Trusted Computing and Provenance: Better Together

John Lyle and Andrew Martin
Oxford University Computing Laboratory
Some Background

- My research is in security, not provenance (sorry!)
- We're interested in **assurance** of platform **behaviour** through reporting **system state**
- Part of provenance is knowing **system state** to support and guarantee consistent **behaviour**
- Growing interest in **secure** provenance

- Surely there's some overlap...
Why Secure Provenance?

- Provenance can provide assurance in the quality of scientific results
  - Many new threats: not just unintentional error
  - High-profile science has a greater risk of malicious intervention. E.g. Climate change
- Provenance is a great defence against:
  - Attacks on reputation (e.g. Climategate)
  - Attempts to influence results
- But only if provenance records are tamper-proof
- Even more important with large, distributed systems
Trusted Computing

- Trusted Computing can provide tamper-proof guarantees of program execution.
- It can provide information about hardware and software.
- To explain how, need to go into some technical details.
- Stop and ask me questions!
  - Time constraints mean I'm glossing over lots of details.
Integrity Reporting

Assessing trustworthiness by asking:

“What programs are you running?”
Root of Trust

Bootloader

Operating System

App

App

App

App

Root of Trust

Loads

Bios

Bootloader

Operating System

App

App

App

App

Measure the binary hash

Trusted Platform Module (TPM)

“Authenticated Boot”
PCR   Hash value   Executable
10    61a3393ccbac66e6f1680b105e97377797c70   boot_aggregate
10    42d5319f874a26c639c2a04f838f177000a60    fan       [kernel module]
10    d9f54f7f0a296ae15a542e0a411f18b8cd9c9c    processor [kernel module]
10    9d4f0f315936756c83cf47b4c41e3cf26df408a   thermal    [kernel module]
...
10    df20bd67ea041bcb2f535b823a876e6540efaf12   /bin/sh
10    6821426a0ede03a51db095b2153db59d1c2bc9    /bin/mount
10    007857f1f7913883d2d6c69455b325cb05c0b152d   /bin/bash
10    932cb250bca27d29a6cb0cfe7e422cefe58f56f93    /bin/mknod
10    851490d5ab05c3457ceebab87d1f59aa8a76fc66    /bin/ln
10    89d06c92e7101e4e491c72f30ba4c4950b286d    /bin/mkdir
10    1f5b13c4667b9347e7026cf71233bc7ab493         /bin/grep
10    db16a079245e5d37539daed12734af5c209fc5290   /bin/cp
...

PCR10 = SHA1( A_n, SHA1( ... SHA1( A_1 , SHA1(A_0, 0x00 ))) )
Remote Attestation

Sign a copy of your boot measurements
What about Provenance?

- Security and provenance rely on establishing a complete picture of the factors influencing a remote computer's behaviour.
- Trusted Computing can do it in a tamper-resistant manner.
- Attestations can be considered trustworthy actor-state p-assertions.
- This is immediately applicable to large-scale grid computing.
- We have the technology already!
Attestation-based Provenance

Job Report: (Attestation)

PCR Boot Hash, Boot log, Job Request, Job Result

Remote Service

Submit job

User

Result

Provenance Store

Send report

Other data sources...

Send report

Send report

Send report

Send report
Information Collected

- Platform unique identity (AIK)
- All software identities
  - Firmware, drivers, operating systems, applications
- Hardware identities*
- Timestamps
- Job information**
  - A hash of the job / request message
  - A hash of the calculated result
Optimising Storage

Reference Manifest Database
(Application → hash list)

Provenance Store

Service Provider

Job Report:
(Job ID,
Boot Hash,
Boot Log,
Request,
Result,
Signature)

Job ID → (Request, Result,
Boot Hash, Signature)

Boot Hash → [RIM1, RIM2, ...]

RIM → (Application, Date, Version, ...)

Service Provider

Job Report:
(Job ID,
Boot Hash,
Boot Log,
Request,
Result,
Signature)
We Have The Technology...

- Software for Java, C++, .net
- TPMs are cheap and available
- Linux has native support for Authenticated Boot and TPMs. Windows too*
- Just needs to be integrated into middleware
- Virtualisation makes much of this easier
  - Report on a virtual machine image
What we can't do (yet)

• Runtime information and configuration details
  – Can be added, but needs some work
  – This is some of my future work

• Needs integrating with other provenance information
  – Purpose of experiment, sources of data, etc...

• Recreating results is not an automatic process
  – Virtual machines may also help here

• Need to have a frequently-updated software database (RMDB)
Part 2: Better together?
Research in Common

- Trusted Computing and provenance have a lot of common research.
- Secure, transparent logging
- Usage control / monitoring
- Compilation histories
- Secure storage
- Even using the same examples
  - Grid, SOA, cloud
- Desire to automate and scale
- Integrity!
A Problem Shared

- Provenance can become more trustworthy if it takes advantage of (and influences) security architectures
  - A fantastic case study for Trusted Computing too.
- Security is about eliminating the hidden factors, the unexpected attacks and variables
  - Good science does the same
- We don't know how to process and filter data. Is provenance further ahead?
  - What do we do with incomplete information?
  - Metadata, semantics, composition of data
... is a problem doubled?

• Different research directions.
  – Cryptographic strength vs data consistency and accuracy
• Lots of new and interesting security challenges, maybe Trusted Computing won’t help with the big ones?
• How do we develop secure software?
  – If grid middleware is vulnerable to runtime attack, have we gained anything?
• Other issues: PKI, performance, usability, privacy ...
• My literature review just became twice as long!
Conclusion

• Two fields that are solving similar problems
  – We both want tamper-proof identification of systems
• There is a lot of immediately applicable software and hardware
  – exciting opportunity for researchers and developers of provenance tools.
• If we work together, trusted provenance shouldn't be that far away.