7085 SW Scholls Ferry Rd.
Beaverton Oregon 97008
503.781.5893
Jesse@JopelDesigns.com

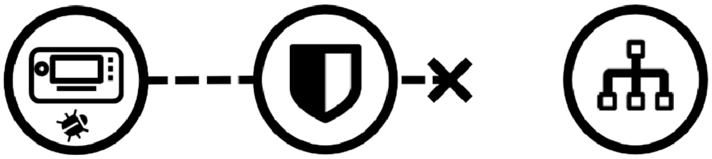
JOPEL DESIGNS JD
Project Portfolio

CANbus Firewall

Brief

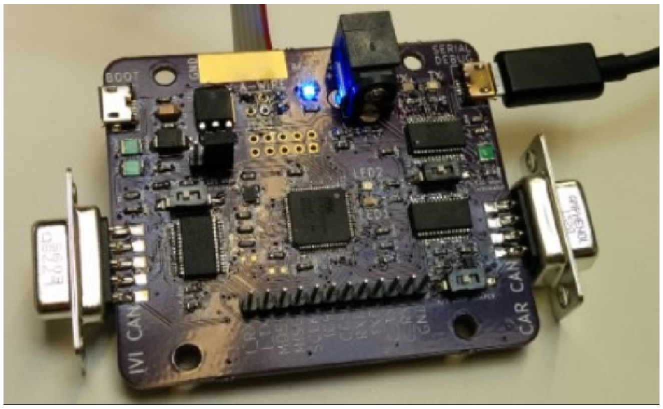
In 2015, security researchers exposed a weakness in vehicles operating the Uconnect "connected vehicle platform" operated by Chrysler, Dodge, Jeep, Ram and FIAT Brand vehicles. I was asked by my employer at the time, Jaguar Land Rover, to develop a method of protecting all vehicles from a similar attack. The project was funded by GENIVI Alliance, a "non-profit automotive industry alliance that develops standards for the centralized and connected vehicle."

Vulnerability



The Uconnect client installed in a vehicle contains a CAN firewall, meant to stop the infotainment system from issuing dangerous commands to the vehicle CAN networks by packet filtering. Unfortunately, it had a critical flaw. For the purpose of deploying over-the-air firmware updates to the vehicle ECU's, the infotainment system had the ability to reprogram the firewall. The researchers exploit removed filters safeguarding the CAN, allowing control of any vehicle sus-system, or removal of any message on the network by CRC faults injection.

Development



The hardware design is relatively simple; A 32-bit microprocessor sits between two CAN2.0B Controllers with up to 1 Mb/s Operation. Defending the Uconnect vulnerability was accomplished by separating the rules and firmware and adding a physical interlock. After the initial bootloader is installed, firmware can only be updated by USB, cementing behaviour. Sensing a jumper in the CAN port means only with a service tool and physical access can someone update the rule set.

Auxiliary Role

The software developer on the project had never written code for an embedded platform. Algorithm efficiency and RAM usage weren't ever a concern. They especially lacked any experience with offloading work to peripheral devices or structuring software for an interrupt driven architecture. I taught the developer what I knew, and helped structure the program, basically deciding the interrupt driven nature of the software.

https://github.com/PDXostc/canfw
https://github.com/PDXostc/can_firewall_software
https://github.com/PDXostc/can_firewall_hardware