Application Security Model
Status Update

Genivi Oct/2016
Fulup Ar Foll Lead Architect
fulup@iot.bzh
2nd technical Contributor

- Application Development
  - SDK (binary distro + development ready docker container)
  - Application Framework (live cycle, cybersecurity provisioning)
  - Application Binder Framework (APIs exposure & protection)

- Integration
  - Yocto recipes
  - Releases automation & Testing (CI)
  - Renesas boards support
  - Security (MAC, Cynara, Systemd, CGroups, NameSpaces,...)

- Low Level Services
  - Audio Manager
  - Connectivity
  - Signaling & Events
  - SOTA
  - Secure Boot & Trusted Zone

- Community Support
  - Documentation (kickstart, developer samples, guides ...)
  - White Papers & Conferences (Genivi, AGL, Fosdem, ELC, ...)
  - Renesas Community support
Security Theory & Concepts Versus Real Implementation

- Untrusted Apps / Middleware
- API
- Harden OS services
- Linux Kernel with up-to-date patches
  - ID Management
  - Private/Secure Store
  - Secured Boot
- Software running on Target

- AppFW
- App Debug
- App Packaging
- Default policies
- Debug
- Sample code
- HowTo
- Signing
- Repo create
- Debug
- Customize
- SoC Drivers

Tools-Doc
AGL-2.0 Security Model

Automotive Grade Linux (AGL)

A Linux Foundation project dedicated to creating open source software solutions for automotive applications.
Make sure we Run the Right Code

- **Trusted Boot : a MUST Have Feature**
  - Leverage hardware capabilities
  - Small series & developer key handling

- **Application Installation**
  - Verify integrity
  - Verify origin
  - Request User Consent [privacy & permissions]

- **Update**
  - Only signed updates with a trusted origin
  - Incremental Update supporting increments jump
How to sign the fitImage in OpenEmbedded build system?

UBOOT_SIGN_KEYDIR = "/keys/directory"
UBOOT_SIGN_KEYNAME = "dev" # keys name in keydir (eg. "dev.crt", "dev.key")
UBOOT_MKIMAGE_DTCOPTS = "-I dts -O dtb -p 2000"
UBOOT_SIGN_ENABLE = "1"
TrustZone

- Two executions contexts: normal world & secure world,
- Peripherals visibility can be configured for each world,
- Integrated into the system on chip,

Credit:
http://genode.org/documentation/articles/trustzone
AGL SOTA Buildchain Integration

- Existing buildchain for AGL Images
- Every AGL distributor will need a similar setup
  Security, Licensing, Compliance etc.
- Developers / hobbyists can use AGL infrastructure
AGL Layered Architecture

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

- **Applications & Services (semi-trusted)**
  - Unknown developers & Multi-source
  - High-grain protection by Linux UserIDs & SMACK labels.
  - Run under control of Application Framework: need to provide a security manifest

- **Platform & System services (trusted)**
  - D-Bus Services started by systemd
  - Fine grain privilege protection by Cynara
  - Part of baseline distribution and certified services only
Application Security Framework

• Application Manager
  • One system daemon for application live cycle installs, update, delete
  • One user daemon per user for application start, stop, pause, resume
  • Create initial share secret between UI and Binder
  • Spawn and controls application processes: binder, UI, ...

• Security Manager
  • Responsible of privilege enforcement
  • Based on Cynara + Dbus plugin + WebSocket

• Application & Services Binders
  • Expose platform APIs to UI, Services, Applications
  • Loads services/application plugins :Audio, Canbus, Media Server…
  • One private binder per application/services [REST, WebSocket, Dbus]
  • Authenticate UI by oAuth token type
  • Secured by SMACK label + UID/GIDs
  • AppBinders runs under user $HOME
Application/Service Isolation

**System Services/Application**
- Create a dedicated UID per service
- Use DAC and MAC to minimize open access

**Drop privileges**
- POSIX privileges
- MAC privileges

**Cgroups**
- Reduce offending power
- RAM/CPU/IO

**Name Space**
- Limit access to private data
- Limit access to connectivity
AGL2 Application Security

Application Framework Live Cycle Management

- Navigation Service
  - Carte handling
  - POI management
  - etc...
- MultiMedia Service
  - Media Player
  - Radio Interface
  - etc...
- Log/Supervision Service
  - Carte handling
  - POI management
  - etc...
- Cgroups NameSpace Containers
- Transport + Access Control
- MAC Enforcement Smack
- Agent-2 Car Environment
  - CAN Bus-A
  - LIN Bus-A
  - Audio
- Agent-3 Engine
  - CAN Bus-B
  - Cluster-Unit
  - etc...
- Agent-4 Remote Signal
  - Smart City
  - RVI
  - Cloud

Distributed Application Architecture
Writing native AGL Applications

• **Write back-end binding**
  - Adds the specialised API to the system
  - Accessible by Web Socket or slow legacy D-Bus
  - Run in its own security domain
  - Can be cascaded

• **Write the Front end**
  - Typically in HTML5, QML but open to any
  - Connect to back-end binding using REST with secured key (OAuth2)

• **Package**
  - Based on W3C widget
  - Feature allow to handle AGL specificities
  - Install via the AppFW
Running Alien Applications

- Anything System-D can start
  - SMACK Label
  - Ressource constrains
  - Optionaly APIs privilèges

- Legacy Apps/Services
  - Typically access/export APIs directly
  - No fine grain API control
  - No transport Transparency

- New Transport (ex: RVI, BUS1, AUTOSAR, ...)
  - Interfaced with Binder mainloop (native)
  - Provided through a Proxy (gateway)
  - Need to pass process label (SMACK)
On Going Work
AGL2+ Distributed Architecture

Cluster
- Head Unix
  - Direction Indication
- Transport & ACL
  - Cluster Virtual Signal
    - Engine-CAN-BUS
    - ABS

Entertainment
- Navigation Service
  - Carte handling
  - Localisation management
  - POI
- CAN-BUS Virtual Signal
  - CAN-BUS
  - LIN-BUS
- Geopositioning Virtual Signal
  - Gyro, Accelerometer
  - CAN GPS

Cloud
- My Car Portal
  - Payment
  - Subscriptions
  - Preference
- Maintenance Portal
  - Know Bugs
  - Maintenances
  - Service Packs
- Log Analytics
  - No-SQL Engine
  - Statistics & Analytics

Multi ECU & Cloud Aware Architecture

AGL-2.0 Application Security Model
Oct-2016 16
AGL Framework Improvement

- Ressource control
  - Containers (LXC)
  - APIs enforced Timeout
  - LSM Secure Tagging of Processes *(long route before being accepted inside kernel)*

- Static Control
  - Introspection
  - OpenAPI for auto-testing
  - Permissions Mapping Tree

- Live Cycle
  - Debug *(change trace level, in/out hook, GDB attachement)*
  - Auto-start, Auto-restart
  - Integration with global system OTA
  - Trustzone integration for PKI and Chrypto
  - CRUI *(pronounced kree-oo)*
AGL2+ Driver Identity

Connected Car

- My vehicles
- Add vehicle
- Family
- Add Family Member
- Edit profile

![Connected Car Interface](image)

**Ford Focus**

**GENERAL**

**USERS**

<table>
<thead>
<tr>
<th>User</th>
<th>Permissions</th>
</tr>
</thead>
<tbody>
<tr>
<td>steinm</td>
<td>Install App, Open Trunk, View Online, Drive Over 60, Update Software</td>
</tr>
<tr>
<td>ddrazer</td>
<td>Install App, Open Trunk, View Online, Drive Over 60, Update Software</td>
</tr>
</tbody>
</table>

Stein Myrseth

Roles:
- User
- Owner
- Maintainer

Donald Draper

Roles:
- User
- Owner
- Maintainer

Permissions:
- Install App
- Open Trunk
- View Online
- Drive Over 60
- Update Software
ASIL Buildchain Integration

- ASIL buildchain will require secure process
- Images built and signed by secure / safe compiler
- OTA Server cannot process / modify images
XEN-4-AGL

• **End 2016**
  - High-level Isolation between VM's data: track VM’s IDs in SMC calls (provided by ARM architecture)
  - VM termination support: unload TAs “dead” VMs
  - Xen hypervisor support: re-routing SMC calls to domains other than dom0

• **Target for 2017**
  - Development of additional GlobalPlatform APIs with Linaro
  - Development of custom TAs: DRM, HW security features support, etc.
  - Extension of testing coverage with GlobalPlatform tools
  - Supporting OP-TEE upstreaming into kernel mainline
AGL-2+ Virtualized Architecture

- Trusted Zone
- Trusted Boot
- Integrity control
- PKI safe Store

- Hypervisor
- Diagnostics
- Emergency Services
- Resources Alloc/Proxy

- AGL Linux Supervisor
- AGL Core Platform Services
- AGL Extra Middleware

- AGL Linux Kernel Guest Operating
- AGL App-1
- AGL App-2
- AGL App-3

- Container
- DomU Entertainment
- DomU Cluster

- App-1
- App-2
- AGL Mini Platform Services
- Linux-RT/Microkernel Guest Operating

- Virt GPU
- Virt Audio

Virtualized Secure Architecture
Further Information

• Some References
  • https://www.automotivelinux.org/automotive-grade-linux-security-white-paper
  • http://bgr.com/2015/10/13/why-is-android-security-so-bad/

• Download links
  • AppFramework code https://gerrit.automotivelinux.org/gerrit
  • AGL-2.0 SDK http://iot.bzh/download/public/2016/sdk
  • Meta-IOT-Layer https://gerrit.automotivelinux.org/gerrit