Evolving Development and Test Strategies in the Automotive Industry

Raghuram Sudhaakar Ph.D.
Senior Technical Leader, Cisco Systems Inc
Evolution of the In-Vehicle Network

- Virtual Function Bus Architecture
- End-to-End Service Orientation
- Comprehensive Telemetry export capabilities
- Enables sensor fusion & autonomous driving with redundant compute

© 2019 Cisco and/or affiliates. All rights reserved
Automotive EE Architecture in Transition

**Individual ECU Based**
- Rigid compute
- Domain connectivity
- Wiring complexity

**Domain Controller**
- Compute flexibility for domain
- Domain connectivity
- Wiring complexity

**Virtualized Central Compute**
- Cross domain compute flexibility
- Domain connectivity
- Wiring complexity

**Flexible Compute & Network**
- Full compute flexibility
- Localized connectivity
- Wiring Simplicity

Flexibility, Simplicity, Modularity

Legacy

Incremental

Technology Alignment

Autonomous

© 2019 Cisco and/or affiliates. All rights reserved
Intra-Vehicle Network Development

**Current**
- Network Design = One master DB
- Process is manual
- Traceability is difficult
- Suppliers receive an ECU subset file
- Suppliers deliver a Black Box
- Validation occurs only in a Proto Vehicle
- Single DB for all variants creates security weakness
- In-field feature enablement is cumbersome
- Network changes for safety issues results in recalls

**Future**
- Model based approach for network design and validation
- Enables Rapid design Iteration
- Supports Automotive Processes (ISO 26262, V-design)
- Automation leveraged for vendor ECU development
- Network communications no longer a black box strategy
- Full simulation and validation of network at design stage
- Model variant based network design enhances security
- In-field Feature enablement and safety fixes requiring network changes are simplified
Connected Vehicle Attack Surfaces

- GM’s OnStar RemoteLink
  July 2015
- Jeep Cherokee Hack, July 2015
- Nissan Leaf
  Feb. 2016
- BMW ConnectedDrive
  July 2016
- Tesla Model S
  August 2016
- Tesla Model S
  August 2017
- Tesla Model S
  late 2018

Experimental Analyses of Automotive Attack Surfaces
UW, UCSD, Usenix 2011
Policy based Runtime

**System Policies**
Defines system level behavior
Examples: System Logging Level, Application Prioritization, etc.

**Network Policies**
Manage and control data flows.
Ensure transmitter, receiver and data flow are all valid and permitted
Priority classification

**Security Policies**
Authenticate devices, external connectivity, etc.
Device Identity
Prevent misbehaving devices from doing any damage to system
Permit and manage after-market devices?
A different view to metrics and trends

Cyclomatic Complexity, Modularity, Dead code

Cost of (NOT) refactoring

Adopting agile workflows while maintaining quality

Feature Test, Functional Test, E2E Test
Scale and Performance in parallel

Trends and (new) metrics

How many times a module of code changes
How many times is a logic block changed

© 2019 Cisco and/or affiliates. All rights reserved
Changing the philosophy of testing

- Identify gaps in specification: Feature level test plan will be developed before development starts.
- Parallel test infra and development progress: Detailed test plan will be developed before the start of dev for each release cycle.
- Previously found defects do not re-occur: Each release is fully regression tested with the previous release. Regression is fully automated with real DUT’s.
- New features do not affect code stability: Agility towards requirement changes and clarification. Nightly regression.
The “People”

Culture! Everyone needs to be involved

Peer review process, Checklists, Strict style guidelines

Significantly more rigor in unit testing

Security must be an integral part of the software design

Don’t be afraid to throw away wrapper functions and internal APIs

Maintain modularity