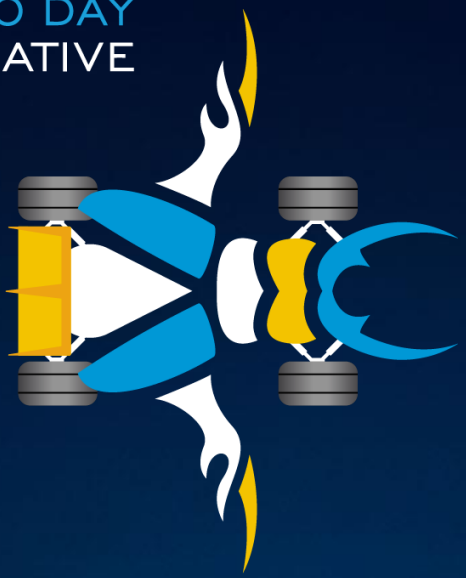
Pwn2Own Automotive

Vulnerabilities, Exploits, and Lessons Learned

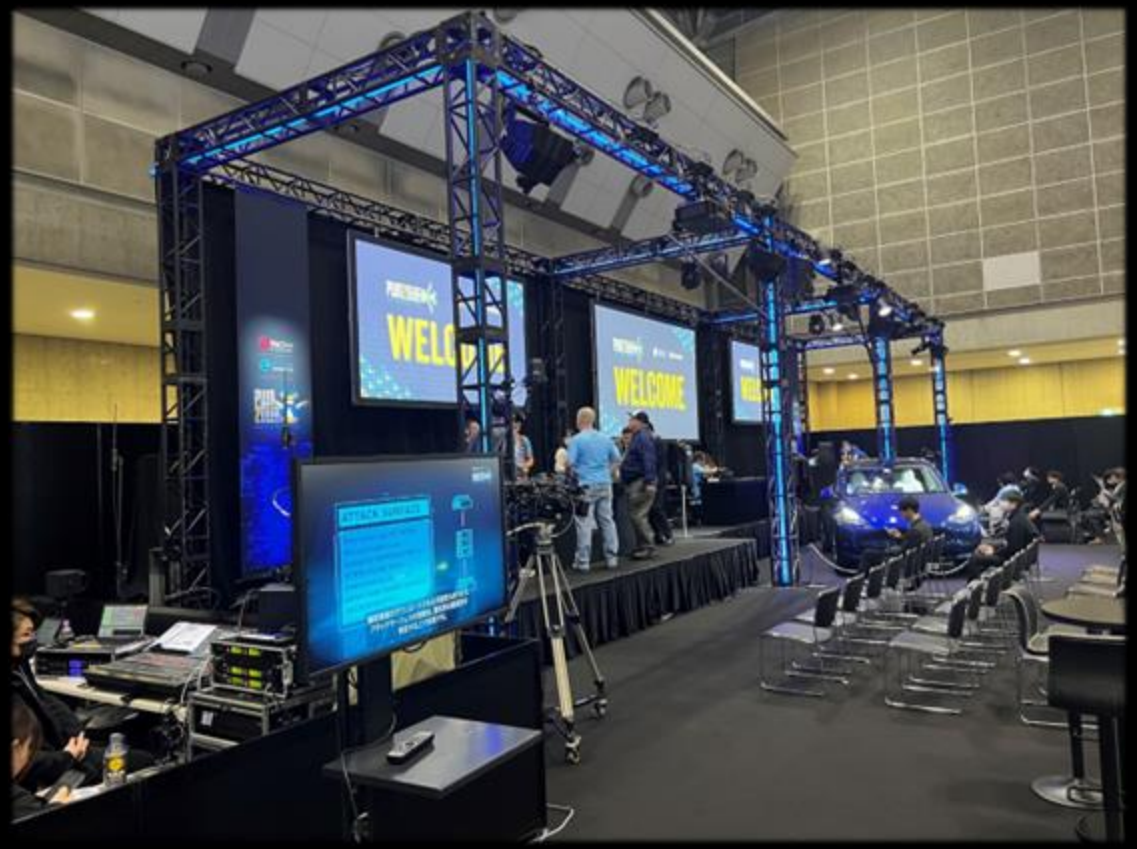
VicOne | ZERO DAY INITIATIVE

PWN2OWN

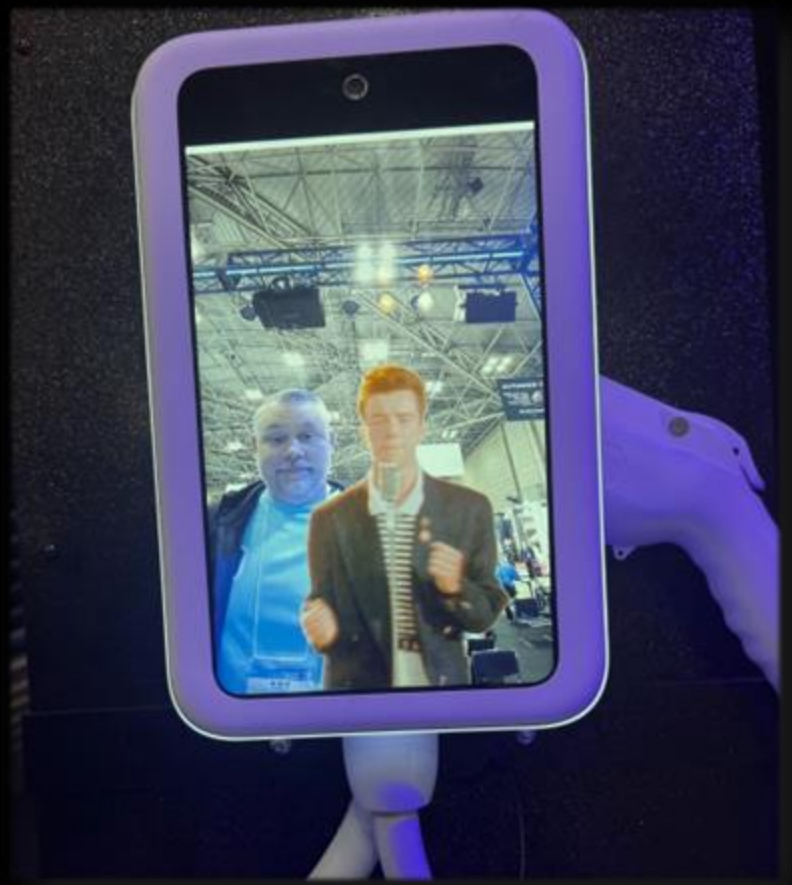


AUTOMOTIVE

Automotive World



Hacks



Results

2024	$1,323,750	49 0-days
2025	$886,250	49 0-days

Variety of Entries

Category	# Entries in 2024	# Entries in 2025
Electric Vehicle Chargers	30	23
In-Vehicle Infotainment (IVI)	16	25
Operating System	3	1
Tesla	2	1
Grand Total	**51**	**50**

Top 5 CWEs in 2024

CWE	Count of CWE
CWE-121: Stack-based Buffer Overflow	**10**
CWE-78: OS Command Injection	**4**
CWE-295: Improper Certificate Validation	**4**
CWE-120 - Buffer Copy without Checking Size of Input	**3**
CWE-416: Use After Free	**2**

Top 5 CWEs in 2025

CWE	Count of CWE
CWE-78: OS Command Injection	**10**
CWE-121: Stack-based Buffer Overflow	**8**
CWE-122: Heap-based Buffer Overflow	**4**
CWE-284: Improper Access Control	**3**
CWE-1328: Security Version Number Mutable to Older Versions	**3**

Electric Vehicle Charger Category

Consumer EV Charger Design

- Designs typically feature at least two major subsystems
  - Application processor subsystem (GUI / Network interfaces)
  - Power supply, metering & control circuitry
- Key components/systems observed
  - Display Modules
  - Memories – Flash / RAM
  - Ethernet, Wi-Fi, Bluetooth, LTE
  - TPM
  - CAN
  - NFC / RFID
  - Cameras
  - SAE J1772 (Standard EV charger plug)
  - Serial console / JTAG / debug ports
  - Power Relays

Hardware

- Many *devices* had serial interfaces available
- Several devices had JTAG/debugging interfaces enabled
- Many use off the shelf SoC/SoM for application processor
  - ESP32 module variants & Silicon Labs WGM Series modules
- Operating system vary from RTOS to Linux and Android
- Multiple firmwares are running in a typical EV charger

Mobile Applications

- Every charger discussed has an associated mobile app
- Communicate with the charger over Bluetooth
- Used for configuration
- Often are responsible for firmware updates
- Disassembling mobile apps provides useful information
  - Easy way to get started understanding the chargers

Communication

- Configuration occurs over Bluetooth
- Charger connects to local network
- Some have cellular network interfaces
  - SIM cards accessible
- Charger connects out to vendor cloud
- Cloud handles user authentication for charging

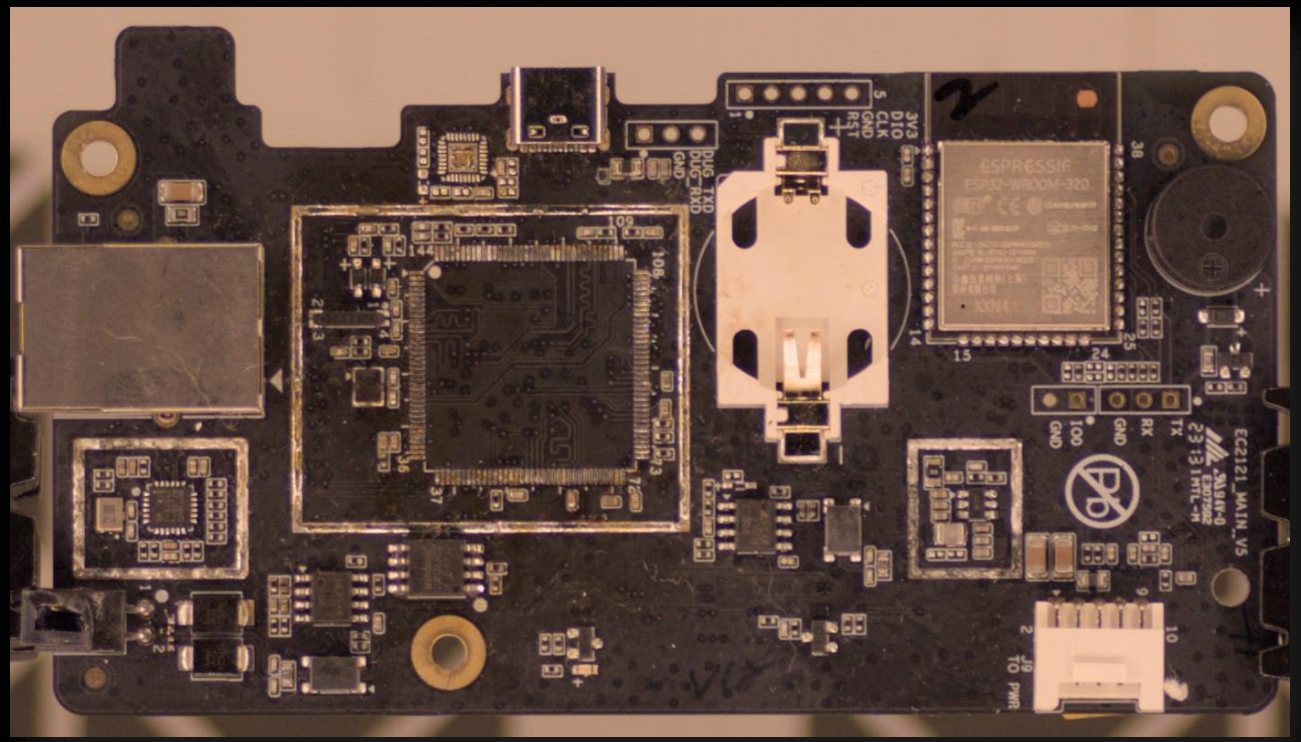
Attack Surface

- Mobile application & Bluetooth LE for configuration
- Wi-Fi & Ethernet connections
- Listening network services
  - OCPP, MQTT, HTTP/S, Telnet, SSH
- Connection to the cloud
- Firmware update process

Autel MaxiCharger

- Multi-PCB design
- CPU Board
  - GigaDevices GD32F407 (ARM Cortex M4)
  - ESP32-WROOM-32D (Xtensa)
- Metrology board
  - ST Micro STM32F407ZGT6 (ARM Cortex M4)
- Mobile Communication Board (LTE)
  - Quectel EC25-AFX

Autel MaxiCharger CPU Board

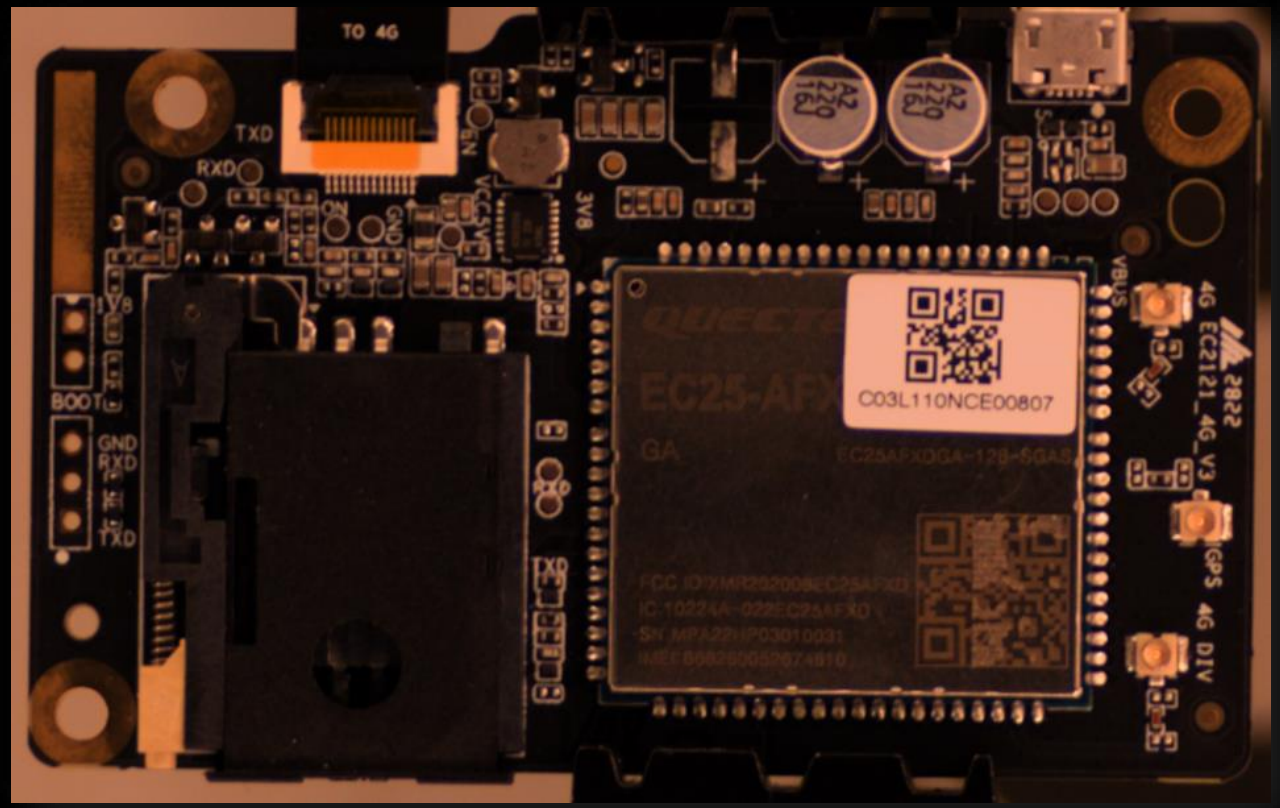


- GigaDevices GD32F407
- ESP-WROOM-32
- Barrot BR8051A01 Bluetooth
- Multiple serial ports emit boot logs for the main CPU and ESP

Autel MaxiCharger Metrology Board

- ST Micro STM32F407ZGT6
- Renergy RN830(B)
- Functional serial port emits boot logs

Autel MaxiCharger Radio Board



- Mobile communications board
- Quectel EC25-AFX
- Functional serial port emits boot logs
- Similar device is present in Tesla vehicles

Computest

Autel MaxiCharger

Zero Day Initiative

Pwn2Own Automotive 2024

TREND MICRO

June 21st, 2024

(Pwn2Own) Autel MaxiCharger AC Elite Business C50 WebSocket Base64 Decoding Stack-based Buffer Overflow Remote Code Execution Vulnerability

ZDI-24-853
ZDI-CAN-23230

CVE ID	CVE-2024-23967
CVSS SCORE	8.0, AV:A/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:H
AFFECTED VENDORS	Autel
AFFECTED PRODUCTS	MaxiCharger AC Elite Business C50
VULNERABILITY DETAILS	This vulnerability allows network-adjacent attackers to execute arbitrary code on affected installations of Autel MaxiCharger AC Elite Business C50 chargers. Although authentication is required to exploit this vulnerability, the existing authentication mechanism can be bypassed. The specific flaw exists within the handling of base64-encoded data within WebSocket messages. The issue results from the lack of proper validation of the length of user-supplied data prior to copying it to a fixed-length stack-based buffer. An attacker can leverage this vulnerability to execute code in the context of the device.
ADDITIONAL DETAILS	Fixed in US Firmware v1.35.00 and EU Firmware v1.50.00.
DISCLOSURE TIMELINE	2024-02-09 - Vulnerability reported to vendor 2024-06-21 - Coordinated public release of advisory 2024-08-15 - Advisory Updated
CREDIT	Daan Keuper, Thijs Alkemade and Khaled Nassar of Computest Sector 7

Autel MaxiCharger Security Conclusions

- Firmware suffers from several discovered stack buffer overflows in multiple features
- Lacks mitigations for stack-based buffer overflows
  - No stack cookies
  - No memory execution protection available
- Hardcoded device credentials

Tesla Category

Tesla Attack Surface

Any Tesla ECU	Vehicle (VEH) CAN Control	$200,000	20	Vehicle Prize	Yes
				Infotainment Root Persistence Add-on	No
				Autopilot Root Persistence Add-on	No
				CAN Bus Add-on	No
	Chassis (CH) CAN Control	$300,000	30	Vehicle Prize	Yes
				Infotainment Root Persistence Add-on	No
				Autopilot Root Persistence Add-on	No
				CAN Bus Add-on	No
	Party CAN Control	$400,000	40	Vehicle Prize	Yes
				Infotainment Root Persistence Add-on	No
				Autopilot Root Persistence Add-on	No
				CAN Bus Add-on	No

SYNACKTIV

TESLA
VCSEC RCE
2024.03.20

David BERARD
Vincent DEHORS
Thomas IMBERT

TREND MICRO

Vehicle Controller Secondary (VCSEC)

- Responsible for unlocking the car from a key fob or phone key, monitoring tire pressure sensors, and more
- Offers a command interface over Bluetooth Low Energy
- Connects and interacts with the different Tire Pressure Monitoring System (TPMS) of the car wheels using BLE
- Has a built-in feature that automatically detects a new set of wheel sensors
  - Feature called Auto Learning

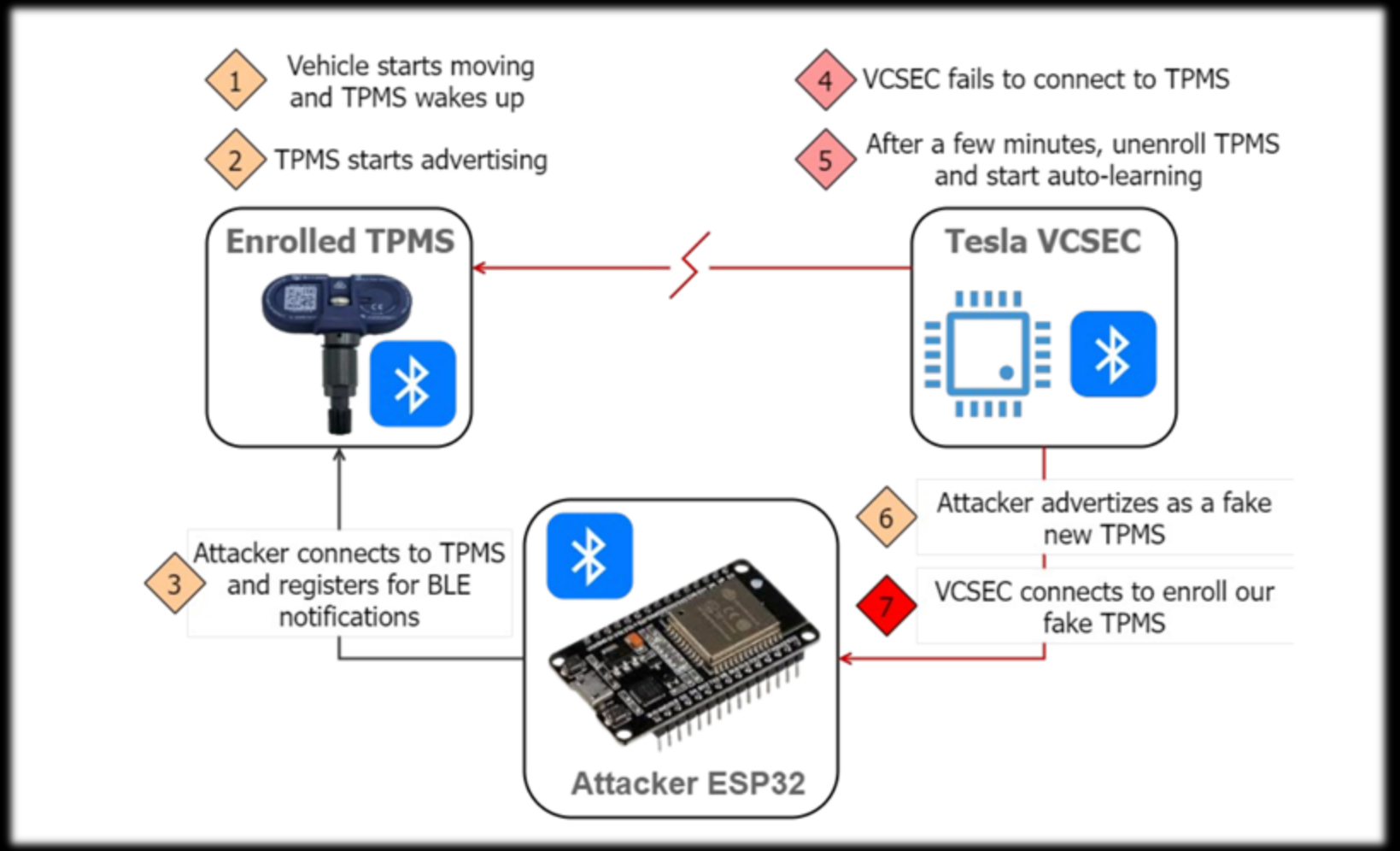
Tire Pressure Monitoring System (TPMS)

- TPMS sensors are integrated into the car tires
- Continuously monitor and report tire pressure to the central system
- If the tire pressure deviates from the optimal range, the user receives a warning via the infotainment user interface

TPMS Enrollment

- On first use of a new TPMS, VCSEC detects TPMS BLE advertising and enrolls the tire pressure sensor over BLE
- On a real vehicle, four tires TPMS are already enrolled thus VCSEC
  - Does not need to perform auto learning
  - VCSEC saves TPMS positions and Bluetooth MAC addresses on a persistent storage
- Vulnerability described requires that VCSEC start enrolling a fake TPMS sensor
- To unenroll one TPMS, the attack will prevent VCSEC from connecting to that TPMS sensor

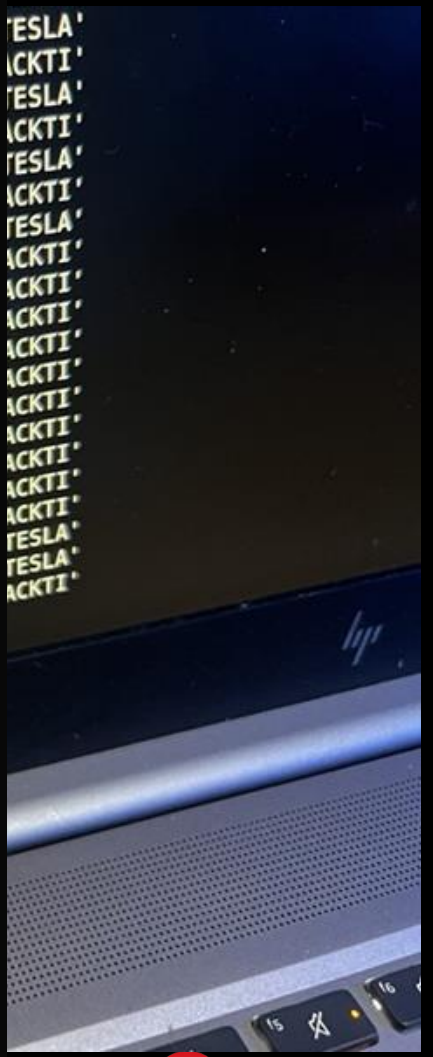
Registration Race Condition



Winner!

Zero Day Initiative @thezdi · Mar 20

Confirmed!!! The @Synacktiv team used a single integer overflow to exploit the #Tesla ECU with Vehicle (VEH) CAN BUS Control. The win $200,000, 20 Master of Pwn points, and a new Tesla Model 3 (their second!). Awesome work as always. #Pwn2Own #P2OVancouver



Observations

Hardware Observations

- Debug access easily available
  - Serial
  - JTAG, SWD, SBW, ADB debug
  - Special network services with complete device control
- Device designs don’t include secure chip variants, or don’t employ security features in the chips being used
- Devices that have support for TPM and TEE (TrustZone) appear unused
- Most devices don’t employ secure boot
- Chargers don’t employ hardware-backed firmware encryption
  - One instance of signed firmware was observed

Software Observations

- Parser implementations that contain buffer overflows
- Protocol handlers that allow for command injections
  - Use of system()/popen() calls that don’t sanitize input
- Use of hardcoded credentials
- Code lacking stack cookies
- Code lacking non-execute permissions on stack and heap memory (NX)
- Code lacking ASLR
- ASLR implementations that preserve relative memory layout

Security Strengths

- Some devices employ secure chip variants with higher security features
  - OTP / Secure boot / JTAG disable / Flash read protection / Flash encryption
  - TPM / TEE / TrustZone hardware on board
- One instance of HW Flash readback protection (but bypassed via EM-FI)
- OTA / Automatic updates / Signed updates
- Frequent use of secure network transports TLS/SSH w/cert validation
- Some devices had memory protections
  - Stack cookies, NX protections, ASLR

Overall Conclusions

- Mitigations are required for consumers that deploy these devices to their network
  - Network segmentation / VLANs
  - Additional network firewalling / Traffic filtering
- Many opportunities for improvement
  - Hardware design
  - Software security mitigations
  - Implement SDLC

Vendor Recommendations

- Employ basic security best practices in:
  - Authentication, input sanitization, use of available mitigations
- Perform static code analysis
- Perform fuzz testing
- Select chips that have memory protection features
  - Employ available security features of chips
  - Firmware encryption doesn’t fix exposed bugs and hinders research
- Use third party audits / bug bounties / engage researcher community / consultants