RSAC Sandbox

RSAConference2020

HUMAN ELEMENT

SESSION ID: **SBX1-R3**

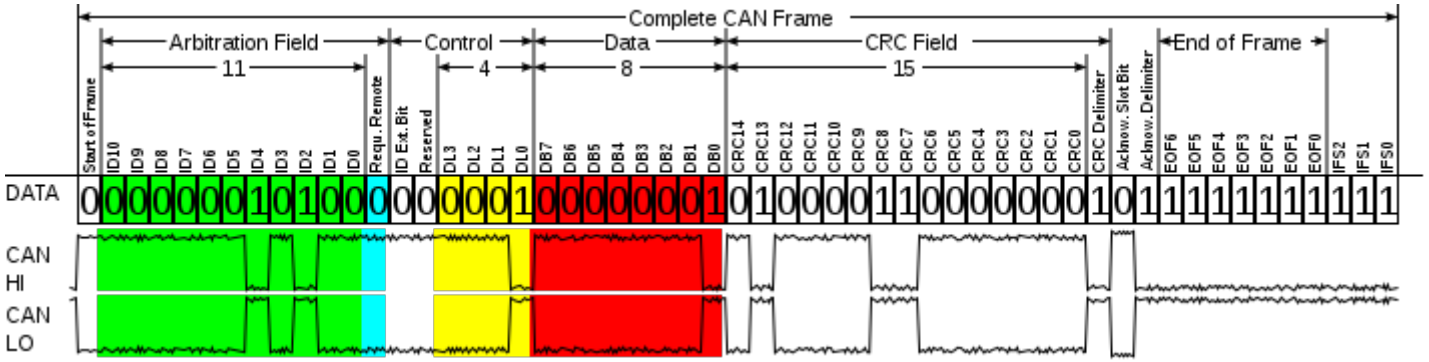
On the CANT Bus, No One CAN Hear You Scream

Tim Brom

Managing Senior Security Researcher, Embedded Systems

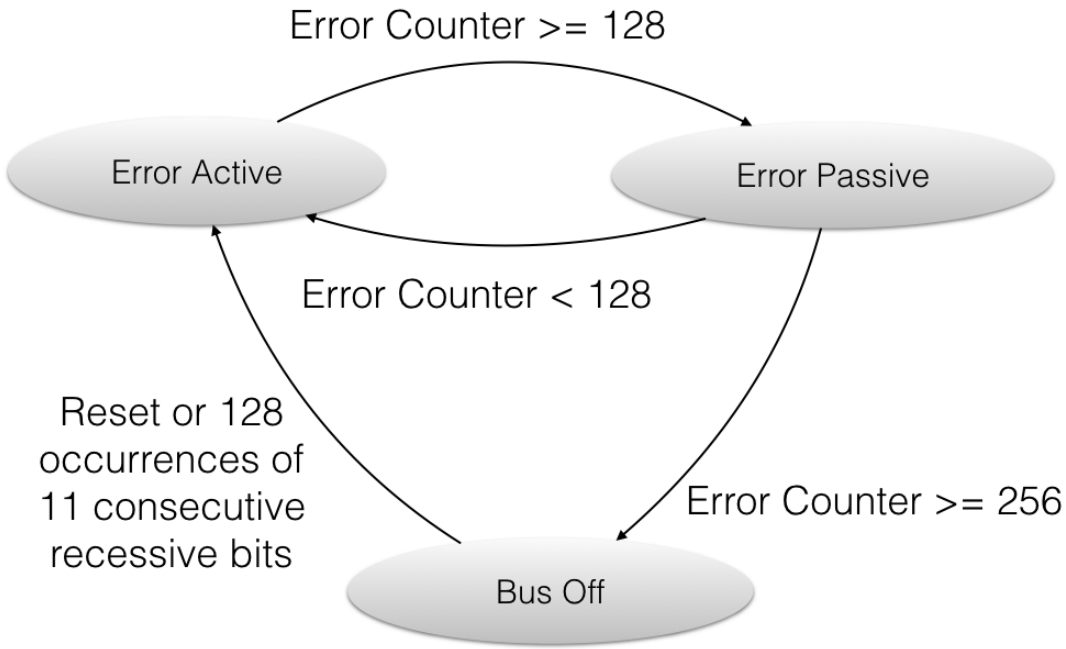
CAN In Two Slides

- Electrically reliable communications bus developed in the ‘80s
- Differential signal, “Dominant (0)” and “Recessive (1)” states
- Bus arbitration decided by Arbitration ID
- If **Any** node is asserting a Dominant bit, the state of the bus will be in the Dominant state (or will it…)



CAN Error Handling

- Two types of error frames
  - Active Error (6-12 Dominant bits)
  - Passive Error (6 Recessive bits)
- Five Detectable Errors
  - Bit, Stuff, CRC, Form and Acknowledgement
- Two Error Counters
  - Transmit and Receive
- Three Error States
  - Active, Passive and Bus Off



CAN Philosophy

RSAC Sandbox

> "... it is in the nature of the MAC sublayer that there is no freedom for modifications."

CAN is based on the premise that the nodes on the bus will behave

Wouldn't it be nice if we had a tool that made this assumption easy to test...

Inspiration and Need

RSAC Sandbox

- Existing research tools use a CAN perpheral
- Inspired by conversations, need
- ICS-ALERT-17-209-01
- Tired of hearing "That CANT Happen!"

Questions Raised

RSAC Sandbox

ICS-ALERT-17-209-01 raised questions

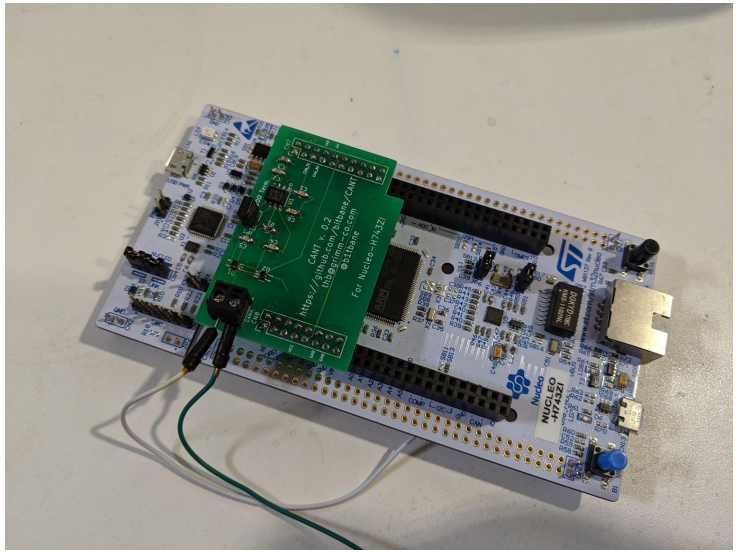
Released with limited proof-of-concept

1. Can attacks be expanded?
2. Will the attacks require specialized hardware?
3. Can the attacks be launched from an ECU?
4. Will the attacks be effective against IDS/IPS systems?

But that CANT Happen!

RSAC Sandbox

- CANT is purpose built to selectively target individual ECUs by abusing the spec
- Based on an ST Micro Nucleo-H743ZI dev board ($20)
- 400 MHz ARM Cortex-M7
- 1MB Ram
- 2MB flash

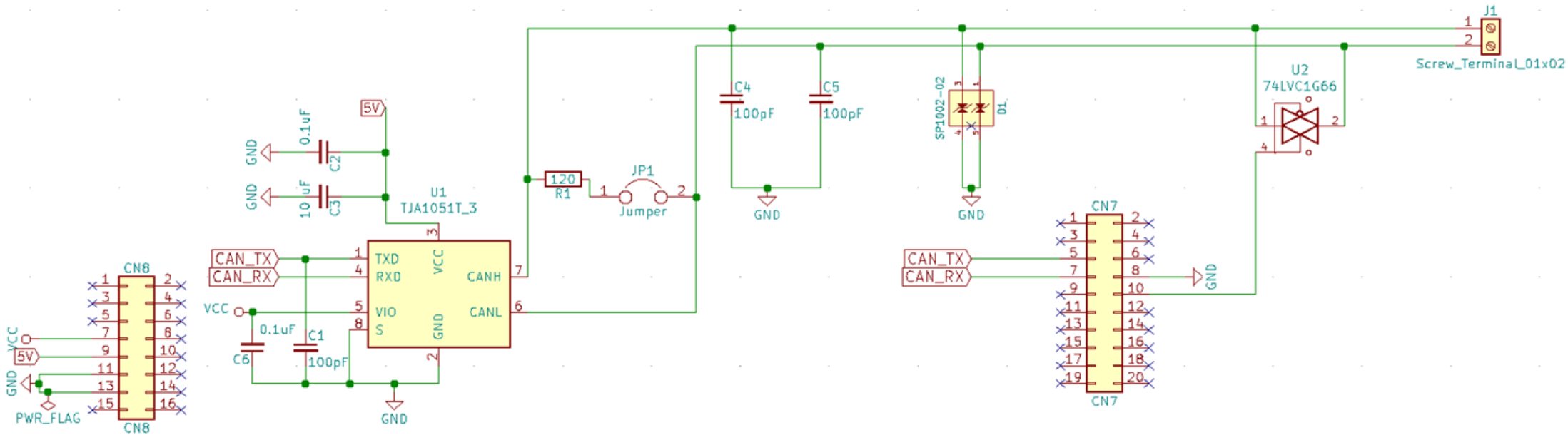


Constraints

- Wanted to be able to port this this to other platforms easily - no reliance on specialized hardware (e.g, NXP FlexIO, BeagleBone Black, FPGA)
- This makes porting to an ECU easier
- CANT relies on 2 timers, re-mapping CAN pins to GPIO, and an external interrupt on the CAN RX pin

CANT Add-on Board Schematic

RSAC Sandbox



But that CANT Happen!

Currently supported attacks

1. DOS all messages (arbid 0x0)
2. Replace data of selected messages
3. Transmit overload frames
4. Bus Short
5. NACK Attack

Demonstration

RSAC Sandbox

GRIMM

CANT Capabilities

- Complete control over signaling
- Follow the CAN spec as it suits us
- Stop when it doesn't
- ECU giving you trouble? Be a shame if something... happened to it
- Suppression of error frames
- More difficult to detect than packet injection

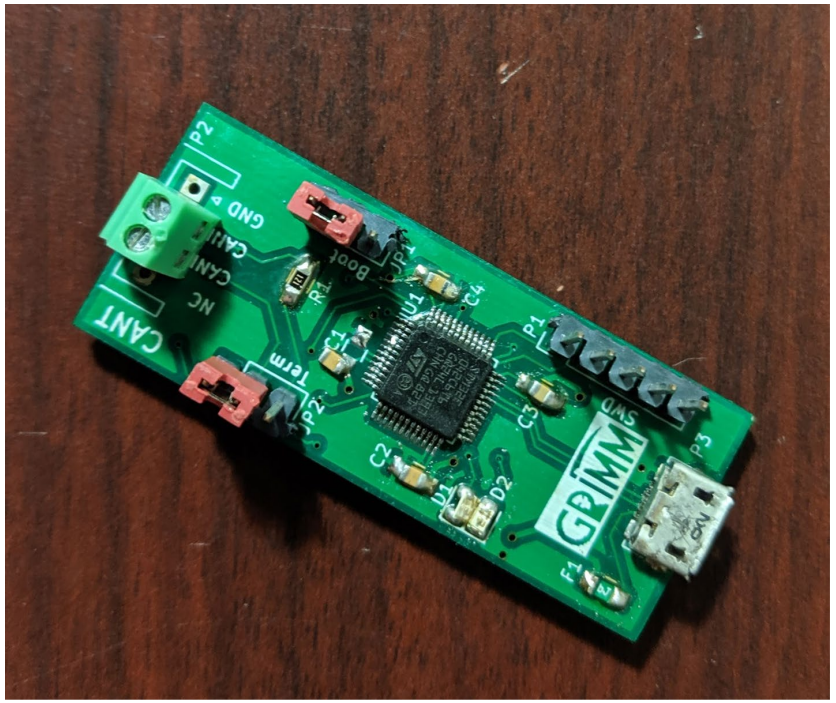
Current Status

RSAC Sandbox

- First two questions answered
  - Expanded attacks are possible
  - Without specialized hardware
- Last two questions
  - Launched from an ECU?
  - Effective against IDS/IPS?

Other CANT Attempts

- Lower-powered processor?
  - CANTable - based on CANable CAN tool
  - 48MHz ARM Cortex-M0
- Works up to 100KBPs



Limitations and Mitigations

- Physical access?
- Increased CAN bus load
  - But this isn't necessarily a sign of a problem
- Power fingerprinting could detect this attack
  - But then what?
- Network segmentation
- Encryption
- Other network technologies

Further Reading

- [1] CAN Specification Version 2.0 https://www.kvaser.com/software/7330130980914/V1/can2spec.pdf
- [2] "A Stealth, Selective, Link-layer Denial-of-Service Attack Against Automotive Networks" https://www.politesi.polimi.it/bitstream/10589/126393/1/tesi_palanca.pdf

Questions?

RSAC Sandbox

https://github.com/bitbane/CANT

Innovation. Break the rules.
Passion. Believe in what could be.
Humility. Only the mission matters.
Capacity. Learn. Share. Ask.
Agility. Change is constant.

GRIMM