Attaining Least Privilege Through Automatic Partitioning of Hybrid Programs

William Enck
Assistant Professor
Department of Computer Science
[in collaboration with Xiaohui (Helen) Gu]
The Protection of Information in Computer Systems

JEROME H. SALTZER, SENIOR MEMBER, IEEE, AND
MICHAEL D. SCHROEDER, MEMBER, IEEE

Invited Paper

Abstract - This tutorial paper explores the mechanics of protecting computer-stored information from unauthorized use or modification. It concentrates on those architectural structures—whether hardware or software—that are necessary to support information protection. The paper develops in three main sections. Section I describes desired functions, design principles, and examples of elementary protection and authentication mechanisms. Any reader familiar with computers should find the first section to be reasonably accessible. Section II requires some familiarity with descriptor-based computer architecture. It examines in depth the principles of modern protection architectures and the relation between capability systems and access control list systems, and ends with a brief analysis of protected subsystems and protected objects. The reader who is dismayed by either the prerequisites or the level of detail in the second section may wish to skip to

Confinement
Allowing a borrowed program to have access to data, while ensuring that the program cannot release the information.

Descriptor
A protected value which is (or leads to) the physical address of some protected object.

Discretionary
(In contrast with nondiscretionary.) Controls on access to an object that may be changed by the creator of the object.

Domain
The set of objects that currently may be directly accessed by a principal.

Encipherment
The (usually) reversible scrambling of data according to a secret transformation key, so as to make it safe for
Least Privilege

• “Every program and every user of the system should operate using the least set of privileges necessary to complete the job”
  – Difficult to achieve in practice
Privilege Separation

• SSH privilege separation (Provos)
• Privtrans
• … others
Idea!

• Sometimes modular software designs create *natural partitions* in the software.

(Image: http://moreindia.in/blog/2011/01/spiti-valley/spiti-valley-himachal-pradesh-india/)
... but are they useful?

- **Research Question:** What security policy should be assigned to partitions and how?
Case Studies

• Java Native Interface (JNI) in Android
  – Root exploits use 3rd-party native libs

• Android’s component abstraction
  – Sub-process thread with a purpose
Android 3rd-Party Native Libs

• Insight: maybe native code “behaves” differently

• Least Privilege Policy:
  – Selective monitoring of system calls
Real-World Monitoring

SBL Training

1. Developer
   - Input Trace (optional)
   - app

2. Market Malware Detection
   - app, trace
   - SBL Module
     - Selective syscall recording
     - Normal profile creation

SBL Enforcement

3. Normal profile
4. SBL Service
   - \{app1, app2,...\}
   - Normal profiles
   - SBL Module
     - Selective monitoring
     - Online root exploit detection
     - Suspicious syscall sandboxing

App Market
   - app

Science of Security
Lablet

Computer Science
NC State University
“Partitioning”

• Initial work focused on learning model

• Modify JNI call bridge to execute 3rd-party native libraries in a separate thread
Results
Android Components

- Android forces developers to organize their applications into a component abstraction

- Starting an Activity for a Result
- Communicating with a Service
- Querying a Content Provider
- Receiving an Intent Broadcast
App Mashups

- The user experience for a task often spans multiple applications.
“Partitioning”

- Separate UI from background functionality
- Expand the protection domain across apps
Aquifer

- Aquifer defines protection policy with respect to the *User Interface Workflow*
Policy

• Aquifer policy is a form of Decentralized Information Flow Control (DIFC)
  – Data owners (apps) define policy

**Definition 4 (Workflow label).** A workflow label $L$ is an expression $L = \{O_1 : (E_1, R_1, F_1); \ldots; O_n : (E_n, R_n, F_n)\}$, where $O_i$ is an owner (application) and $E_i$, $R_i$, and $F_i$ are an export list, required list, and workflow filter, respectively, specified by $O_i$. 
Prototype

• Prototype implementation on Android
• Case Studies
  – Enhanced K-9 Mail
  – Document choosers
  – Compatibility
Towards Least Privilege Principles

• So far, the results are promising
  – Selective monitoring on JNI
  – UI vs background components

• Beginning to expand to new case studies
Collaboration

• Expand on existing projects, or …
• Where do you see natural partitions in the systems/software you work with?
  – Cloud (e.g., MapReduce), Web
Collaboration

• Expand on existing projects, or …
• Where do you see natural partitions in the systems/software you work with?
  – Cloud (e.g., MapReduce), Web
• Contact:
  – William Enck (enck@cs.ncsu.edu)
  – Xiaohui (Helen) Gu (gu@cs.ncsu.edu)