Eternal Sunshine of the Spotless Machine: Protecting Privacy with Ephemeral Channels

Alan M. Dunn, Michael Z. Lee, Suman Jana, Sangman Kim, Mark Silberstein, Yuanzhong Xu, Vitaly Shmatikov, Emmett Witchel

University of Texas at Austin

OSDI 2012

October 8, 2012
Wanted: Application Privacy

• Goal: Run programs without leaving traces

• Current state: Private browsing
  – Popular feature in web browsers
  – Ideal: When private browsing session terminates, all traces erased
A Privacy Problem

• Private browsing **unachieved**
  – Evidence of site visits leaks into OS [Aggrawal, 2010]

• Problem: **No system support**
  – Applications interact with user and world
  – Data leaks into OS, system services
  – Applications **cannot** remove traces they leave
Example: Browsing a Website

What traces still remain on the computer?
Leaks From Browsing

Memory contents:
Complete packets, like:

HTTP/1.1 200 OK
Date: Mon, 17 Sep 2012 ...
Server: Apache/2.2.14 ...

X server caches, graphics drivers

Network

Audio

PulseAudio server
Secure Deallocation Is Not Enough

• **Secure deallocation**: Zero memory when freed
  – Research implementation [Chow, 2005]
  – PaX: Security patch for Linux kernel

• Sensitive data remains allocated
  – X caches, PulseAudio buffers not freed
Resisting a Strong Adversary

• Goal: Provide forensic deniability – no evidence left for non-concurrent attacker
• Once program terminated, protection maintained under extreme circumstances

Root-level compromise (after program terminates)  Computer physically seized
Goals

• Provide privacy
  – Private sessions with forensic deniability

• Maintain usability
  – Simultaneous private/non-private applications
  – Support a wide variety of private applications
  – “Pay as you go” - costs only for private programs
  – Impose low overhead
Lacuna

- System to accomplish our privacy and usability goals
- Host OS (Linux), VMM (QEMU-KVM) modified
- Applications unmodified

**Lacuna** [luh-kyoo-nuh]
1. a gap or missing part, as in a manuscript, series, or logical argument...
Outline

• Design
  – Erasable program container
  – Allow communication with peripherals

• Evaluation
  – Lacuna provides privacy
  – Lacuna maintains usability
Erasable Program Container

VM contains Inter-Process Communication

VM alone is insufficient
Communicating with Peripherals

- Sensitive data

Program must communicate with peripheral

Dependencies on rest of OS
Communicating with Peripherals

- Sensitive data

Code with potential data exposure

Host OS
Two Peripheral Types

1) Storage
- Encrypted data
  - Encrypt before data passes through OS

2) All other peripherals
- Sensitive data
  - Must ensure no traces left that are readable later

Solve with ephemeral channels
Ensuring No Readable Traces

Strategy 1: Leave no trace

Strategy 2: Make traces unreadable later
Ephemeral Channels

- Sensitive data
- Encrypted data

Host OS

Erase channel key
(complex OS paths)

Proxy

Encrypted ephemeral channel

Hardware ephemeral channel

Guest control of hardware
# Channel Type Comparison

<table>
<thead>
<tr>
<th></th>
<th>Hardware</th>
<th>Encrypted</th>
</tr>
</thead>
<tbody>
<tr>
<td>Host drivers unmodified</td>
<td>✓</td>
<td>✗</td>
</tr>
<tr>
<td>Host code never sees unencrypted data</td>
<td>✓</td>
<td>✗</td>
</tr>
<tr>
<td>Hardware virtualization support unnecessary</td>
<td>✗</td>
<td>✓ (No graphics)</td>
</tr>
<tr>
<td>Guest modification unnecessary</td>
<td>✗</td>
<td>✓ (Run Windows, Linux, unmodified programs)</td>
</tr>
</tbody>
</table>
Encrypted Graphics Channel

- No hardware virtualization support for graphics
- Solution: Encrypt VM output to GPU memory

![Diagram of GPU memory, CUDA, Host OS, Driver, VM, and Emulated graphics card connections]
Hardware USB Channel

Controller under guest control

Controller: non-private

Controller: private

VM

Guest’s USB host controller driver

VMM

Host OS

Driver

Encrypted USB, audio, network channels described in paper

USB host controller HW

USB mouse

USB keyboard

Switch into private mode

USB host controller HW

Host OS
Sanitizing Storage

• Encrypt VM writes to storage
  – VM image file unmodified
  – **Diffs file** contains VM writes to storage
  – Diffs file encrypted

• Leave no evidence of which storage locations read
  – Free buffer cache pages for VM image file only

• Encrypt swapped memory from private VM
  – Encrypt swapped pages for VMM process only

• **Encryption keys erased on VM exit**

• Techniques here “pay as you go”
Evaluation

• Lacuna provides privacy
  – Measure that Lacuna does not leak private data
  – Quantify size of code that handles sensitive data

• Lacuna maintains usability
  – Low switch time to private environment
  – Application performance near that of running program in VM

• More evaluation in paper
Lacuna Protects Privacy

• Experiment to locate leaks
• Inject random “tokens” into peripheral I/O paths, scan memory to locate [Chow, 2005]
• Tokens almost always found without Lacuna
• Tokens never found with Lacuna
Little Code Handles Sensitive Data

<table>
<thead>
<tr>
<th>Subsystem</th>
<th>Lines of Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>Graphics</td>
<td>725 (CUDA)</td>
</tr>
<tr>
<td>Sound</td>
<td>200 (out) 108 (in)</td>
</tr>
<tr>
<td>USB</td>
<td>414</td>
</tr>
<tr>
<td>Network</td>
<td>208</td>
</tr>
</tbody>
</table>

- Measurements are lines of code outside of QEMU that handle unencrypted data
  - Data within QEMU erased at VM exit
Time to Switch to Private Programs is Low

<table>
<thead>
<tr>
<th>Channel Type</th>
<th>Switch Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>USB passthrough (encrypted)</td>
<td>1.4 ± 0.2</td>
</tr>
<tr>
<td>keyboard</td>
<td>2.3 ± 0.2</td>
</tr>
<tr>
<td>keyboard + mouse</td>
<td>2.4 ± 0.2</td>
</tr>
<tr>
<td>PCI assignment (hardware)</td>
<td>3.8 ± 0.2</td>
</tr>
</tbody>
</table>

- USB driver disconnect significant (0.8-1.0 s)
- Switch time achieved by eliminating two extra disconnects in guest USB initialization
Impact on Full-System Workloads is Low

• Benchmarks
  – MPlayer: Watch video in across network
  – Firefox: Browse Alexa top 20 websites
  – LibreOffice: Create 2,994-character, 32-image document

• No execution slowdown, higher CPU utilization

<table>
<thead>
<tr>
<th></th>
<th>Video (75 s)</th>
<th>Browser (20 s)</th>
<th>Office Suite (175 s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>QEMU</td>
<td>32.2 ± 7.4</td>
<td>25.9 ± 1.3</td>
<td>8.1 ± 1.2</td>
</tr>
<tr>
<td>Lacuna</td>
<td>49.7 ± 0.3</td>
<td>46.2 ± 1.5</td>
<td>21.1 ± 0.6</td>
</tr>
<tr>
<td></td>
<td>(+ 17.5)</td>
<td>(+ 20.3)</td>
<td>(+ 13.0)</td>
</tr>
</tbody>
</table>

Measurements are % CPU utilization.

• CPU utilization lowered by hardware AES (AES-NI)

Worst case: additional 20 percentage points
Conclusion

• Modern computer systems leak secrets
• Lacuna provides **forensic deniability**: secrets removed after program termination
• **Ephemeral channels** provide private peripheral I/O
• Lacuna runs full-system workloads efficiently