarantees
  - **Insiders** can subvert even hardened systems

Data Loss Incidents

<table>
<thead>
<tr>
<th>Year</th>
<th>Incidents</th>
</tr>
</thead>
<tbody>
<tr>
<td>'06</td>
<td>641</td>
</tr>
<tr>
<td>'07</td>
<td>770</td>
</tr>
<tr>
<td>'08</td>
<td>986</td>
</tr>
<tr>
<td>'09</td>
<td>695</td>
</tr>
<tr>
<td>'10</td>
<td>678</td>
</tr>
<tr>
<td>'11</td>
<td>903</td>
</tr>
</tbody>
</table>

Incident Attack Vector

- **External**: 54%
- **Accidental**: 23%
- **Insider**: 16%
- **Unknown**: 7%

Credit: The Open Security Foundation [datalossdb.org](http://datalossdb.org)
Cloudy Future

• New problem or new solution?
  ‣ New challenges brought on by the cloud (plus old ones)
  ‣ Utility could provide a foundation for solving such challenges
Cloudy Future

• Improve on data centers? On home computing?
  ‣ Seems like a low bar
Consumers use published instances [CCS 2011]

Instances may be flawed - have adversary-controlled public and private keys
Security Configuration

- **Zillions of security-relevant configurations for instances**
  - Firewalls
  - Mandatory access control
    - SELinux, AppArmor, TrustedBSD, Trusted Solaris, MIC
  - Discretionary access control
  - Application policies (e.g., Database, Apache)
  - Pluggable Authentication Modules (PAM)
  - Application configuration files
  - Application code enforces security

- **Plus new configuration tasks for the cloud - e.g., storage**
Insiders

- Although the vendor may have a good reputation, not every employee may trust us as well.

Trust me with your code & data

You have to trust us as well

Client  Cloud Provider

Cloud operators
Side Channels

- Shared infrastructure leads to visibility for others
  - You can’t monitor, but others can
- Get Off My Cloud - Ristenpart et al. [CCS 2009]
  - Caches (Memory)
  - Devices (I/O)
  - CPU
  - Scheduling
- Ari Juels -- “Many of the security implications of the cloud stem from tenants entrusting computing resources to a third party that they controlled in the past.”
- Not really going to discuss this further
Policy-Sealed Data: A New Abstraction for Building Trusted Cloud Services

Nuno Santos¹, Rodrigo Rodrigues², Krishna P. Gummadi¹, Stefan Saroiu³
MPI-SWS¹, CITI / Universidade Nova Lisboa², Microsoft Research³
Managing the Cloud is Complex & Error-Prone

Is my data properly managed?

Cloud software admins. can compromise customers’ data

Customer

Data

Cloud Software Administrator

Cloud Provider
1. Newer hypervisors can offer protection from SW admins.
   - e.g., nested virtualization: CloudVisor [SOSP’11], Credo [MSR-TR]

2. Trusted computing can attest cloud node runs “correct” hypervisor
   - Trusted Platform Module (TPM)

But, TPMs alone ill-suited for the cloud
Our Contributions

1. **Policy-sealed data abstraction**
   - Data is handled only by nodes satisfying customer-chosen policy
   - Examples:
     - Handle data only by nodes running CloudVisor
     - Handle data only by nodes located in the EU

2. **Use attribute-based encryption (CP-ABE) to implement abstraction efficiently**
   - Binds policies and node attributes to node configurations
   - Ciphertext-Policy Attribute-Based Encryption [Bethencourt07]

---

Excalibur incorporates both contributions
Excalibur Addresses TPM Limitations in Cloud

Policy-sealed data
- Enables flexible data migration across cloud nodes
  - Customer data accessible to any node that satisfies the customer policy

Attribute-based encryption
- Hides node’s identities and low-level details of the software
  - Only high-level attributes are revealed
- Masks TPMs’ poor performance
  - Enforcing policies does not require direct calls to TPMs
Policy-Sealed Data

星光 Seal
加密和绑定数据到策略

政策加密数据

星光 Unseal
解密数据如果节点满足策略

星光 Seal to:
visor = “secure visor”

Customer

Provider

政策加密

Hypervisors

Secure

Commodity
Side Channels

- Shared infrastructure leads to visibility for others
  - You can't monitor, but others can

- Get Off My Cloud - Ristenpart et al. (CCS 2009)
  - Caches (Memory)
  - Devices (I/O)
  - CPU
  - Scheduling

- Ari Juels -- "Many of the security implications of the cloud stem from tenants entrusting computing resources to a third party that they controlled in the past."

- Not really going to discuss this further

---

Excalibur Architecture

- Check node configurations
  - Monitor attests nodes in background

- Scalable policy enforcement
  - CP-ABE operations at client-side lib

[Diagram of Excalibur Architecture]

- Customer
- Policy-Sealed Data
- Monitor
- Datacenter

- Seal
- Unseal
- Attest & send credential

---

13
Excalibur Mediates TPM Access w/ Monitor

Monitor goals:

- Track node ids + TPM-based attestations
  - Hides low-level details from users

- Track nodes’ attributes that cannot be attested via today’s TPMs
  - e.g., nodes’ locations (EU vs. US)

- Form the cloud’s root of trust
  - Customers only need to attest the monitor’s software configuration
**IaaS Cloud Platform**

- IaaS clouds rely on a variety of *cloud services* to provision and manage *users’ data* (e.g., VM and container)

---

**IaaS service vendors:**

- Amazon Web Services
- Google Compute Engine
- IBM Bluemix
- Microsoft Azure
- Rackspace

**IaaS software stack vendors:**

- OpenStack
- Eucalyptus
- CloudStack
- OpenNebula

---

Systems and Internet Infrastructure Security Laboratory (SIIS)
Vulnerabilities in Cloud Services

### CVE Details

**The ultimate security vulnerability datasource**

**Openstack : Security Vulnerabilities**

<table>
<thead>
<tr>
<th>CVE ID</th>
<th>CWE ID</th>
<th># of Exploits</th>
<th>Vulnerability Type(s)</th>
<th>Publish Date</th>
<th>Update Date</th>
<th>Score</th>
<th>Gained Access Level</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>CVE-2016-7498</td>
<td>399</td>
<td>DoS</td>
<td></td>
<td>2016-09-27</td>
<td>2016-09-28</td>
<td>6.8</td>
<td>None</td>
<td>Remote</td>
</tr>
<tr>
<td>CVE-2016-5363</td>
<td>254</td>
<td>DoS Bypass</td>
<td></td>
<td>2016-06-17</td>
<td>2016-06-20</td>
<td>6.4</td>
<td>None</td>
<td>Remote</td>
</tr>
<tr>
<td>CVE-2016-5362</td>
<td>254</td>
<td>DoS Bypass</td>
<td></td>
<td>2016-06-17</td>
<td>2016-06-21</td>
<td>6.4</td>
<td>None</td>
<td>Remote</td>
</tr>
<tr>
<td>CVE-2016-2140</td>
<td>200</td>
<td>+Info</td>
<td></td>
<td>2016-04-12</td>
<td>2016-04-21</td>
<td>3.5</td>
<td>None</td>
<td>Remote</td>
</tr>
<tr>
<td>CVE-2016-0757</td>
<td>284</td>
<td></td>
<td></td>
<td>2016-04-13</td>
<td>2016-04-18</td>
<td>4.0</td>
<td>None</td>
<td>Remote</td>
</tr>
<tr>
<td>CVE-2016-0739</td>
<td>399</td>
<td>DoS</td>
<td></td>
<td>2016-01-29</td>
<td>2016-03-03</td>
<td>5.0</td>
<td>None</td>
<td>Remote</td>
</tr>
</tbody>
</table>

- **Total number of vulnerabilities**: 162
- **Page**: 1

**Over 150 vulnerabilities reported!**
Attacks via Service Vulnerabilities

- Cloud services run with all users’ permissions, and even cloud vendor’s permission
  - Confused deputy attacks
  - Inadvertent or intentional data leakage
  - Problem compounded by the need of cloud services to make critical security decisions over users’ data
Attacks via Flawed Trust Assumption

- Cloud services *fully trust each other*
  - Once an adversary controls a cloud service or node (e.g., via hypervisor vulnerabilities), she can perform arbitrary operations on benign cloud nodes via cloud service interactions
  - Compromise of one cloud service can lead to data compromise cloud wide
  - A user’s TCB includes each and every cloud service & node

![Diagram showing cloud node operations](image-url)
Insight

• Cloud services themselves *cannot control data propagation* due to vulnerabilities
  • Information flow control (IFC) over cloud services

• Compromised cloud services and nodes have *unlimited access* to any user’s data on any cloud node
  • Bound the data accessibility of a cloud node to the users that are using (thus trusting) the cloud node
  • **Decentralized security principle**: a user’s data security does not depend on system components that the user does not trust [Arden 2012]
DIFC over Cloud Services

- Enforce *Decentralized Information Flow Control* (DIFC) over cloud services to mitigate cloud service vulnerabilities
  - Confine cloud services to individual users’ security labels
  - Cloud services must *explicitly declassify or endorse* data using *ownerships*

Isolating cloud users

Export protection
Control of Cloud Services

- Cloud services (stateless) —> ephemeral *event handlers*
  - [Insight] Cloud services are constructed using event dispatch loop [Efstath. 05]
  - *Dispatcher* on a cloud node spawns event handlers on-demand with users’ labels
Spawning Event Handlers

- Requirements:
  - **[who can spawn]** prevent nodes that do not have a user’s authority from spawning event handlers that may access that user’s data
  - **[where can it spawn]** prevent nodes that fail to satisfy cloud policy (e.g., Col) from being selected to execute the user’s event handler
  - **[best place to spawn]** find the “ideal” cloud node to spawn

\[\text{Spawn} \]

- Coke’s API
  - \(S = \{a\}, I = \{a\}\)
  - Cloud Node

- Coke’s Compute
  - \(S = \{a\}, I = \{a\}\)
  - Cloud Node

- Pepsi’s Compute
  - \(S = \{a\}, I = \{a\}\)
  - Cloud Node
Daemon (dispatcher) needs to be delegated with authority to spawn new event handlers with ownership capabilities:

- authority = \{ownership, node, auth\}
- Having the authority indicates the node is trusted by the user
Capability (Ownership) Delegation

- **centralized control** over authority distribution:
  - [who can spawn] only a cloud node trusted with a user’s authority can spawn on other cloud nodes with the user’s label
  - [where can it spawn] enforces mandatory cloud policy (e.g., ICAP)
  - [best place to spawn] most compatible security requirements
Mitigating Vulnerabilities

- Pileus blocks 6* zero-day OpenStack vulnerabilities that were newly reported after Pileus’s deployment

<table>
<thead>
<tr>
<th>CVE ID</th>
<th>Affected Cloud Service</th>
<th>Mitigated</th>
</tr>
</thead>
<tbody>
<tr>
<td>CVE-2015-1195</td>
<td>Image Service (Glance)</td>
<td>Yes</td>
</tr>
<tr>
<td>CVE-2015-1850</td>
<td>Volume Service (Cinder)</td>
<td>Yes</td>
</tr>
<tr>
<td>CVE-2015-3221</td>
<td>Network Service (Neutron)</td>
<td>No*</td>
</tr>
</tbody>
</table>

Systematic mitigation of 1/3 vulnerabilities reported in OpenStack
OpenStack on Pileus

- Pileus does not block normal cloud operations
  - Cloud services are confined as-is in majority of cloud operations
  - Few requires declassification and endorsement
    - When an operation causes *data flow across user boundaries* (i.e., resource sharing)
- Declassifiers and endorsers are simple
  - Volume declassifier (50 SLOC), image endorser (150 SLOC)

<table>
<thead>
<tr>
<th>Type</th>
<th>Number of cloud operations</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>DIFC-aware</td>
<td>13</td>
<td>nova boot</td>
</tr>
<tr>
<td>DIFC-unware</td>
<td>135</td>
<td>nova volume-attach</td>
</tr>
<tr>
<td>Total</td>
<td>148</td>
<td></td>
</tr>
</tbody>
</table>
Conclusion

• Pileus is a model and system that protects users’ data from vulnerable cloud services
  • It mitigates cloud service vulnerabilities by enforcing Decentralized Information Flow Control (DIFC)
  • It addresses the mutual trust assumed by cloud services and nodes by enforcing Decentralized Security Principle (DSP)