Security Architecture for Connected Vehicles

Architecture proposal for AGL-2.0
January-2016
Isolation & Segregation

• Client/UI (untrusted)
  – Risk of code injection (HTML5/QML)
  – UI on external devices (Mobiles, Tablets)
  – Access to secure service APIs only [REST]

• Applications & plugins (semi-trusted)
  – Unknown developers & Multi-sources
  – High grain protection by Linux UserID & SMACK labels.
  – Run under control of Application Framework: need to provide a security manifest

• Platform & System services (trusted)
  – Services started by DBUS
  – Fine grain privilege protection by Cynara
  – Part of baseline distribution and certified services only
Layered Security Architecture
HTML5, QML & Native Apps

Security framework should make standard operations simple, while keeping complex operations possible.

- **Standard Model**
  - UI under HTML5 or QML or external device running in the untrusted zone.
  - Application plugins accessed through REST APIs and control by authentication token provided by the application framework.
  - Platform services unmodified, Cynara control is handled transparently at DBus level.

- **AdHoc Model** *(when standard approach is not possible)*
  - UI and Application logic run directly at App-Level
  - Direct access to platform services bypassing DBus
  - Fine grain privileges accessed directly from a modified service daemon.
Sample Radio Application Startup

- (1+2) Home screen sends an “App Start” request through the corresponding binder to App. Framework service
- (3a+3b) App. Framework starts two processes with a shared secret.
  - Application Binder in charge of presenting Radio, Pulse & Multi-Media APIs
  - Radio UI in HTML5/QML running in a local webview or a remote HTML5 browser.
- (4) Radio client UI connects onto its binder and exchanges initial authentication secret as provided by App. Framework
- (5) Radio UI sends requests to PulseAudio through its binder API.
  - Pulse audio is unmodified and nevertheless under Cynara protection.
Sample Radio Application Flow

Security Starting Radio Sample

Isolation By Layer

Platform Trusted

App. Level Semi-Trusted

UI Level Untrusted

Segregation Of Duties

1. On request AppFramework start both a new binder and a new client.

2. App Binder gets a dedicated SMACK label, all logic is coded within plugins.

3a. Generic Shared Components

3b. Radio UI HTML5/QT

4. REST/HTML

5. DBUS Cynara Proxy

Pulse Audio Not Modified

UPnP Rygel Not Modified

Application Framework

Cynara

Application Binder

Per Application Binder Plugins

Radio Pulse M.Media

Per User System Daemons complemented if needed by SMACK Label

January-2016 AGL-2.0 Security Architecture Proposal 6
Conclusion

• Strong isolation
  – Untrusted client can only access services through a network interface and never have access to direct library mapping.
  – Application Binders in charge of presenting APIs to clients are constrained with a private SMACK label and run with userID rights.
  – Platform Services are protected by DBUS Cynara proxy and only receive permitted requests.

• Native apps and shortcuts remain possible
  – Services not compatible with a full isolation model, can bypass part of the security framework while still benefiting partially of it.

• Reduce costs of development
  – Compliant with external devices
  – Plugins are independent of Web Engine (browser) or Graphical Toolkit (QT and others)
  – DBUS platform services don’t need to be changed.
  – Compliant with standard Web/Mobile UI toolkit as Angular/Foundation.