Computer Meteorology: Monitoring Compute Clouds

Lionel Litty, H. Andrés Lagar-Cavilla, David Lie

University of Toronto
Infrastructure as a Service (IaaS)

Examples:
- Amazon EC2
- GoGrid
- Mosso
- ...
Q: Rate the **challenges/issues** ascribed to the 'cloud'/on-demand model

(1=not significant, 5=very significant)

- **Security**: 74.6%
- **Performance**: 63.1%
- **Availability**: 63.1%
- **Hard to integrate with in-house IT**: 61.1%
- **Not enough ability to customize**: 55.8%
- **Worried on-demand will cost more**: 50.4%
- **Bringing back in-house may be difficult**: 50.0%
- **Regulatory requirements prohibit cloud**: 49.2%
- **Not enough major suppliers yet**: 44.3%

Source: IDC Enterprise Panel, August 2008  n=244
Security

• Miscreants can abuse the cloud provider’s resources:
  – Spam.
  – Use infrastructure to attack other computers.
  – Hosting illegal content.

• This has consequences for the cloud provider:
  – Damage to reputation.
  – Technical consequences: Shared IPs blacklisted.
  – Potential legal concerns.
Solutions?

Network monitoring (NM) has limitations:

- Encrypted traffic
- Stealthy malicious traffic

Distributed attack using botnet.
ISPs use NM and have done poorly.

Unlike ISPs, cloud providers control the execution platform:
Can they use this to their advantage?
Introspection

Reductionist approach: understand a complex system by understanding its parts.

• Identify processes.
• Analyze the behavior of each process.
Non-malicious and Malicious VMs

- Non-malicious: may be vulnerable, not yet compromised.
- Malicious: under miscreant control.
  - Attacker can blur boundaries between processes.
- Tamper-evident monitor:
  - Either report accurate information
  - Or report that it cannot obtain accurate information.
Introspection properties

• Power
  Can it see everything?

• Robustness
  Is it resilient to changes in the monitored system?

• Unintrusiveness
  Can it negatively impact the monitored system?
Host agent

- Power
- Robustness
- Unintrusiveness

Cloud provider infrastructure

VMM

VM’s OS

Customer VM

Process

Process

Host agent
Host agent w/ driver

Power
Robustness
Unintrusiveness

Cloud provider infrastructure

VMM

VM’s OS
Driver

Process
Process

Host agent

Customer VM
Trap & Inspect

- Power
- Robustness
- Unintrusiveness

Customer VM
- Process
- Process
- Process

VM's OS
- Traps

VMM
- Introspect ion code

Cloud provider infrastructure
Checkpoint & Rollback

Cloud provider infrastructure

Power
Robustness
Unintrusiveness

Customer VM
Process
Process
Process

VM’s OS

Traps

VMM

Introspection code

Introspection code

Introspection code
Architectural Introspection

Power
Robustness
Unintrusiveness

Cloud provider infrastructure

VMM

Customer VM
Process
Process
Process

VM’s OS

Introspection code
### Summary of introspection approaches

<table>
<thead>
<tr>
<th></th>
<th>Power</th>
<th>Unintrusiveness</th>
<th>Robustness</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Host agent</strong></td>
<td>Good</td>
<td>Poor</td>
<td>Good</td>
</tr>
<tr>
<td><strong>Host agent w/ driver</strong></td>
<td>Best</td>
<td>Worst</td>
<td>Poor</td>
</tr>
<tr>
<td><strong>Trap &amp; Inspect</strong></td>
<td>Best</td>
<td>Best</td>
<td>Worst</td>
</tr>
<tr>
<td><strong>Checkpoint &amp; Rollback</strong></td>
<td>Best</td>
<td>Best</td>
<td>Poor</td>
</tr>
<tr>
<td><strong>Architectural monitoring</strong></td>
<td>Poor(?)</td>
<td>Best</td>
<td>Best</td>
</tr>
</tbody>
</table>
Introspection example

• Goal:
  – Which applications are run by a customer VM?
  – What’s the version of these applications?

• Why?
  – Detect malicious code
  – Inform customer of vulnerable code
  – Deploy vulnerability-specific filters
Execution monitoring

- Goal: Identify all running binary code in a VM.
- Examples
  - Host agent: /proc, Process Explorer
  - Trap & inspect: examine OS data structures
  - Architectural monitoring: leverage MMU to identify all executing code
Execution monitoring

Customer VM

Process

Process

Process Page fault

VM’s OS

VMM
File monitoring

• Goal: What byte code is Java executing? What about the PHP interpreter?

• Examples:
  – Host-based: strace, filemon
  – Trap & inspect: examine OS data structures
  – Architectural monitoring: taint-tracking?
File Monitoring

Customer VM

- Process
- Process
- Process

VM’s OS

VMM

Script
Conclusion

• Architectural introspection should be used when possible.

• More research is needed to explore the range of events that can be monitored using Architectural introspection.

• Cloud providers should be mindful of the limitations of introspection.