Trustworthy Whole-System Provenance for the Linux Kernel

Adam Bates, Dave (Jing) Tian, Thomas Moyer, and Kevin R. B. Butler

In association with…

College of Engineering
UNIVERSITY of FLORIDA

LINCOLN LABORATORY
Massachusetts Institute of Technology

USENIX Security Symposium, Washington D.C., USA
13 August, 2015
Provenance Matters!

Do E-Mails Reveal Scientist Claims On Climate Change are...

BUNK?

Hackers break into servers of a major British climate change research facility and purportedly uncover e-mails urging scientists to 'hide the decline' of temperatures, manipulate data and silence skeptics.
How a Burnt Lady Gaga CD Helped Leak Thousands of Intelligence Files

Kat Hannaford
Filed to: WIKILEAKS   11/29/10 7:20am

Do E-Mails Reveal Scientist Claims On Climate Change are...
BUNK?

Hackers break into servers of a major British climate change research facility and purportedly uncover e-mails urging scientists to ‘hide the decline’ of temperatures, manipulate data and silence skeptics.
“40,000 Massachusetts defendants may be affected by chemists’s alleged misdeeds”
- Morgan Windsor, CNN, Aug 2013
Provenance Matters!

GIZMODO

How a Burnt Lady Gaga CD Helped Leak Thousands of Intelligence Files

But is our provenance secure?

defendants may be affected by chemists’s alleged misdeeds”

- Morgan Windsor, CNN, Aug 2013
Overview

- **Linux Provenance Modules (LPM)**, for trustworthy provenance monitors in Linux.

- **Provenance-Based Data Loss Prevention**, to monitor and control the propagation of sensitive data in enterprise environments.

**Evaluation:**

- Collection agent imposes 3%-8% runtime overhead
- Provenance queries return in under 3 milliseconds
Def: provenance \( \text{prä-və-nän(t)s} \ n: \)

- A complete description of system Agents…
  - e.g., Users, Groups

- … controlling Activities…
  - e.g., Processes, Forks

- … and their interactions with Controlled Data Types.
  - e.g., Inodes, Sockets, IPC, Memory
Threat Model

• **Provenance-Aware Adversary** attempts to disable collection agent, tamper with logs, etc.

• **Provenance-Aware Applications** can be compromised, and may lie about system events.

• Kernel is trusted on install, but can later be attacked.

• PKI stores and distributes keys for Prov-Aware Hosts.
Design Goals

1. Completeness
   • Gapless descriptions of system activity

2. Tamperproof
   • Impervious to attacks launched in user space

3. Verifiable
   • Formal assurance of G1, G2

4. Authenticated Channel
   • Tamper-evident provenance transmission

5. Secure Disclosure
   • Validate annotations disclosed in user space
Kernel layer collection agent:

- LPM architecture mirrors Linux Security Modules.
- Kernel instrumented with 170 provenance hooks.
- Modules efficiently transmit provenance to user space with relay buffer.
Kernel layer collection agent:

Text Editor

open System Call

Look Up Inode

Error Checks

DAC Checks

LSM Hook

LPM Hook

Access Inode

"Authorized?" Yes or No

LSM Module
Examines context.
Does request pass policy?
Grant or deny.

"Prov collected?" Yes or No

LPM Module
Examines context.
Collects provenance.
If successful, grant.

Example control flow through an LPM provenance hook.
Support for provenance storage:

- Recorders translate provenance stream for various storage backends.
- Support recording to file, relational DBs, graph DBs.
- Upcoming: Accumulo.
Support for networked provenance-aware systems:

- Message Commitment Protocol enforced with Netfilter subsystem.
- LPM performs per-packet DSA signing and verification.
- Signatures are embedded in IP Options, ensuring (nearly) universal compatibility.
Support for networked provenance-aware systems:

Example control flow for authenticated packet transmission.
Support for layered provenance-aware systems:

- Kernel provenance suffers from *semantic gap* problem.
- Layered provenance bridges the gap, but expands attack surface.
- Authenticity and integrity of workflow provenance must be validated, but how?
Support for layered provenance-aware systems:

- LPM includes a gateway for upgrading low integrity provenance.
- Integrity Measurement Architecture (IMA) check verifies load time integrity of application.
- Only correctly validated provenance is recorded.
1. Completeness
   • Provenance hooks observe all sensitive operations performed on controlled data types.

2. Tamperproof
   • SELinux preserves run-time kernel integrity
   • Secure Boot techniques prevent booting into another kernel

3. Verifiable
   • By mirroring LSM hooks, LPM inherits formal analysis that ensures complete mediation of controlled data types.

4. Authenticated Channel
   • Message Commitment Protocol ensures integrity and identity.

5. Secure Disclosure
   • IMA check verifies load time integrity of Prov-Aware Applications.
Data Loss Prevention tools take the following forms:

- **Regex-Based**: Fails to recognize data transformations
- **Manual Labelling**: Not tamper proof, may fail to handle data fusions.
- **Provenance-Based**: All lineage information is recorded, any sensitive ancestry can be traced.
## Evaluation: Collection Costs

<table>
<thead>
<tr>
<th>Benchmark</th>
<th>Vanilla Kernel</th>
<th>LPM w/ Provmon</th>
<th>Overhead</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Kernel Compilation</strong></td>
<td>598 sec</td>
<td>612 sec</td>
<td>2.7%</td>
</tr>
<tr>
<td><strong>Postmark</strong></td>
<td>25 sec</td>
<td>27 sec</td>
<td>7.5%</td>
</tr>
<tr>
<td><strong>Blast Sequencing</strong></td>
<td>376 sec</td>
<td>390 sec</td>
<td>4.8%</td>
</tr>
</tbody>
</table>

Overhead is highest on I/O intensive tasks with frequent file creation, deletion, and open.

Costs amortize over reads and writes.
Storage overhead is high, but consistent with other system layer provenance/audit tools. Compression techniques can be used to reduce storage burden.
Evaluation: Query Costs

99% of queries return in 2.5 ms or less!

Worst case: 17,696 nodes Returns in 21 ms

PB-DLP Ancestry Queries for Inodes in a 6 million node graph.
(Only inodes with over 50 ancestors were considered)
Evaluation: Network Prov.

90% throughput reduction with message commitment protocol...

IPerf TCP benchmarks of Message Commitment Protocol.

Alternatives: SSL or IPSec, which require app rewriting.
In this work, we...

- identify the requirements for trustworthy provenance in distributed, heterogeneous environments.

- design, implement, and deploy the first fully-realized provenance monitor.

- propose a mechanism for provenance-based data loss prevention that offers improved capabilities over existing enterprise systems.
Questions?

Thank you for your time.

adambates@ufl.edu

LPM is available at http://linuxprovenance.org