

Fast and Vulnerable

A Story of Telematic Failures

Center for Automotive Embedded Systems Security
Ian Foster, Andrew Prudhomme, Karl Koscher,
and Stefan Savage



UCSDCSE

Telematic Control Units

- Connects to car's OBD-II port
- Monitors vehicle state
- Local sensors
 - GPS
 - Accelerometers
- Transmits data off device
 - Cellular, WiFi, Bluetooth
- Common uses:
 - Fleet tracking
 - Remote diagnostics



Our TCU

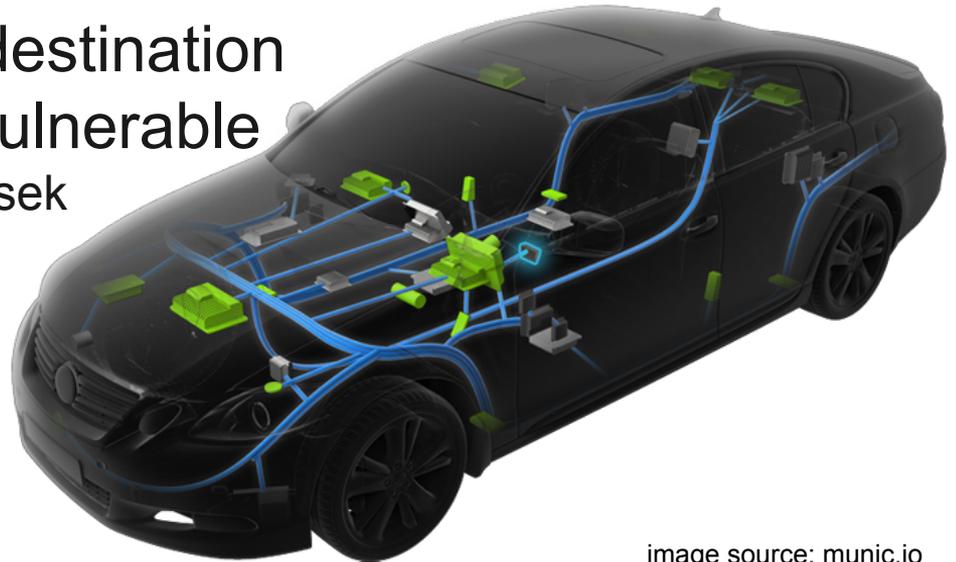
Mobile Devices Ingenierie - C4E (munic.box)

- ARM 11 500MHz CPU
- 64 MB RAM
- 256 MB Flash Storage
- Sensors
 - GPS
 - 3D accelerometer
 - 3 axis gyroscope
- Communication
 - GSM modem
 - USB “Debug” port
 - OBD Connector



Controller Area Network (CAN Bus)

- Connects various ECUs in cars
- Message based protocol
- Identifier for addressing destination
- Previously shown to be vulnerable
 - Charlie Miller and Chris Valasek
 - UCSD & UW



Attack Surface

Local

- USB “debug” port
- NAND flash

Adversary has physical access to the TCU. Do not consider the automobile communications in this model.

Remote

- SMS
- 2G/3G

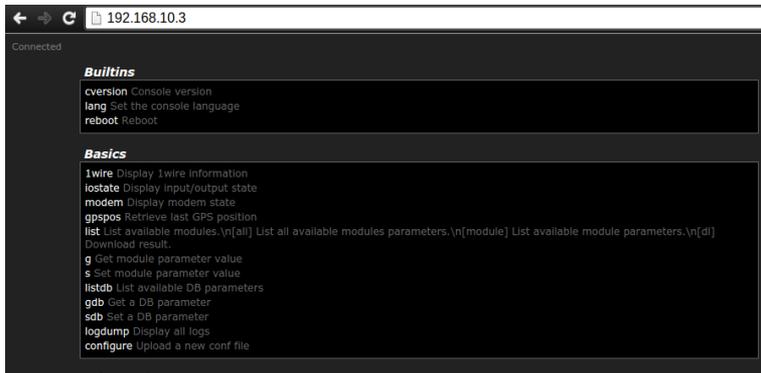
Adversary does not have physical access to the TCU, and may not even know where the TCU is geographically located.

Local Attacks



Debug Interface

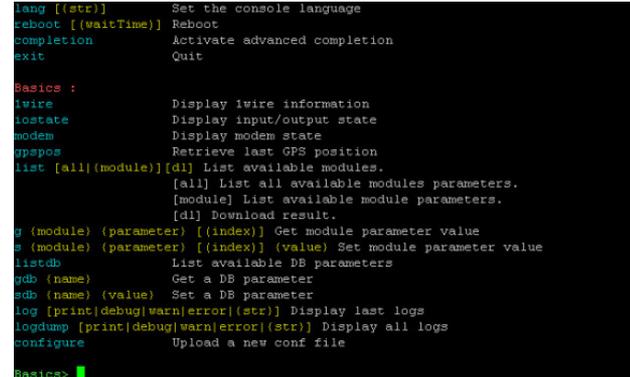
- Exposes USB network
 - Web & Telnet server for debug “console”
 - SSH Server
 - FTP Server for log retrieval and update uploading



```
← → 192.168.10.3
Connected

Builtins
cversion Console version
lang Set the console language
reboot Reboot

Basics
!wire Display Iwire information
!iostate Display input/output state
!modem Display modem state
!gpspos Retrieve last GPS position
list List available modules.\n[all] List all available modules parameters.\n[module] List available module parameters.\n[d] Download result.
g Get module parameter value
s Set module parameter value
!listdb List available DB parameters
gdb Set a DB parameter
sdb Set a DB parameter
logdump Display all logs
configure Upload a new conf file
```



```
lang [(str)] Set the console language
reboot [(waitTime)] Reboot
completion Activate advanced completion
exit Quit

Basics :
!wire Display Iwire information
!iostate Display input/output state
!modem Display modem state
!gpspos Retrieve last GPS position
list [all|(module)][d] List available modules.
[all] List all available modules parameters.
[module] List available module parameters.
[d] Download result.
g (module) (parameter) [(index)] Get module parameter value
s (module) (parameter) [(index)] (value) Set module parameter value
!listdb List available DB parameters
gdb (name) Get a DB parameter
sdb (name) (value) Set a DB parameter
log [print|debug|warn|error|(str)] Display last logs
logdump [print|debug|warn|error|(str)] Display all logs
configure Upload a new conf file

Basics>
```

NAND Dump

- Filesystem layout pulled from debug logs
 - dmesg
- NAND flash removed and dumped
 - de-soldered & read using hardware reader
- NAND flash simulated from dump
 - nandsim Linux kernel module
- Partitions mounted for reading
 - Unsorted Block Image File System (UBIFS)



SSH

Mounting the NAND flash dump revealed the private key for the root user.

SSH

Mounting the NAND flash dump revealed the private key for the root user.

The same private key worked on all C4 TCUs we tested.

SSH

Mounting the NAND flash dump revealed the private key for the root user.

The same private key worked on all C4 TCUs we tested.

`/etc/shadow` was identical across devices and included weak passwords.

CAN Bus Capabilities

- PIC Coprocessor
 - Used by devices with older firmware.
 - Custom interface for sending & receiving can messages.
 - Required ACC or ignition to be on to function.
 - Bypassed by reflashing PIC firmware without this check.
- SocketCAN
 - Used on devices with newer firmware.
 - Exposes the CAN interface as a traditional network interface.
 - Shipped with can-utils package.
 - Supports reading, saving, creating, and replying CAN messages.

Local Access Summary

- No authentication for debug consoles
- USB provides root access via web, telnet console, and SSH.
- Can send and receive arbitrary CAN messages.

Remote Attacks



IP (2G)

- All services bound to all network interfaces.
 - web
 - telnet console
 - SSH
- Same local network attacks work over the internet.
- Some devices protected by wireless carrier's NAT implementation.

SMS

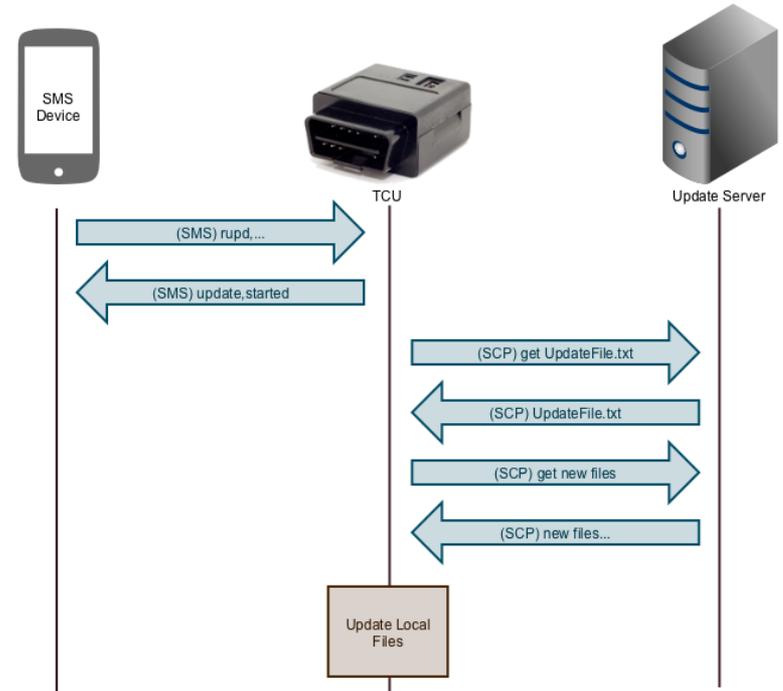
The device responds to SMS “commands”

Examples:

- status
- gps position
- reset
- remote update

Normal Update Procedure

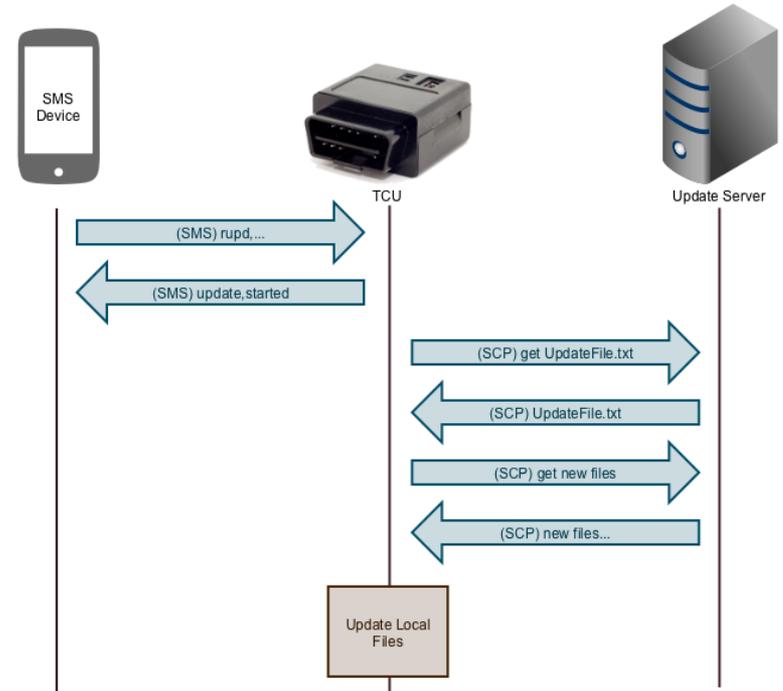
1. SCP UpdateFile.txt from update server to device
2. SCP new files from UpdateFile.txt from update server to temp folder
3. Move new files from temp folder to destination directory
4. Optionally perform an additional action
 - a. clear
 - b. identify
 - c. status
 - d. reset



Normal Update Procedure

Problems

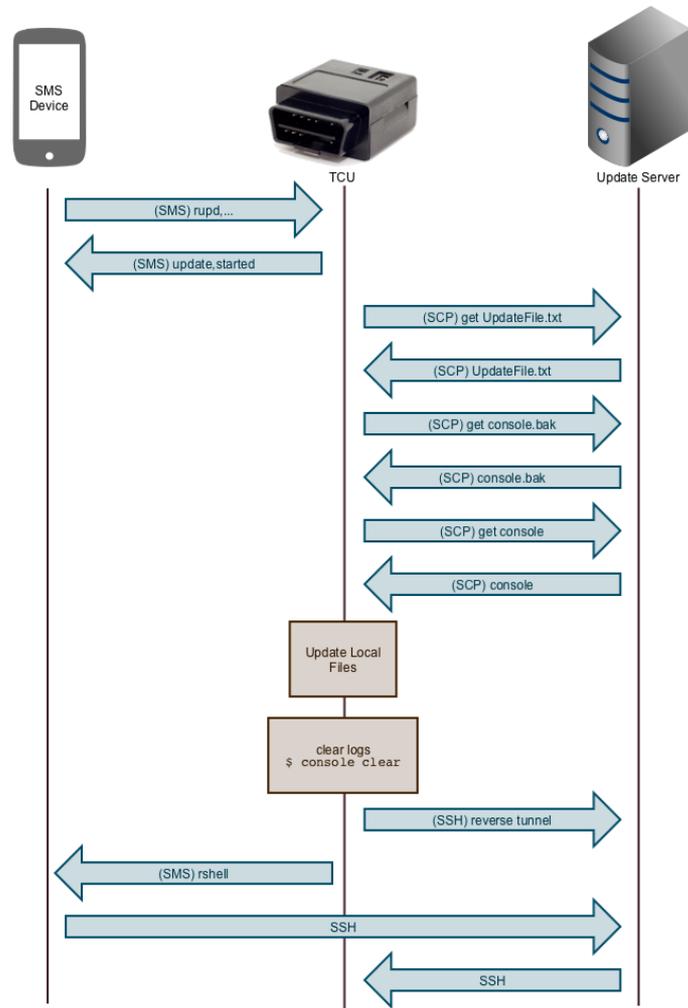
1. Updates are not cryptographically signed.
2. TCU does not authenticate the update server, instead the update server authenticates the TCU.



Exploiting Update

Replaced a binary (console) that was called post update to execute commands:

1. Replace console with console.bak (original)
2. Start reverse SSH tunnel to update server
3. Send SMS notification when reverse shell is ready
4. Execute original console command



Remote Access Summary

- Same local debug consoles exposed remotely.
- SMS allows access if wireless carrier uses NAT.
- Can obtain root shell from IP or SMS.
 - Send arbitrary can packets remotely.
 - Get GPS coordinates remotely.

Finding Devices

- Need to know either IP address (without NAT) or SMS number.
- SMS numbers were found to be from the 566 area code, which is reserved for “personal communication devices”
- Numbers were not random; appeared to be sequentially assigned.
- Could likely enumerate them all by sending a “status” SMS request to all numbers.

Shodan Search

SSH Server Fingerprint

TOP COUNTRIES



Spain	1,162
United States	121
Portugal	88
Netherlands	65
Sweden	63

Telnet Console Prompt

TOP COUNTRIES



Spain	2,622
Chile	246
United States	153
Germany	111
Sweden	97

Proof of Concept Attack



Proposed Solutions

1. Require update authentication
2. Disable remote SMS administration
3. Don't distribute identical private keys
4. Use strong passwords
5. Disable WAN administration
6. Require debug console authentication
7. Maintain update server

Disclosure

- June 29th - Reach out to Mobile Devices with details of vulnerabilities
- July 2nd - Mobile-devices responds
 - Developer SIM
 - Advanced debug mode
 - Older software version
- July 8th - Reach out to Metromile with details of vulnerabilities
- July 8th - Metromile responds, will disable debug mode and disable SMS.

Disclosure - CERT

- July 12th - Inform CERT of vulnerabilities found in C4 platform
- July 14th - CERT responds, assigned vulnerability #209512
- August 6th - CERT assigned 5 CWEs:
 - CWE-306: Missing Authentication For a Critical Function
 - CWE-321: Use of a Hard-Coded Cryptographic Key
 - CWE-798: Use of Hard-Coded Credentials
 - CWE-285: Improper Authorization
 - CWE-345: Insufficient Verification of Data Authenticity
- Ongoing - Creating CVEs.

Thank You

Questions?

idfoster@cs.ucsd.edu