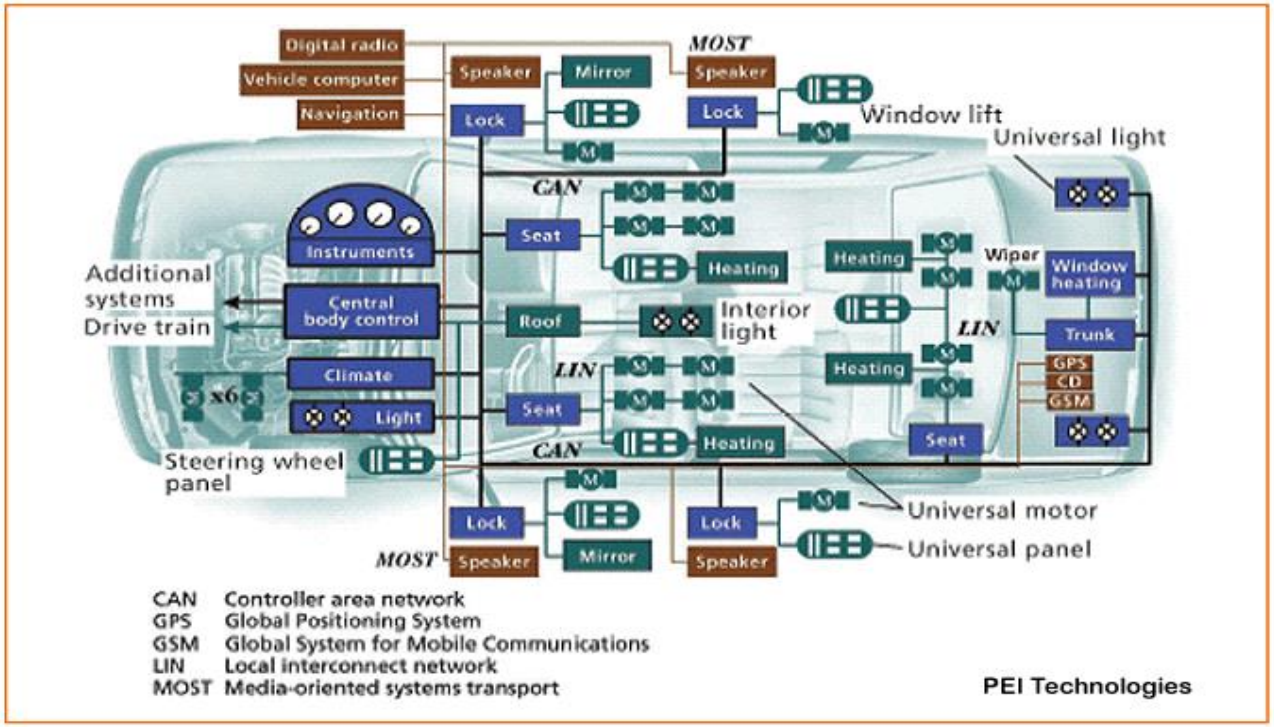
Automobile Computer Security

- Modern car systems
- Threat models
- Vulnerability analysis

The Modern Car

- Shift of modern cars towards control by distributed computing systems
  - Systems controlled by tens of Electronic Control Units (ECUs)
    - Entire system consists of millions of lines of code
    - Multiple separate communication buses → Security driven?
  - ECUs interconnected by common wired networks
    - i.e., Controller Area Network (CAN), FlexRay bus, …



Controller Area Network

- Comprised of 2 buses
  - High speed bus: safety critical, more trusted
  - Low speed bus: non-critical, convenience modules
- A gateway can route things between the buses
- Required for diagnostics in all cars sold in US since 2008

List of ECUs

Component	Functionality	Low-Speed Comm. Bus	High-Speed Comm. Bus
ECM	*Engine Control Module* Controls the engine using information from sensors to determine the amount of fuel, ignition timing, and other engine parameters.		✓
EBCM	*Electronic Brake Control Module* Controls the Antilock Brake System (ABS) pump motor and valves, preventing brakes from locking up and skidding by regulating hydraulic pressure.		✓
TCM	*Transmission Control Module* Controls electronic transmission using data from sensors and from the ECM to determine when and how to change gears.		✓
BCM	*Body Control Module* Controls various vehicle functions, provides information to occupants, and acts as a firewall between the two subnets.	✓	✓
Telematics	*Telematics Module* Enables remote data communication with the vehicle via cellular link.	✓	✓
RCDLR	*Remote Control Door Lock Receiver* Receives the signal from the car's key fob to lock/unlock the doors and the trunk. It also receives data wirelessly from the Tire Pressure Monitoring System sensors.	✓	
HVAC	*Heating, Ventilation, Air Conditioning* Controls cabin environment.	✓	
SDM	*Inflatable Restraint Sensing and Diagnostic Module* Controls airbags and seat belt pretensioners.	✓	
IPC/DIC	*Instrument Panel Cluster/Driver Information Center* Displays information to the driver about speed, fuel level, and various alerts about the car's status.	✓	
Radio	*Radio* In addition to regular radio functions, funnels and generates most of the in-cabin sounds (beeps, buzzes, chimes).	✓	
TDM	*Theft Deterrent Module* Prevents vehicle from starting without a legitimate key.	✓	

Table I. Key Electronic Control Units (ECUs) within our cars, their roles, and which CAN buses they are on.

- Offer significant benefits to efficiency, safety, and cost
- but, are these systems secure?
  - no large scale attack yet
  - need to understand potential security risks when cars become more and more connected

A first look ...

CAN packets: header that says where the packet goes

- No addresses used
- All packets broadcast to all nodes
- Each node decides if it should process the packet

Vulnerabilities:

- All nodes see all traffic
- All nodes communicate with all other nodes
- DoS-able
- No identifiers
- Firmware updates
- Weak access controls to sensitive info
- Protections often ignored by ECUs

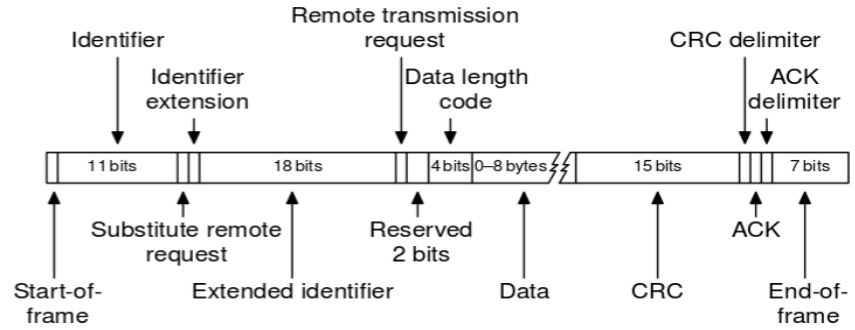
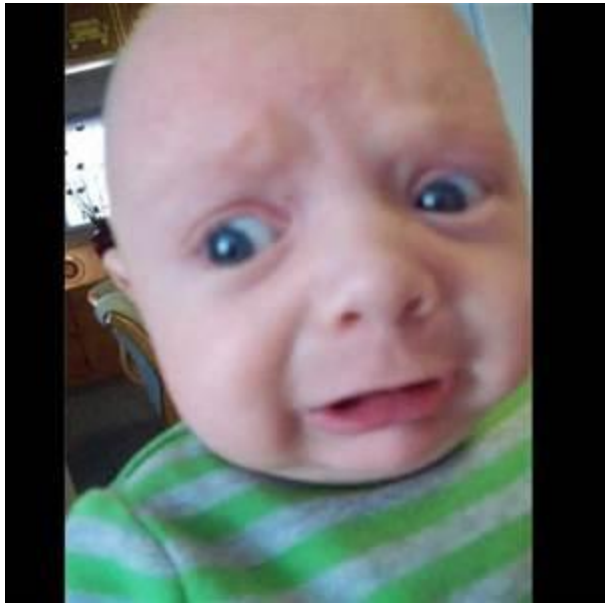


Figure 5. CAN packet structure. Extended frame format is shown. Base frame format is similar.

Seed to key algorithm

- Authentication method for sensitive operations
  - One ECU sends the seed (the challenge)
  - The other replies with the key
- Each ECU has its own seed and key
- Keys and seeds are fixed and stored in the memory of each ECU
- Algorithms used to compute them are not stored in ECUs for "security"!!!
- Return of challenge not always used



- Comprehensive experimental analyses of automotive attack surfaces
  - By researchers from UC San Diego and Univ. Washington (USENIX Security 2011, Oakland'2010)

Outline

1. Threat model characterization
   - Systematically synthesize a set of *possible* external attack vectors
2. Vulnerability analysis
   - For each attack vector category, investigate one or more concrete attacks in depth
3. Threat assessment
   - Utility to an attacker
4. Synthesis
   - Security recommendations to raise the bar to attackers

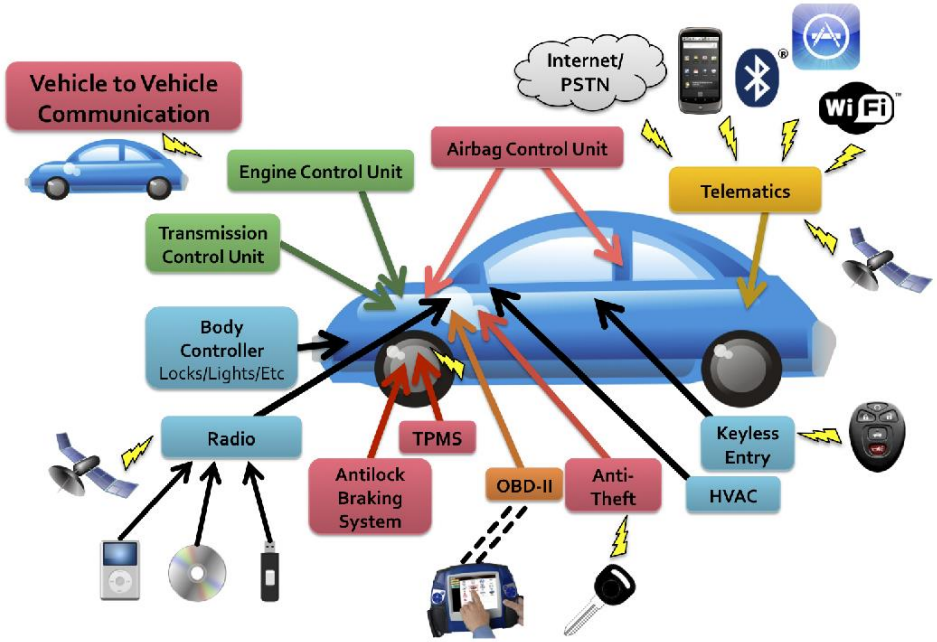
1. Threat models

- Technical Capabilities → Adversary capabilities in analyzing the system and developing exploits
  - Focus on making technical capabilities realistic

- Operational Capabilities → Adversary capabilities in delivering a malicious input to a particular vector in the field
  - Direct physical access
  - Indirect physical access
  - Short-range wireless access
  - Long-range wireless access

Attack Vectors

- Direct physical access
  - OBD-II (PassThru)
- Indirect physical access:
  - OBD-II (PassThru)
  - Entertainment system
- Short-range wireless access:
  - Bluetooth
  - Remote Keyless Entry
  - Tire Pressure (TPMS)
  - Wifi
  - Emerging: DSRC
- Long-range wireless access:
  - GPS
  - Satellite Radio
  - Digital Radio
  - Remote Telematics Systems
    - Ford's Sync, GM's OnStar, Toyota's SafetyConnect, Lexus's Enform, BMW's BMW Assist, …



OBD-II

- The most significant automotive interface, federally mandated in the U.S.
- Provide direct access to the automobile's key CAN buses for diagnostics and ECU programming
  - Ford's NGS, Nissan's Consult II, Toyota's Diagnostic Tester
- A laptop computer interfaces with a "PassThru" device (via USB or WiFi) that in turn is plugged into the car's OBD-II port

2. Vulnerability Analysis

- Direct physical access
- Indirect physical access
- Short-range wireless access
- Long-range wireless access

Experimental Context

Moderately priced late model sedan with the standard options and components

- 100,000~200,000 of this model were produced in the year of manufacture
- Car includes < 30 ECUs controlling
- Exposes to a number of external vectors
  - OBD-II port, media player, satellite radio, RDS, telematics unit
- Purchased multiple replacement ECUs and a PassThru device

<u>Every vulnerability demonstrated allowed complete control of vehicle's system</u>

- General Procedure:
  - Identify microprocessor (PowerPC, ARM, Super-H, etc)
  - Extract firmware and reverse engineering using debugging devices/software where possible
  - Exploit vulnerability or simply reprogram ECU

Exploitation Summary

Vulnerability Class	Channel	Implemented Capability	Visible to User	Scale	Full Control	Cost
Direct physical	OBD-II port	Plug attack hardware directly into car OBD-II port	Yes	Small	Yes	Low
Indirect physical	CD	CD-based firmware update	Yes	Small	Yes	Medium
	CD	Special song (WMA)	Yes*	Medium	Yes	Medium-High
	PassThru	WiFi or wired control connection to advertised PassThru devices	No	Small	Yes	Low
	PassThru	WiFi or wired shell injection	No	Viral	Yes	Low
Short-range wireless	Bluetooth	Buffer overflow with paired Android phone and Trojan app	No	Large	Yes	Low-Medium
	Bluetooth	Sniff MAC address, brute force PIN, buffer overflow	No	Small	Yes	Low-Medium
Long-range wireless	Cellular	Call car, authentication exploit, buffer overflow (using laptop)	No	Large	Yes	Medium-High
	Cellular	Call car, authentication exploit, buffer overflow (using iPod with exploit audio file, earphones, and a telephone)	No	Large	Yes	Medium-High

2.1 Direct physical access

- Researchers used two identical 2009 model cars
- Wrote a packet sniffer/injection tool, introduced into the CAN by simply plugging a device into the car's federally mandated universal *OBD*-II diagnostics port
- Used "fuzzing" to enumerate the commands that the car responds to
- Using the commands they discovered, performed live tests to see how much of the car they could control

Results

Researchers could not only fully control the car using their device, they could do it while the car was going 40 MPH

Among the things they could control:

- Disable brakes
- Engage brakes
- Disable wipers and continuously spray fluid
- Permanently activate horn
- Kill engine
- Unlock all doors

Also found that they could write programmatic commands, or "viruses", that would activate under certain conditions

- Disable all lights when driving over 40MPH

Even though they had physical access to the CAN, they noted that the same commands could potentially be executed wirelessly

2.2 Indirect Physical Exploits

Media Player → found two exploits

1) Latent update capability of player manufacturer
    - Automatically recognize an ISO 9660-formatted CD with a particularly named file
    - Present a cryptic message to the user
    - Reflash the unit when user does not press the right button
2) WMA parser vulnerability
    - Buffer overflow vulnerability
        - Allows execution of arbitrary code
    - Audio file parse correctly on a PC - In vehicle send arbitrary CAN packets

Indirect Physical Exploits Ctd.

OBD-II:

- Looked at PassThru device from manufacturer (used on all their production vehicles)
- Found no authentication for PC's on same WIFI network
- Found exploit allowing reprogramming of PassThru
  - Allows for PassThru worm
  - Allows for control of vehicle reprogramming
  - Includes unsecured and unused Linux programs

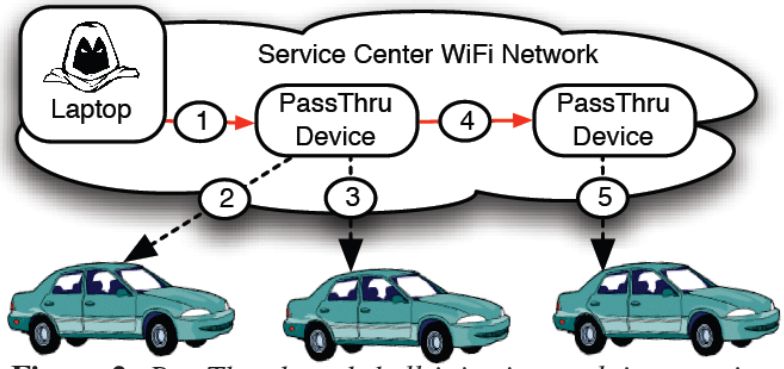


Figure 2: *PassThru-based shell-injection exploit scenario.* The adversary gains access to the service center network (e.g., by compromising an employee laptop), then (1) compromises any PassThru devices on the network, each of which compromise any cars they are used to service (2 and 3), installing Trojan horses to be activated based on some environmental trigger. The PassThru device also (4) spreads virally to other PassThru devices (e.g., if a device is loaned to other shops) which can repeat the same process (5).

2.3 Short-Range Wireless Exploitation

Bluetooth: allow the occupants' cell phones to connect to the car (e.g., to enable hands-free calling).

- Found popular Bluetooth protocol stack with custom manufacture code on top
  - Custom code contained 20 unsafe calls to *strcpy()*
- Indirect attack → assumes attacker has paired device
  - Implemented Trojan on Android device to compromise machine
- Direct attack → exploits without a paired device
  - Requires brute force of PIN to pair device (10 hours) → Limited by response of vehicle's Bluetooth

2.4 Long-range Wireless Exploitation

Telematics Connectivity:

- Focus on cellular capabilities built into the experimental car's telematics unit
- Similar to Bluetooth → 3rd party device with manufacturer code on top
  - ❖ Again found exploit in transition from 3rd party to manufacturer "Command" program for data transfer
  - ❖ Lucky for manufacturer → bandwidth did not allow exploit transfer within timeout
    - Exploit required of authentication code
      1) Random nonce not so random
      2) Bug that allows authentication without correct response

3. Threat Assessment

- Have shown that gaining access to a car's internal network provides sufficient ***means*** for compromising all of its systems
- Also demonstrated that an adversary has a practical ***opportunity*** to effect this compromise without having physical access to the vehicle

Now we want to answer:

- What's the ***motive***?
- How serious are the threats?

Threat Motivation

Theft:

- Scary version → mass attack cellular network creating vehicle botnet
  - Able to have cars report VIN and GPS
  - Enterprising thief
  - Concrete exploit
    - unlock doors, start engine and fully startup car
    - cannot disable steering column lock

Surveillance:

- Allows audio recording from in-cabin microphone

4. Synthesis

- Why vulnerabilities exist today
- Problems
  - CAN is an insecure low-level protocol
  - Every message is an unencrypted plain-text broadcast to every device on the CAN
  - Possible messages and communication procedures are often documented and made available freely
  - No component authentication
  - Any device can send a command to any other devices.
    - Attacker could use tire pressure gauge to turn off brakes

Implementation Fixes

Restrict access

- Many interfaces open to unsolicited communications
- Not allow Bluetooth paring without dirver's intervention
- Cellular interface: let the car dial out for requests, not for inbound data transfer
- …

Improve code robustness

- Standard security engineering best-practices e.g. don't use unsafe *strcpy* → instead *strncpy*
- Removing debugging symbols and error strings
- Use simple anti-exploitation mitigations such as stack cookies and ASLR (Address Space Layout Randomization)
- Remove unused services e.g. telnet and ftp
- Behavioral monitoring for critical services
- Secure software update as part of automotive component design
- …

Vulnerability Drivers

- Vulnerability causes:
  - Lack of adversarial pressure
  - Conflicting interests of ECU software manufacturers and car manufacturers
    - Ex: Telematics, Bluetooth & Media Player
    - Penetration testing?

- Will it evolve like PC security?

Conclusions

Auto makers need to understand the seriousness of having networked car components and take security measures accordingly

These security flaws are growing in seriousness as cars automate more and more things

- Electric steering and acceleration

References

Experimental Security Analysis of a Modern Automobile
http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5504804&tag=1

Security in Automotive Bus Systems
http://weika.eu/papers/WolfEtAl_SecureBus.pdf

Hacking Cars
http://dl.acm.org/citation.cfm?id=2018396 (pg 18)

Highway Robbery: Car Computer Controls Could Be Vulnerable To Hackers
http://www.scientificamerican.com/article.cfm?id=wireless-car-hacking