Practical Protection for Personal Storage in the Cloud

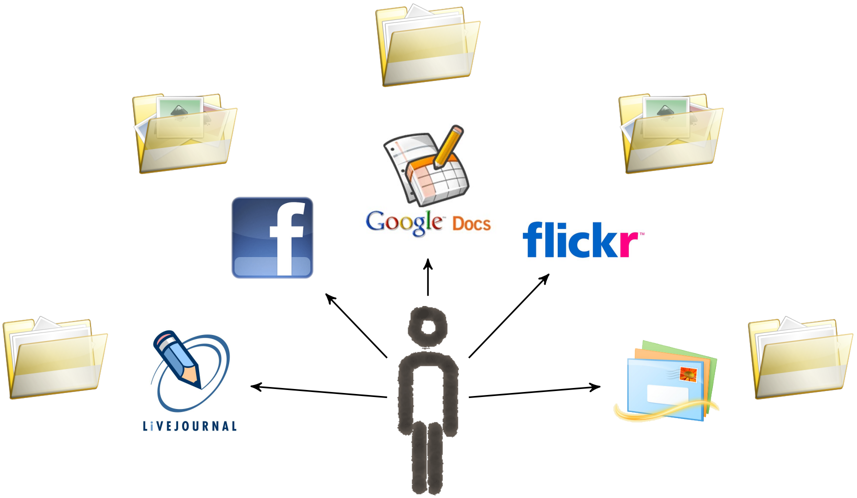
Neal H. Walfield, Paul T. Stanton,
John Linwood Griffin and Randal Burns
Johns Hopkins University

EuroSec '10
April 13th, 2010

Outline

- Personal Storage Today
- Practial Protection Mechanisms

Web 2.0: Today



- Each service provides the user with storage
- Limited support for sharing between services

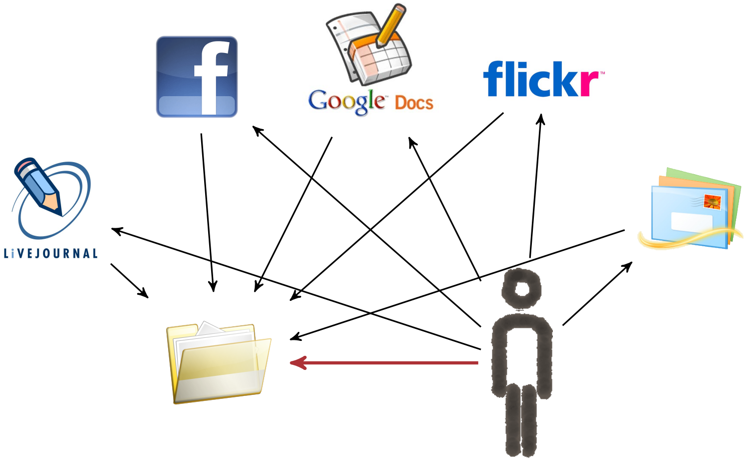
An Emerging Issue

- **Data Management is Hard!**
  - Data Lock-In
    - No standardized access interface (à la POSIX)
    - Must use service's interface; point solutions
  - Data Spew
    - Data is hard to find
  - Version Drift
    - Sharing across services $\Longrightarrow$ divergent copies

An Emerging Issue

- **Data Management is Hard!**
  - Data Lock-In
    - No standardized access interface (à la POSIX)
    - Must use service's interface; point solutions
  - Data Spew
    - Data is hard to find
  - Version Drift
    - Sharing across services $\Longrightarrow$ divergent copies
- Underlying Architectural Problem:
  - **Many storage providers**
  - $\Longrightarrow$ No unified view of data

A Simple Solution: One Storage Provider



- User has direct access to data
- Single, authoritative copy
- Cross-service sharing

A Simple Difficulty

- **Access Control**
  - Facebook should not be able to access EMail

A Simple Difficulty

- **Access Control**
  - Facebook should not be able to access EMail
- **Reputation!**

A Simple Difficulty

- **Access Control**
  - Facebook should not be able to access EMail
- **Reputation is not enough!**
  - Users less likely to experiment
  - Raises barrier to entry

Outline

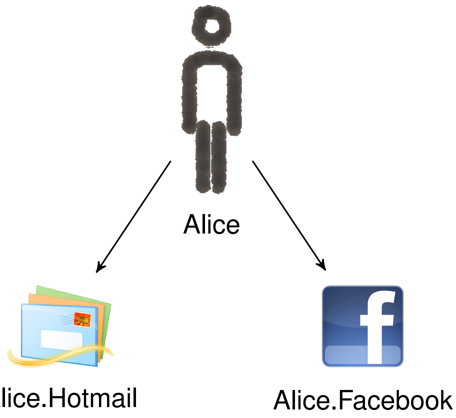
Per-User Storage: Major Design Goals

- Protection
    - Least Privilege
        - Not Unix
        - Fine-grained, dynamic delegation and revocation
- Usability
    - Minimal user interactions with security manager
        - Opening, saving files
    - Delegate access to not-yet-existing objects
        - Flickr can access all JPEG files
    - Consistent naming of objects
        - `/photos/paris/dsc_1076.jpg` always has same name

S4: Simple, *Secure* Storage Service

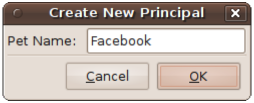
- Hierarchical Principals
- Filtered Views
- Powerbox
  - Security manager implements open, save-as dialogs

Principals

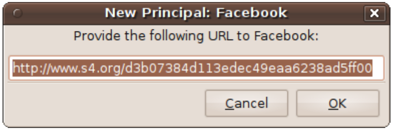


- Hierarchical
    - Alice dominates Alice.Hotmail
- Principals identified using public key cryptography

Creating a new Principal


Create New Principal

Pet Name: Facebook

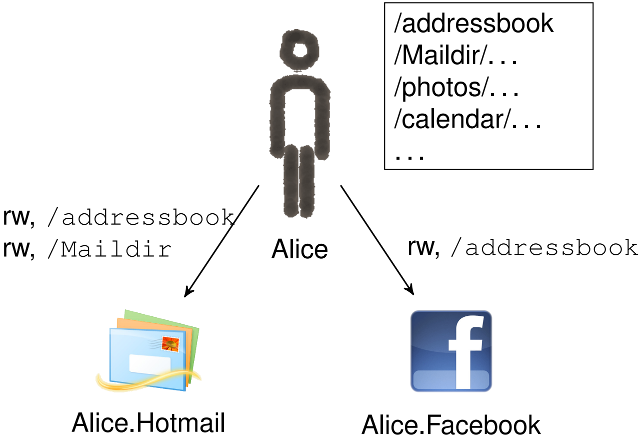

New Principal: Facebook

Provide the following URL to Facebook:

http://www.s4.org/d3b07384d113edec49eaa6238ad5ff00

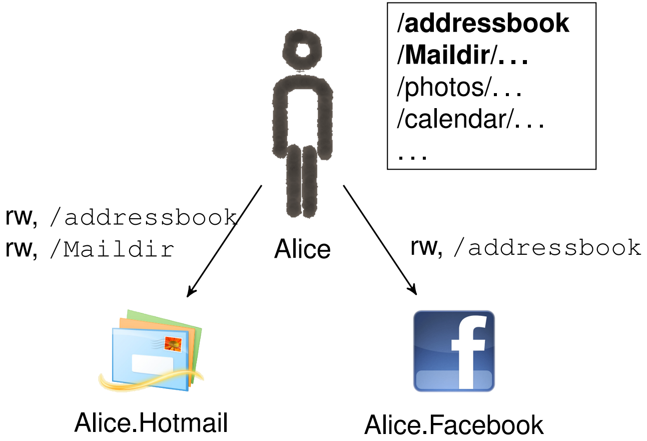
- Credentials communicated using a Webkey
  - Includes service's public, private keys
  - Includes storage server's public key

Filtered Views



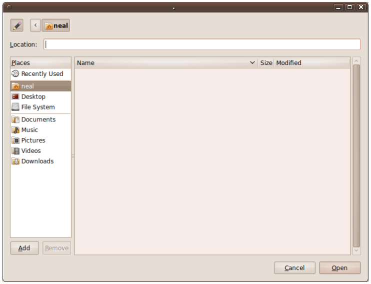
- Filter parent's name space
  - Principal can access that which it can name
- e.g., Regular expressions
- Enables consitent naming, future delegations

Filtered Views

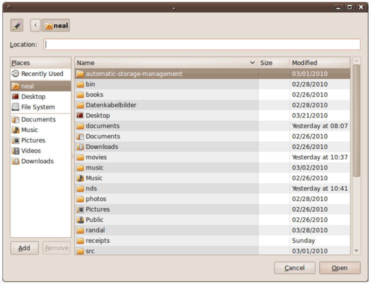


- Filter parent's name space
  - Principal can access that which it can name
- e.g., Regular expressions
- Enables consitent naming, future delegations

Powerbox



Least Privilege View



Name	Size	Modified
automatic-storage-management		03/01/2010
bin		02/28/2010
books		02/26/2010
Datenkabelbilder		02/28/2010
Desktop		03/21/2010
documents		Yesterday at 08:07
Documents		02/26/2010
Downloads		02/26/2010
movies		Yesterday at 10:37
music		03/02/2010
Music		02/26/2010
nds		Yesterday at 10:41
photos		02/28/2010
Pictures		02/26/2010
Public		02/26/2010
randal		03/28/2010
receipts		Sunday
src		03/01/2010

Powerbox View

Powerbox

- Concept
  - Replaces application’s open, save-as dialog box
  - Service sends an RPC to security manager
  - Security manager displays dialog box
- Essential for usable least privilege
  - Dynamic delegation
  - No (explicit) user interactions with security manager

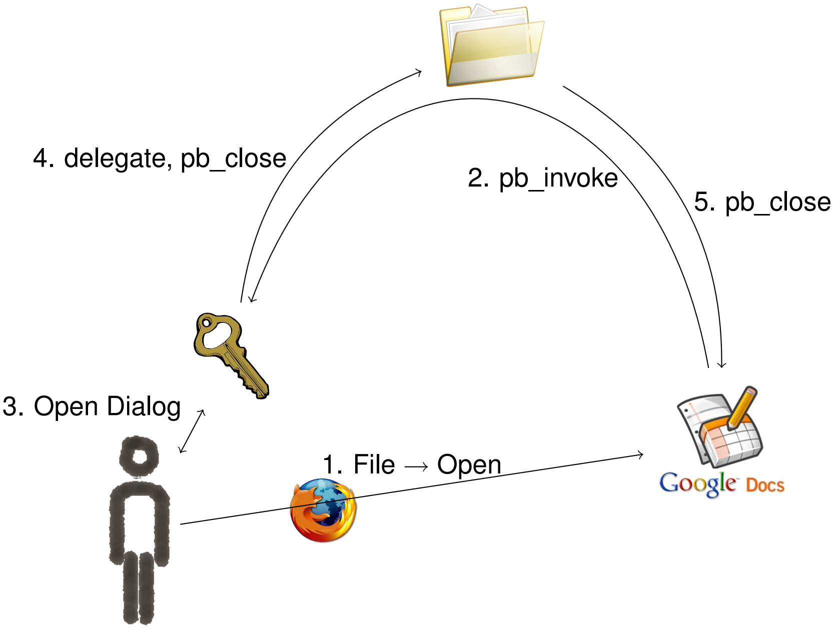
Integrating the Powerbox into Flickr

- Alice creates a Flickr account at flickr.com
- Alice creates a principal using security manager
- Alice gives credentials to Flickr
- Flickr starts an import photos wizard
  - Invokes Powerbox
  - What files would you like to import to Flickr?
  - Alice selects one or more directories

Integrating the Powerbox into Flickr

- Alice creates a Flickr account at flickr.com
- **Alice creates a principal using security manager**
- Alice gives credentials to Flickr
- Flickr starts an import photos wizard
  - Invokes Powerbox
  - What files would you like to import to Flickr?
  - Alice selects one or more directories
- Differences:
  - One additional step
  - But, Alice can use her own tools to upload photos

Powerbox Protocol in S4



Performance

- User's storage is *authoritative*
- Services can (should) still cache
  - Prompt propagation of updates

Adoption

- User’s want it
  - Improved usability, control
  - $\Longrightarrow$ Current services lost control
    - Differentiator for new service providers

Adoption

- User's want it
  - Improved usability, control
  - $\Longrightarrow$ Current services lost control
    - Differentiator for new service providers
- Big services providers want it?
  - Increase user traffic by becoming a storage provider

Implementation

- 4000 lines of Python (SLOCCount)
  - Single machine, Single threaded
- S3 compatible
- S3 and SQLite backends
- Principal and filter interfaces complete, some Powerbox

Future Work

- Filters based on files' tags
- Snapshots for recovery
- COW for experimentation
- Publish/subscribe for updates
- Throttling bandwidth intensive services
- Do not disclose content to server

Summary

The Bad (the status quo)

- Data lock-in
- Data spew
- Version drift

The Good (what S4 tries to achieve)

- Single (perceived) file system
- Least privilege
- Minimal user interaction with security monitor
  - Powerbox
  - Protection mechanisms consistent with user's intuitions
    - All `JPEG` files
- Delegate access to not-yet-existing objects
- Consistent naming of objects

Take Aways

- Filtering matches how users think about security policies
- Powerbox helps make security invisible

Image Attributions

- User Images - User Experience Deliverables by Peter Morville and Jeffery Callender - `http://www.flickr.com/photos/morville/3220961846/` - CC Attribution 2.0
- File Images - `http://www.openclipart.org/user-cliparts/sarxos` - Public Domain
- Key Image - `http://www.openclipart.org/people/johnny_automatic/` - Public Domain

Summary

The Bad (the status quo)

- Data lock-in
- Data spew
- Version drift

The Good (what S4 tries to achieve)

- Single (perceived) file system
- Least privilege
- Minimal user interaction with security monitor
  - Powerbox
  - Protection mechanisms consistent with user's intuitions
    - All `JPEG` files
- Delegate access to not-yet-existing objects
- Consistent naming of objects