Privacy-Sensitive VM Retrospection

Wolfgang Richter\textsuperscript{1}, Glenn Ammons\textsuperscript{3}, Jan Harkes\textsuperscript{1}, Adam Goode\textsuperscript{4}, Nilton Bila\textsuperscript{2}, Eyal de Lara\textsuperscript{2}, Vasantha Bala\textsuperscript{3}, Mahadev Satyanarayanan\textsuperscript{1}

\textsuperscript{1} Carnegie Mellon University
\textsuperscript{2} University of Toronto
\textsuperscript{3} IBM Research
\textsuperscript{4} Google
Introspection vs Retrospection

- Examine **active state** of VM during execution
- Examine **historical state** of VMs and their snapshots

VM Instance A

Examine live logs

Examine all historic logs A*
Change: Shift in Thinking

- Traditionally a VM == executable content
- VM Image Libraries break this paradigm
- Think of VMs as big data
- What can we do with them?
Change: Shift in Thinking

- Traditionally a VM == executable content
- VM Image Libraries break this paradigm

Apple's Time Machine over all VM instances including their complete snapshotted history
Retrospection

• Deep search over historical VM data
  • Snapshots, Virtual Disks, ...

• Help with:
  • Debugging and troubleshooting
  • Legal establishment of data/code provenance
  • Malware tracking
  • License violations
Deep Search?

- Search content of files
  - Pictures, Documents, Binary files
- Enable proprietary plugins
  - Adobe, MS Office, Norton, SW Discovery Tools
- While respecting privacy...
Example: Forensics - Before

1. Crawl live or frozen logs
2. Pull backups if available
3. Examine differences
4. Determine root causes
5. Redeploy
Example: Forensics - After

1. Crawl live or frozen logs
2. Retrospect entire history
3. Examine differences
4. Determine root causes
5. Redeploy
Unified interface for searching historic state:
uncover suspicious log entries, infected binaries, etc. at once

1. Crawl live or frozen logs
2. **Retrospect entire history**
3. Examine differences
4. Determine root causes
5. Redeploy
Example: Copyright

• Examine a set of instances
• Retrospect to find history of transforms
• Provide evidence in court
• Multiple companies with similar cloud infrastructures supporting retrospection could perform the same queries
Privacy via Cryptography

- Complete trust, if encrypted keys shared
- Some trust, key escrow service
- No trust, no external search infrastructure
- Per-file, per-directory, per-partition
Design Principles

1) Support on-demand queries, scoped to a minimal set of data.

2) Control of retrospection policy resides with VM owners, not cloud operators.

3) Place as few constraints as possible on the generality of search computations.
Retrospection

- VMs become **big data**
- **New opportunities** with deep search over historical VM data
- **Retrospection** is the unifying mechanism for examining historical VM data
- **Nanuk** – Our implementation
Questions?
IBM Research Mirage

- **Virtual Image Library**

- **File-level** deduplication
  - Files are referenced by SHA-1

- Reads VM Image partitions and file systems
OpenDiamond Platform

• Distributed, interactive, unindexed search

• Focuses on the principle of early discard

• Enables arbitrary search queries
  • Arbitrary x86 binary code as query primitives
Achievable Efficient Retrospection
Effect of Deduplication - Bytes

Data from 78 NCSU VCL VM Images based on Windows XP
Effect of Deduplication - Files

Data from 78 NCSU VCL VM Images based on Windows XP