Software Management
Thursday 4/28 14:00 – 16:00
Arthur T. Taylor
CTO – ATS
Contributor – Systems Infrastructure EG
SOTA / SCOTA / FOTA / OTA

- Delivering updates to vehicles in the field, over the internet
- Urgent requirement for existing and future IVI systems
- Non-differentiating feature - has to work

Existing solutions

- Red Bend (Harman) - feature-rich, fully proprietary
- CarSync (Arynga) - Part of Arynga's proprietary server-side software suite, open source client
- OMA DM / FUMO - Open Standard - complex, some parts available as open source
- The Update Framework [1] - designed to overlay on existing solutions
- "Roll your own" - time consuming, insecure
- JLR commissioned end-to-end open source solution

1. http://theupdateframework.com
SOTA / SCOTA / FOTA / OTA

• Delivering updates to vehicles in the field, over the internet
• Urgent requirement for existing and future IVI systems
• Non-differentiating feature - has to work

Existing solutions

• Red Bend (Harman) - feature-rich, fully proprietary
• CarSync (Arynga) - Part of Arynga's proprietary server-side software suite, open source client
• OMA DM / FUMO - Open Standard - complex, some parts available as open source
• The Update Framework [1] - designed to overlay on existing solutions
• "Roll your own" - time consuming, insecure
• JLR commissioned end-to-end open source solution
• OTA Plus – ATS commercially ready OTA solution based on GENIVI SOTA

1. http://theupdateframework.com
SOTA - PROJECT BACKGROUND

GENIVI is a registered trademark of the GENIVI Alliance in the USA and other countries.
This work is licensed under a Creative Commons Attribution-Share Alike 4.0 (CC BY-SA 4.0)
SOTA - FEATURES

• Package upload
• Software / Firmware / Hardware inventory management by VIN
• Package installation campaign management
  • Filtering
  • Scheduling
  • Tracking
• Package download and installation
  • Reliable delivery by RVI
  • Interface with Software Loading Manager / local package manager
  • Success / Failure reporting to the server
LIVE DEMO!
SOTA CORE - ROADMAP

- Multi-tenant, multi-user support
- Standardisation of SOTA Web API with W3C
- HTTPS Transport support
- OAuth2 / SSO integration
- Configuration Management
- File Upload
- Migration to github.com/genivi
- Implementation of Security recommendations from UPTANE / AutoTUF
What do these companies have in common?
They all had a publicly disclosed software update hack!
Repository compromise impact

- **SourceForge mirror distributed malware.**
- Attackers impersonate Microsoft Windows Update to spread Flame malware.
- **RubyGems compromised** with RCE.
- Opera users automatically installed malware signed by compromised key.
- **Node Packaged Modules compromised.**
- **Attacks** on software updaters have massive impact
  - E.g. South Korea faced **765 million dollars** in damages.
Why Are Software Updaters Important?

• Run as root

• Update / change software!
  – Goal of most attacks

• Traditional defenses are ineffective
  – Firewalls / IDSes do not protect against these attacks

• Ubiquitous

• An attack often appears benign
  – Can look like a software repository misconfiguration
Possible Attacks on Software Updates

• Malicious package
  – Crash / DoS / Arbitrary Code execution

• Version attacks
  – Downgrade to vulnerable version

• Metadata attacks
  – Change signatures, add bad dependencies

• Key Compromise
  – TLS root key, code signing key
Path to failure...

• Have everyone build their own “secure” updater

– the problem is subtle
2010: The Update Framework (TUF)

- TUF is a **secure** software update **framework**.
- Built using some ideas from Thandy.
- Developed by security researchers from NYU.
- **Best practices** to protect users from security attacks.
- **Plug-and-play** (like TLS).

“Survivable Key Compromise in Software Update Systems” (**CCS 2010**).
2010: The Update Framework (TUF)

• Standard mitigations and best-practices
• Resilient to Key Compromise  
  – Multi-party signatures  
  – Roles / Delegation  
• Equivocation  
  – Detect "back-doored" packages
2016: UPTANE / AutoTUF

- NYU / UMich collaboration, funded by DHS
- Including leading players
  - OEM, Tier 1, OTA Vendors
- GENIVI SOTA will align with UPTANE
SOTA CLIENT - ROADMAP

• Integration with GENIVI Development Platform
• Integration with AGL / Open IVI
• Support for generic Linux platforms (32-/64-bit RPM/Deb)
• Support for Binary Diffs
• Signature Validation
• Security hardening
• C/C++ implementation?
SOTA – WHAT WE PROVIDE

• **Server-side**
  • Server implemented in Scala
  • Deployment scripts / instructions for AWS, Docker and developer machines
  • Complete Design, Architectural and API documentation
  • Complete test suite

• **Client-side**
  • Client implemented in Rust
  • Build / installation scripts for GDP
  • Complete test suite
  • Integration with GENIVI Software Loading Manager
OPEN SOURCE!

All software available under MPL v.2!
OPEN SOURCE!

GENIVI SOTA Project — Software, Over the Air

“It is a truth universally acknowledged, that a software distribution in possession of many component parts must be in want of a good update manager.”

— Jane Austen
Pride, Prejudice & Software Development

1. Updating software is important.
   It’s even more important when it’s software running on 3 metric tons of fast-moving steel.

2. Getting your users to update software is hard.
GET INVOLVED!

Github

https://github.com/genivi/rvi_sota_server
https://github.com/genivi/rvi_sota_client

Mailing lists – GENIVI Projects
TRY IT OUT!

- Launching this summer: SaaS version of OTA Plus
- Based on GENIVI SOTA
- Minimise develop/build/test/deploy cycle
- Allow engineers to focus on differentiating value
- Support for any GENIVI SOTA Client device