Encapsulating Mobile Objects

Hermann Härtig Lars Reuther

Dresden University of Technology
Department of Computer Science
{haertig,reuther}@os.inf.tu-dresden.de

The Problem

Mobile Code (Applets, Servlets, Agents …)
can be useful,

Iff it has access to resources it needs,
but *only* to these

Current Approaches

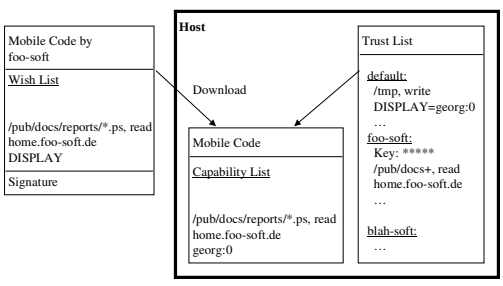
- Mandatory access control
- Access control lists
- Sandboxing
- Firewalls
- Capability lists
- Active interposition
  (e.g. Clans & Chiefs, ACL-programs, Custodians, …)

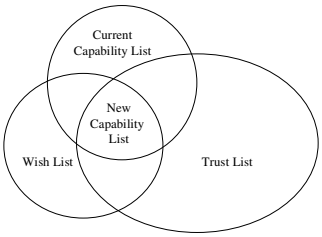
Requirements

- Mobile Code must
  - Be effectively isolated within systems
  - Be given access to resources it needs
- A proposed solution must
  - Scale
  - Work in heterogeneous environments
  - Be simple to administrate

Our Scheme: The Principles

- System owner: expresses (dis-)trust in vendors / programs
  ⇒ *Trust List*
- Mobile Code: denotes resources it needs
  ⇒ *Wish List*
- On entry: denoted resources are checked
  ⇒ *Capability List*
- During execution: Capabilities are enforced

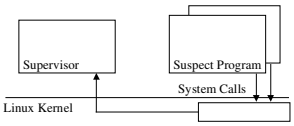




Integration into a System Architecture

- Interpreter / Compiler
  - \+ portability
  - – limits the range of applications (e.g. only Java VM)
  - – reimplementaion for each language system
- Operating System
  - \+ no limits for applications
  - – reimplementation for each OS

General Structure



The Linux Kernel
- intercepts system calls
- enforces capabilities

The Supervisor
- verifies signatures
- handles Trust and Wish Lists
- generates Capability Lists

Exec System Call

- is intercepted
- the signature of the new program is checked
- new Cap. List := old Cap. List ∩ Trust List ∩ Wish List
- inspect file descriptors

Status

- prevent attacks on confidentially and integrity
- no symbolic links
- Linux on L4:
  uncontrolled access to other servers than Linux
- overhead:

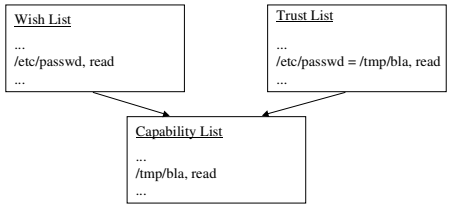
	with *-Lists	without *-Lists
open	18500	10250
connect	45500	42250

Related Work

- I. Goldberg et al., UCB
  Janus:
  System call tracing uses Solaris `/proc`-System
- N. Islam et al. , IBM Watson
  FlexxGuard:
  Capability lists integrated into Java VM

Interesting Problems

- *setuid*-Programs



⇒ **exec su**

Summary

- Effective isolation of Mobile Code
- Mobile Code denotes the resources it needs
- Administrators express trust in vendors / programs
- Implementation can be done by integration into the OS