Writing secure HTML5 applications for automotive systems

Manuel BACHMANN

<manuel.bachmann@iot.bzh>
Writing secure HTML5 applications for automotive

Table of contents

• Introduction

• HTML5 application development
  • Modern development methods
  • Platform APIs as HTTP REST APIs

• Application framework platform design
  • Application Framework Manager
  • Application Framework Binders(s)

• Privilege isolation through SMACK and Cynara
  • SMACK labels, Cynara policies
  • Security for Application Framework Binder
Introduction

What is an automotive system?

- Homescreeen
- AM/FM radio
- HVAC control
- Geolocation
- Media Player
- Phone
- Rear cameras
- Navigation helper
- Application manager
Introduction

Sample : Renesas Porter (R-Car M2)
Introduction

Application development prerequisites

• For developers
  • support modern HTML5/JavaScript frameworks
    *(AngularJS, Foundation, Polymer…)*
  • any W3C-compliant application should work out-of-the-box!
  • consuming platform APIs should be straightforward & easy!

• For users
  • installing, uninstalling, starting, stopping… apps
  • application privileges rights will be clearly exposed

• Security concerns, for developers & users
  • privileges need to be enforced at system level
Modern development methods

• Using modern JavaScript frameworks:
  • AngularJS
  • Foundation
  • Polymer

• With a full-fledged IDE:
  • Eclipse
  • NetBeans

• With a dev/prod build system:
  • Gulp
  • Grunt

• and an adapted browser:
  • Chromium with LiveReload extension
Platform APIs as HTTP REST

Template: \texttt{http://<board>/api/<plugin>/<method>(?value=<var>)}

- \texttt{http://<board>/api/radio/mode?value=FM} : select FM mode
- \texttt{http://<board>/api/radio/freq?value=110.2} : select 110.2 Hz

- demo AM/FM Radio app is written using AngularJS ;
- platform glue is provided in plugins, written in C/C++, JavaScript ;
- developers should be able to write and provide their own plugins...
- … but then, how do we :
  - package applications ?
  - enforce security ?
HTML5 application development

Demos: HTML5/JS Radio, Annex, Rabbit
Application Framework design

• Application Framework Manager \[^8\]
  - System daemon: installs, uninstalls, list… applications system-wide
  - User daemon (1 per user):
    • starts, stops, pauses… applications
    • when a HTML app starts, forks an Application Framework Binder with plugins & security context related to app category & privileges

• Application Framework Binder \[^9\]
  - is a lightweight web server, based on libmicrohttpd \[^10\]
  - loads platform plugins:
    • Audio, AM/FM Radio, Media Server…
  - provides platform APIs as HTTP REST APIs

• Web applications are displayed locally or remotely
Application Framework platform design

Application Framework design
Demo: installing & running applications

- Uploading - Installing
- Starting
- Stopping
SMACK labels, Cynara policies

- SMACK (Simplified Mandatory Access Control Kernel) [11]
  - is a LSM (Linux Security Module)
    *others include: SELinux, AppArmor...*
  - associates security labels to files, processes and streams;
  - “hard” security (system access denied on resource access)

- Cynara [12]
  - stores complex policies in databases;
  - “soft” security (access is checked by framework);

- Security concerns, for developers & users
  - SMACK labels are attached to user-level Binder
  - Cynara is requested by Binder
Privilege isolation through SMACK and Cynara

Security for Application Framework Binder
Demo: exploitation attempt
Annex
Links


Questions & Answers

Q&A
That's All Folks!