Advanced Systems Security: Future

Trent Jaeger
Systems and Internet Infrastructure Security (SIIS) Lab
Penn State University

Privilege Separation

PENNSTATE 1855

- Has been promoted for some time
  - Software-Fault Isolation (1993)
  - Kernel driver isolation (1990s)
  - OpenSSH (early 2000s)
- Can be a time-consuming task
  - Automate – not there yet
- Questions
  - What is the state of automating privilege separation?
  - Do we still need it?

PtrSplit

PENNSTATE 1855

- Automated privilege separation
  - Function partitioning
  - And IDL generation
  - To generate RPC (marshalling/unmarshalling) code

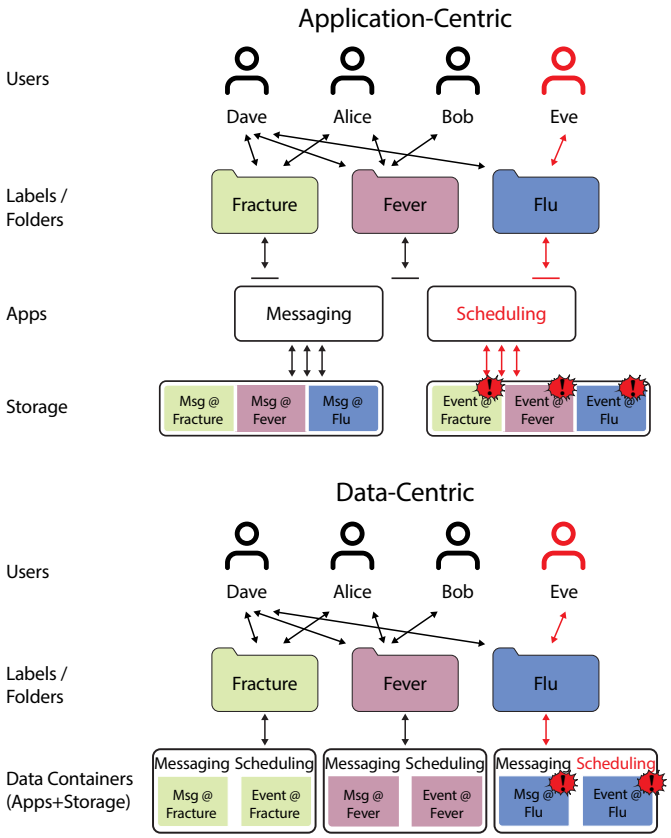
DATS

PENNSTATE 1855

- Combines a number of technologies that we have studied this semester into one system
- Single use services
  - Launch service for particular user/request
    - Unique web-application instance (container)
- Access control
  - Limit that single-use service to only the user/request permissions
    - Folder-level ACLs
- Privilege separation
  - Isolate untrusted front-end from processing of key data (folders)
    - Within web applications, but trust the backend storage (storage declassifier)

DATS

- Different view of sharing



DATS

- Different view of sharing

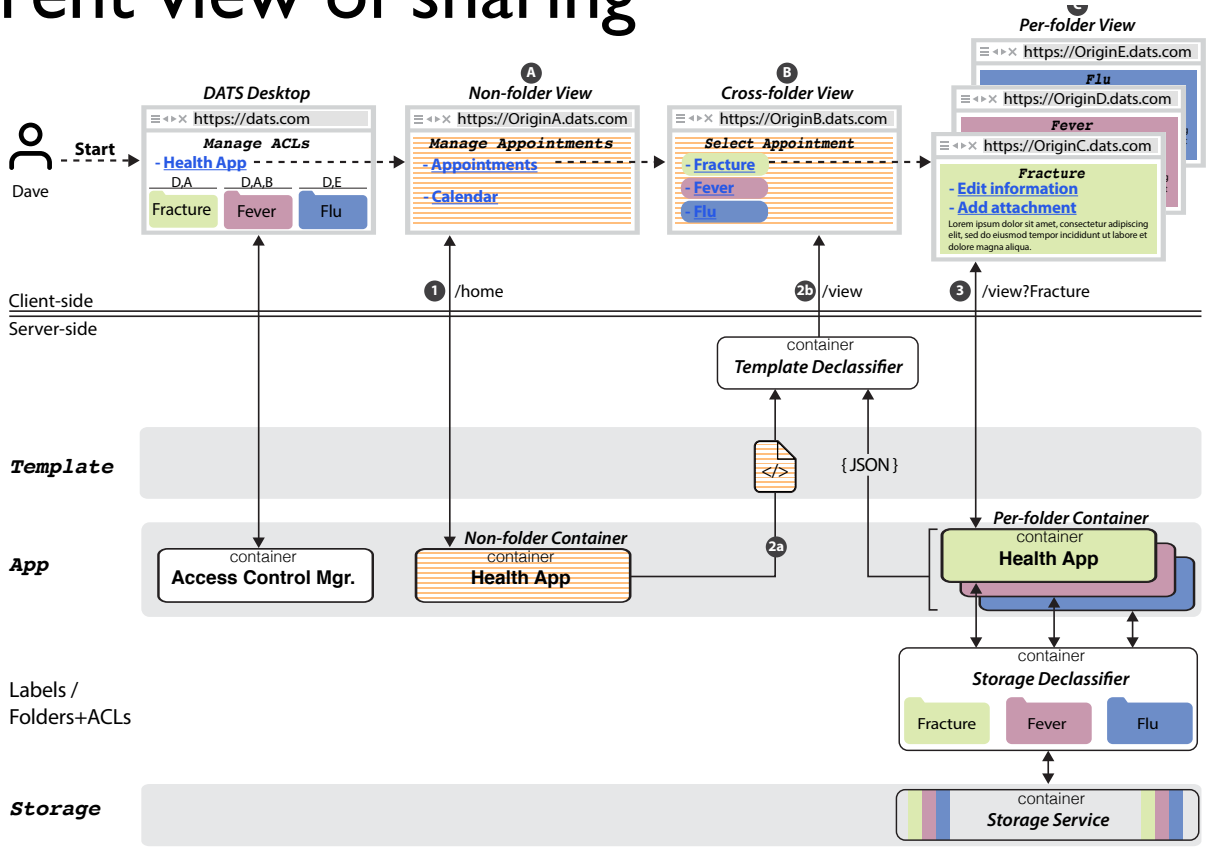


Figure 2. *Example web page flow from a user ("client-side"), DATS's main components, and an application's* `app-template-storage` *components (and their relation to MVC). Application code, application data, and storage services are untrusted (grayed areas and colored boxes), while DATS's trusted components (boxes with white background) enforce folder non-interference. Application components run inside OS-level containers, which can very easily enforce per-folder MAC policies. Note that the client's browser is allowed to run untrusted application code (e.g., JavaScript).*

DATS – Take Aways

PENNSTATE 1855

- Questions
- Do programmers know how to build web application instances?
  - Is this an automated privilege separation task?
- Can we enforce information flow guarantees comprehensively?
  - Currently SELinux
  - Should we use DIFC?
- Can we really trust the backend? Do we really need to?
  - Is this another privilege separation problem?