In Vehicle Networks

CyberTruck Challenge™ 2025

Ben Gardiner, NMFTA

NMFTA™
National Motor Freight Traffic Association, Inc.

Agenda

120 mins

- Overview of Various Truck Vehicle Networks
- Publicly Disclosed Vehicle Network Cyber-physical Attacks
- Vehicle Network Segments and ‘Gateways’
- Replay Attacks
- Enumerating ‘Controller Applications’
  - ( the hands-on activities are here)
- Tips for Assessment Days

NMFTA™
National Motor Freight Traffic Association, Inc.

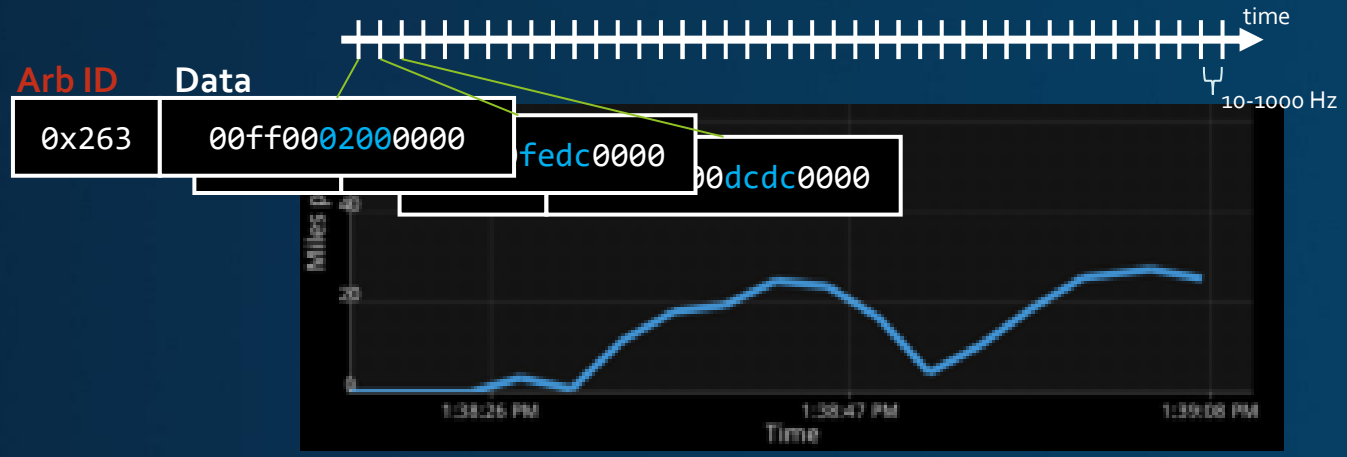
Learning Objectives of this Course

- Knowledge of different networks found in trucks
  - J1708
  - PLC4TRUCKS / J2497
  - J1939
  - Automotive Ethernet
- Knowledge of the functions for proprietary messages.
- Knowledge of public attacks causing de-rates
- Knowledge of the purpose of the vehicle network gateway and ideal filtering rules
- How to passively enumerate controller applications from logs of network traffic.
- How to actively enumerate controller applications.
- How to bisect network replays to search for desired traffic

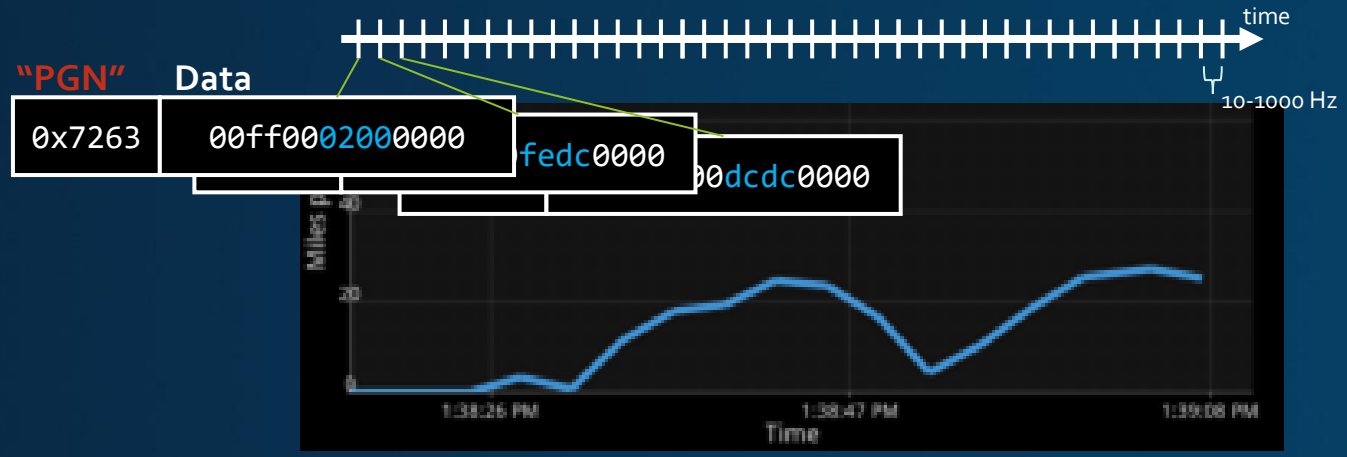
Truck Vehicle Networks: J1939

J1939 in relation to CAN in Passenger Cars

- Both: encoding time-varying signals into bitfield locations and diagnostics
- Passenger cars:
  1. *proprietary* Arbitration ID,
  2. *proprietary* bitfield locations,
  3. standard diagnostics (mostly)
- J1939:
  1. standard PGNs (mostly),
  2. standard SPNs (mostly),
  3. *proprietary* diagnostics



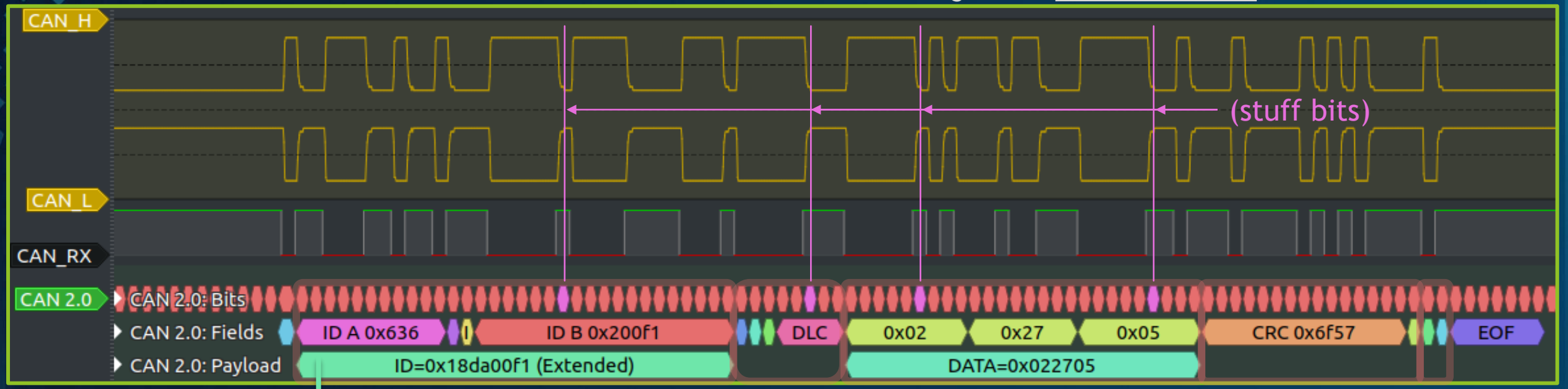
Ford OpenXC Platform 2017



Ford OpenXC Platform 2017

J1939 Specifics: CAN Frames

sigrok with kentindell/canhack can2 decoder:



29bit Arbitration ID

Control Field

Data Field (8byte max)

Error Checking

ACK

	ID A											ID B															
Arb ID bits (host):	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
for unicast:						PGN>>8								Dest Addr															
for bcast:						PGN																							
for all:	priority			res	page																	Source Addr							

J1939 Features

- Both unicast (PGNs < 0xF000) & broadcast (>= 0xF000)
- Transport fragmentation and reassembly (PGNs 0xEC00 and 0xEB00)
- Address claiming (0xEE00)
- Request of PGNs (0xEA00)
- Proprietary messages:
  - destination-specific (propA 0xEF00, propA2 0x1EF00) and
  - broadcast (propB0 0xFF00-0xFFFF, propB1 0x1FF00-0x1FFFF)
- Dump, reconfigure, reflash (👈 'the fun stuff') is all protected by a challenge-response system called *Seed-Key Exchange*
  - over ISO 15765-2 aka *ISO-TP* for UDS (0xDA00)

- For more details see Hannah Silva's CyberTruck Challenge™ 2021 Training www.cybertruckchallenge.org/wp-content/uploads/2021/08/Truck-Networks-Print.pdf

Prop v Interop

- CAN (and J1939) was introduced for distributed control systems
- Today:
  - Most control loops use proprietary space messages
  - J1939 messages remain for interoperability

Finding J1939 (1/6)

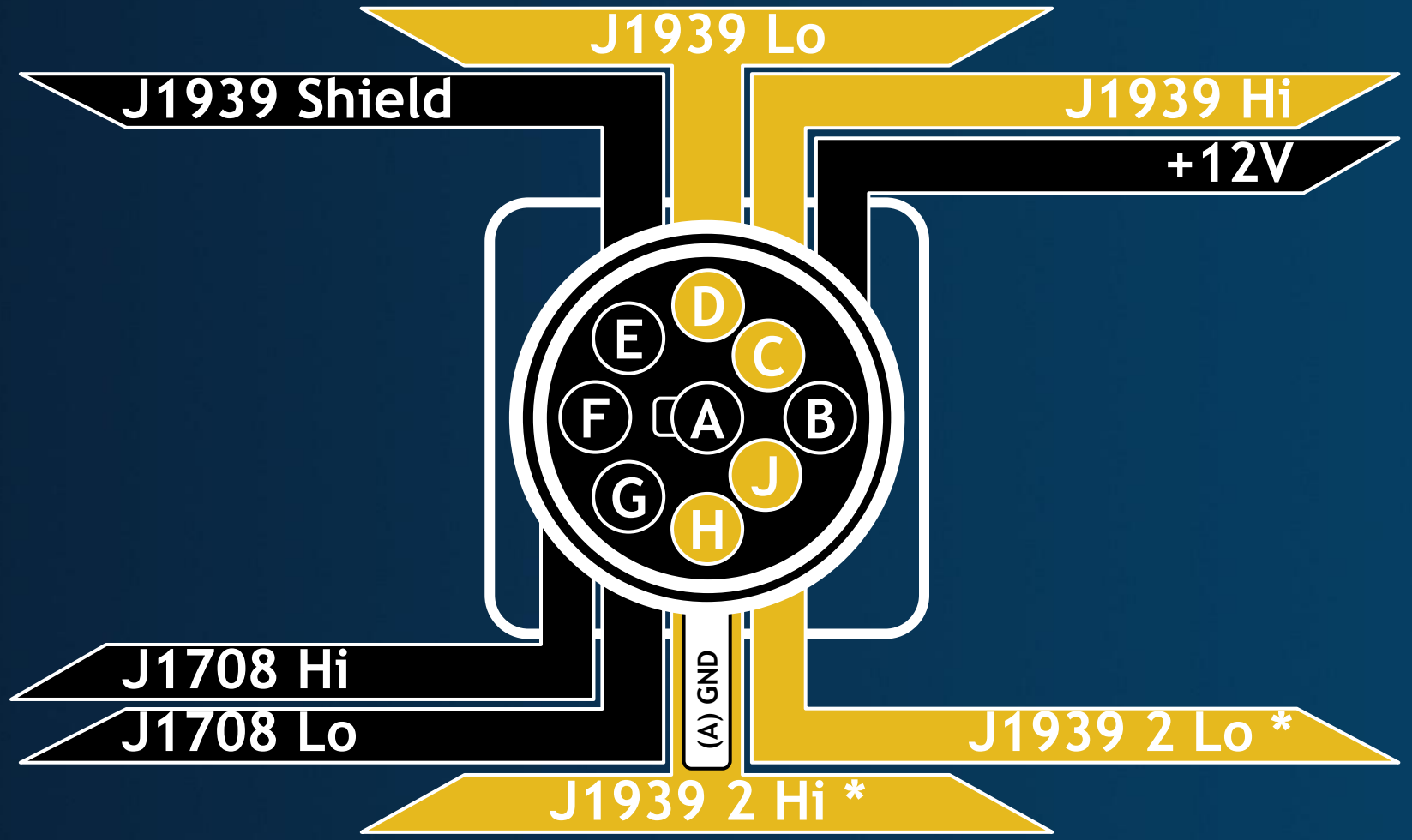
- In-cab or On-Board Diagnostics J1939 connector
- Black or Green
- Some OEMs use the passcar OBD-II connector. 👇

CC BY-SA Florian-schäffer

Dr. Jeremy Daily

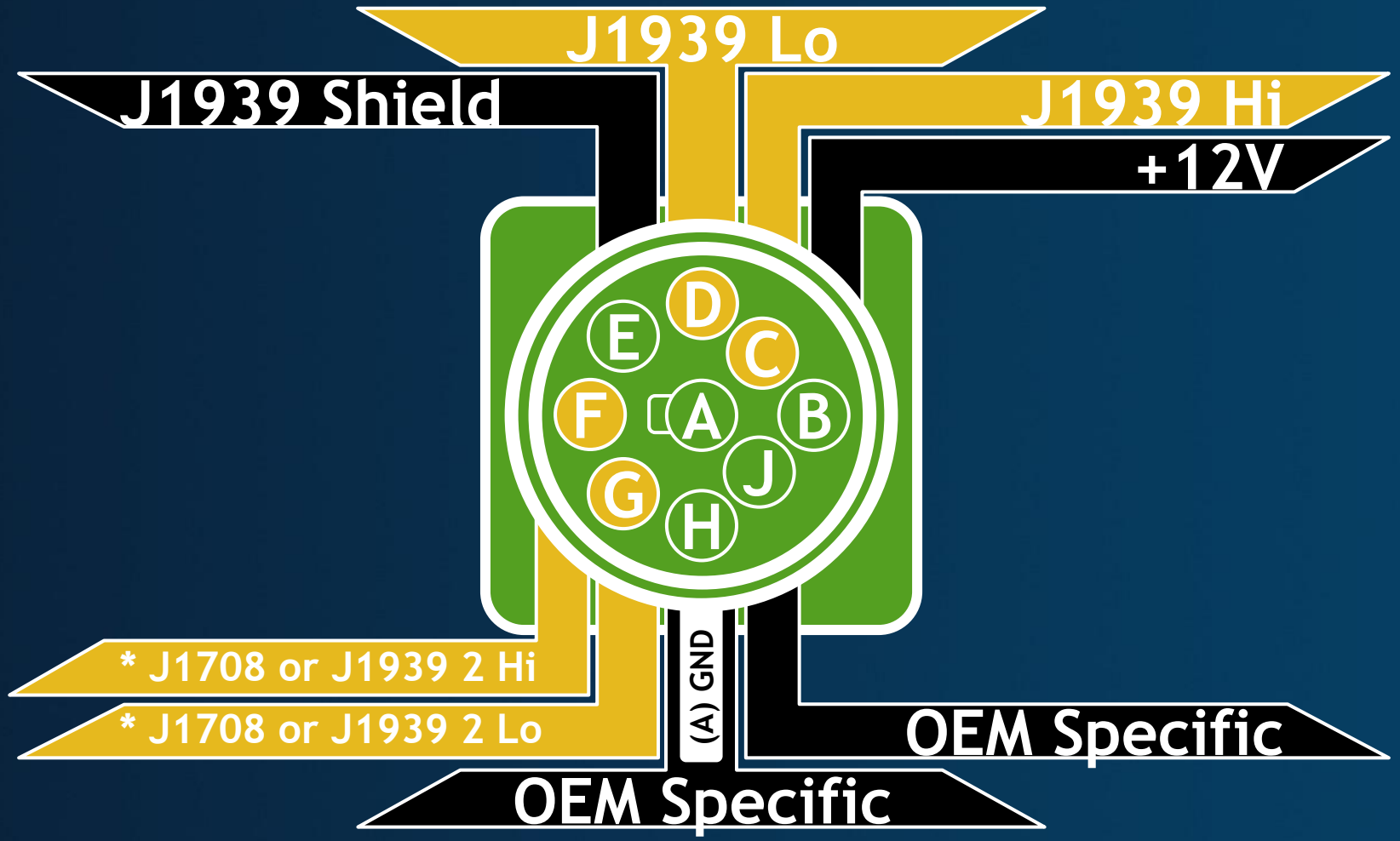
Finding J1939 (2/6)

- On black socket
- (* means optional)



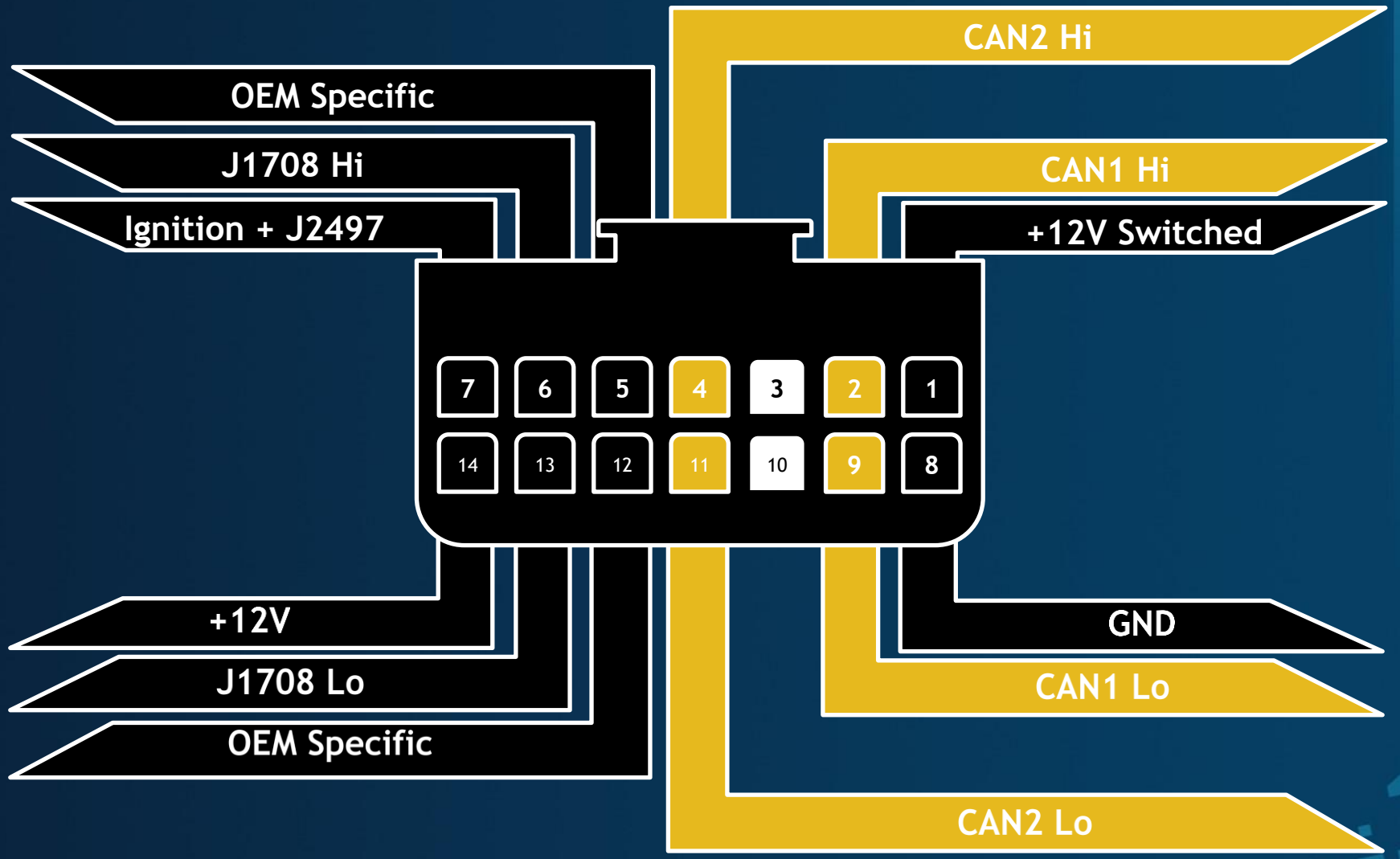
Finding J1939 (3/6)

- On green socket



Finding J1939 (4/6)

- On the RP1226 (Aftermarket/ Telematics) Connector
- Found behind dash or in berth



Finding J1939 (5/6)

- Other wires in the truck too…
  - ~6 separate CAN segments 👉

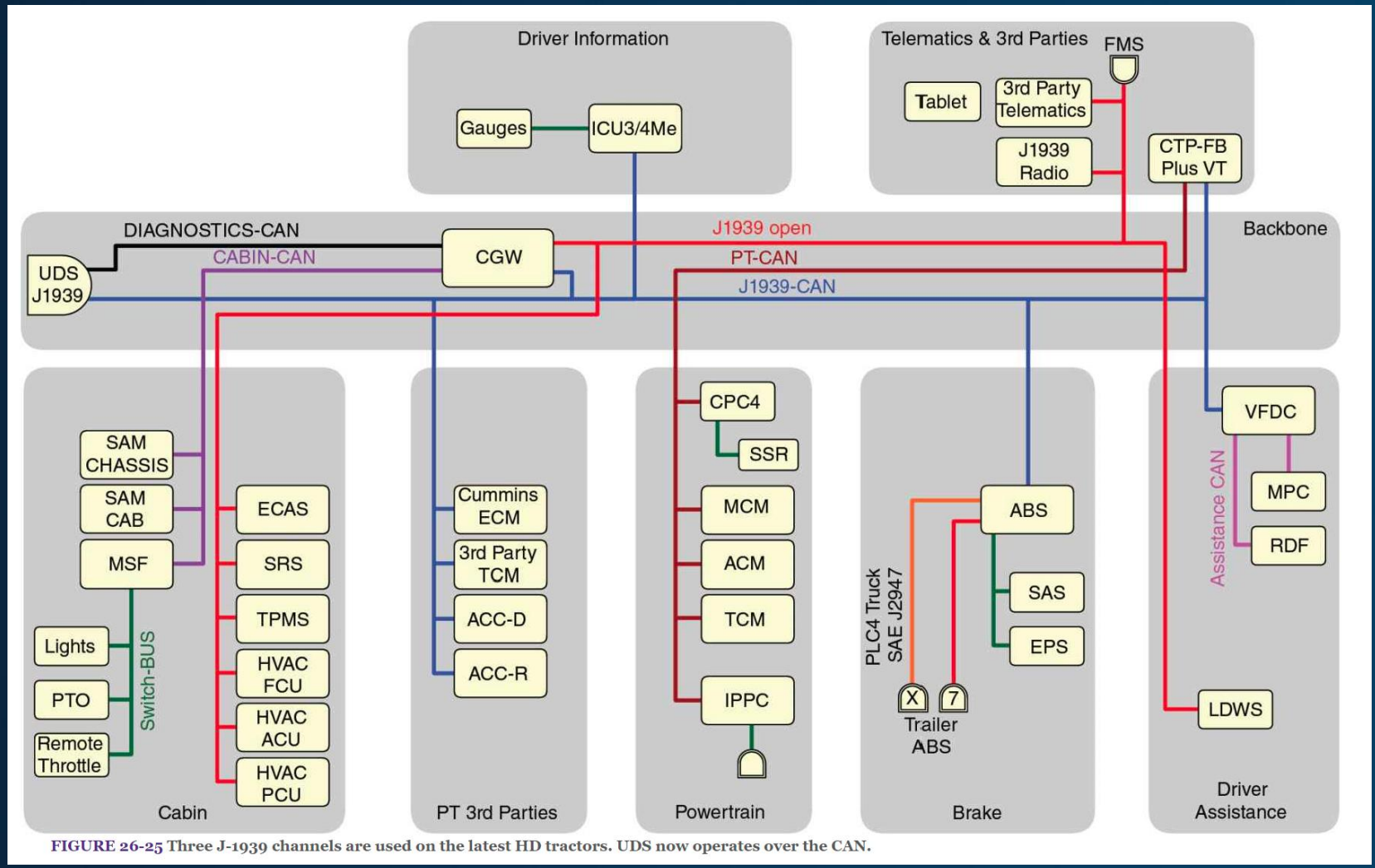


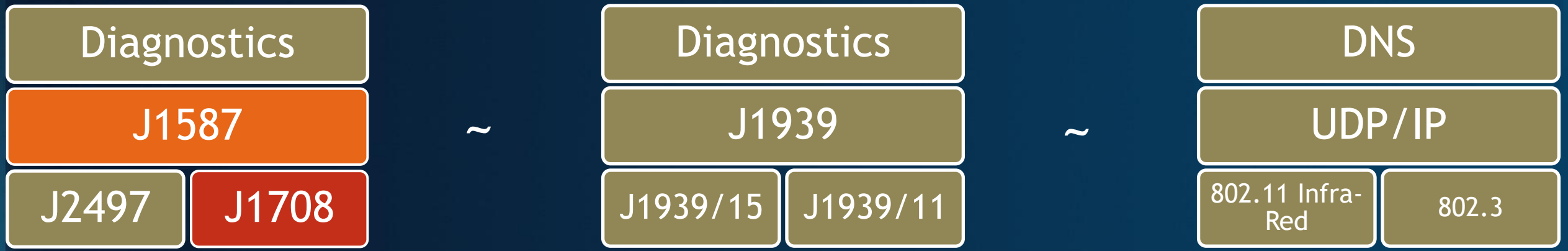
FIGURE 26-25 Three J-1939 channels are used on the latest HD tractors. UDS now operates over the CAN.

[Duffy, Owen C., and Gus Wright. *Fundamentals of Medium/Heavy Duty Commercial Vehicle Systems*: 2014 NATEF Edition. Jones & Bartlett Publishers, 2015]

Truck Vehicle Networks: J1708/J1587

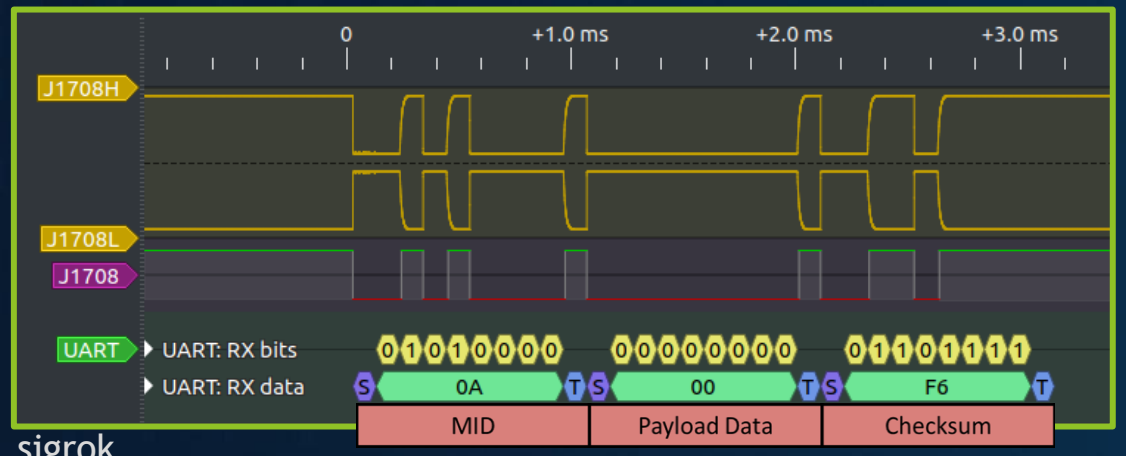
J1708/J1587 Specifics (1/3)

- Predates J1939 by many years. Sometimes still found in the tractor. Always still found in the Trailer as J2497 (more on that later).
- J1708/J1587 by analogy:



J1708/J1587 Specifics (2/3)

- Has similar bus arbitration to CAN: lowest first byte wins.
- 9600bps / 8N1
- Very much like an RS-485 bus at physical layer
- Has RT constraints for framing and bus arbitration
- The first byte is like a source address: the MID



sigrok

- Some noteworthy MIDs from the specs (J1708/J1587/J2497)
  - 111 is used for factory test
  - 128-255 are defined by J1587
  - 64 & 172 are off-board diagnostics
  - 48 & 153 are on-board diagnostics
  - 182 is off-board programming
  - 163 is 'vehicle security'
  - 207 is for drivetrain bridge
  - 217 & 218 tractor & trailer bridges
  - 87 is for J2497 active ABS event
  - 125 is for J2497 identification
  - 10 (0x0a) and 11 (0x0b) are J2497 lamp on/off

J1708/J1587 Specifics (3/3)

- signals are identified by a PID byte prepended to the signal
- can be multiple PIDs in one J1587 frame

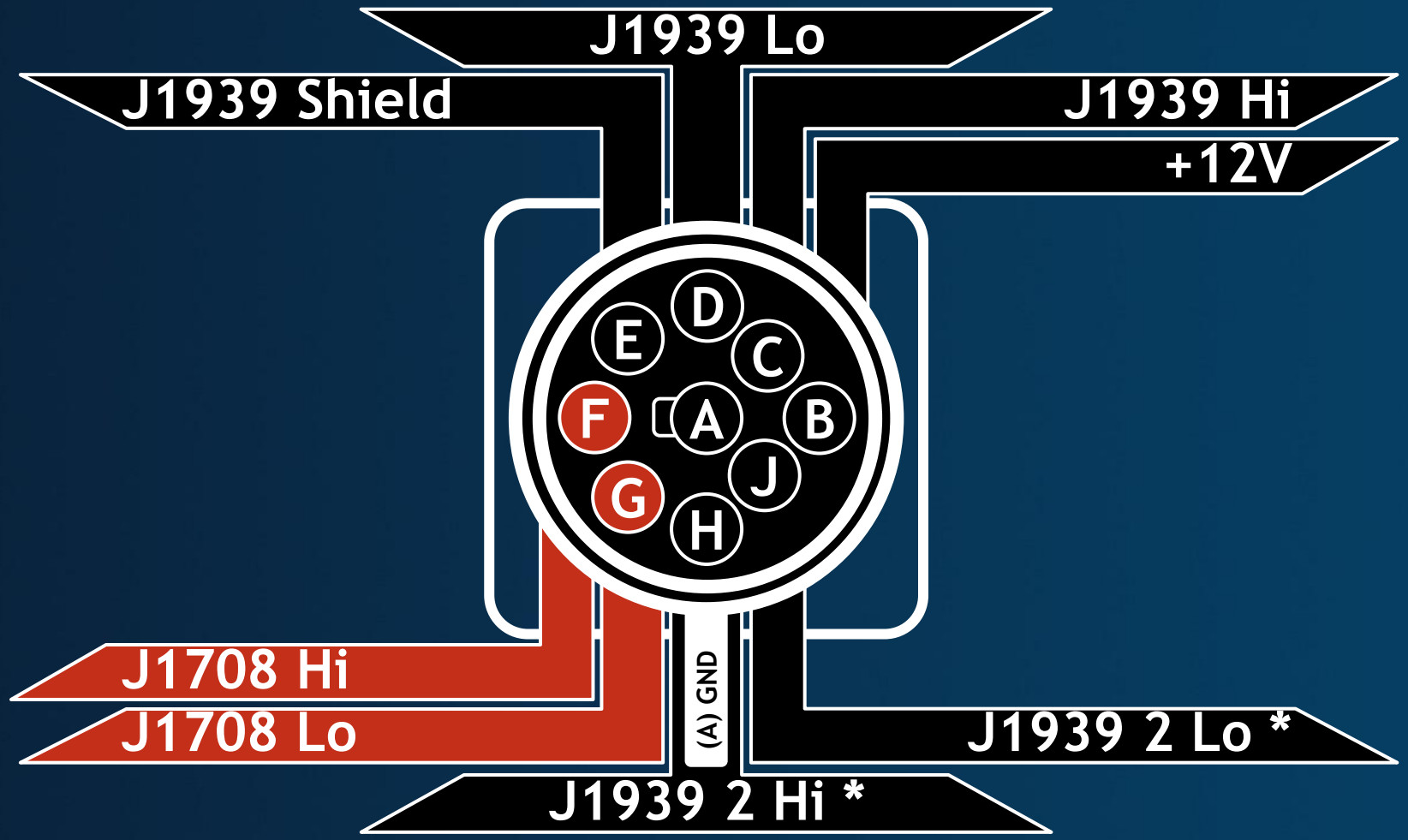
- PIDs range: 0-1021
  - PIDs > 255 use multi-byte PID extension
- Decoding of PIDs is done by reference to the J1587 specification.
  - There are tools that can convert the SAE PDF into a database and do decode.

J1587 Features

- Mostly broadcast (some unicast)
- Requests for data:
  - PID 0 (broadcast) / PID 128: Component specific (unicast)
- Has fragmentation and reassembly (Frames *should be* less than 21 bytes if the vehicle is in motion)
  - PID 192: ‘multisection’ parameter (broadcast)
  - PID 197 and 198: transport protocol (unicast)
- ‘Standardized Free-Format Data’ requests on transport protocol
  - e.g. ‘Programmable Params’ / ‘Calibration’, ‘Executable Code’
- Proprietary messages: ‘Data Link Escape’ (unicast) 👈 ‘the fun stuff’
  - PID 254 and 510
  - e.g. “AC FE 80 F0 17”
    is from MID 0xAC to ‘MID’ 0x80

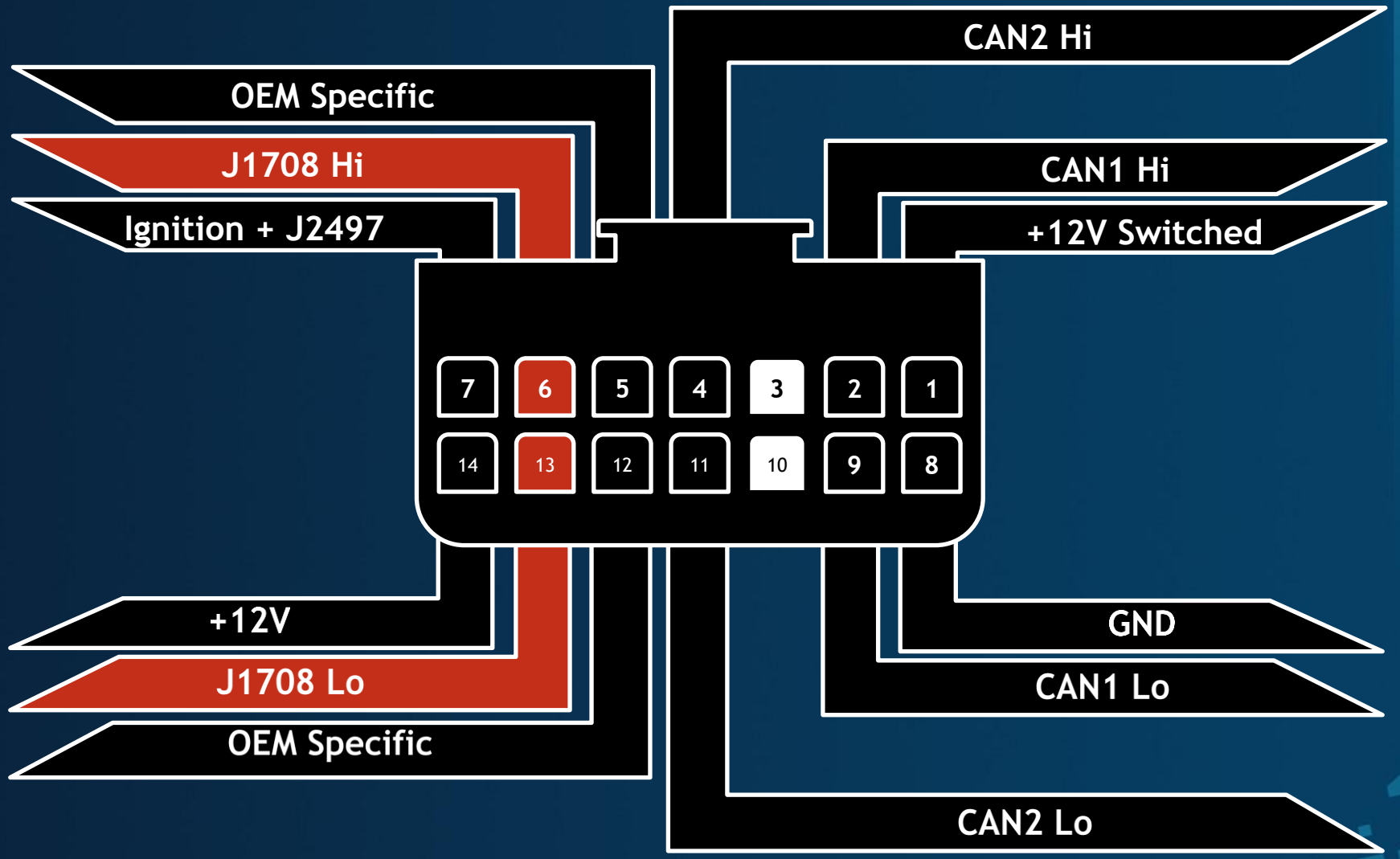
Finding J1708/J1587 (1/3)

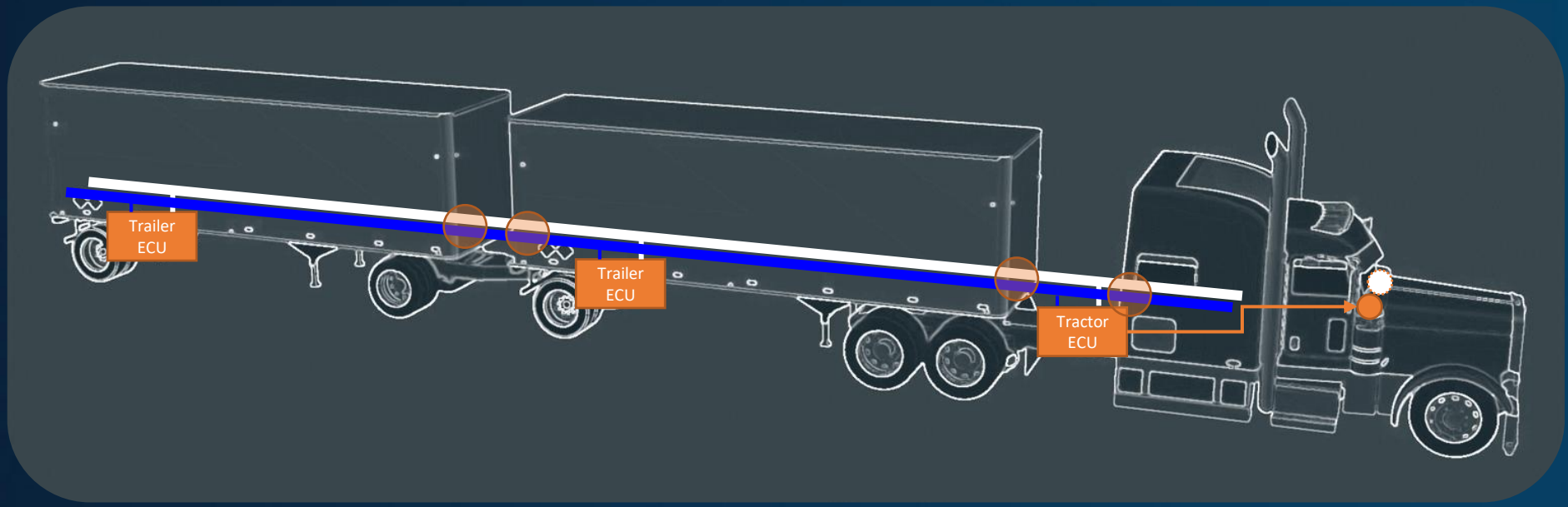
- Present on black socket
- Optional on green socket



Finding J1708/J1587 (2/3)

- On the RP1226 (Aftermarket/ Telematics) Connector

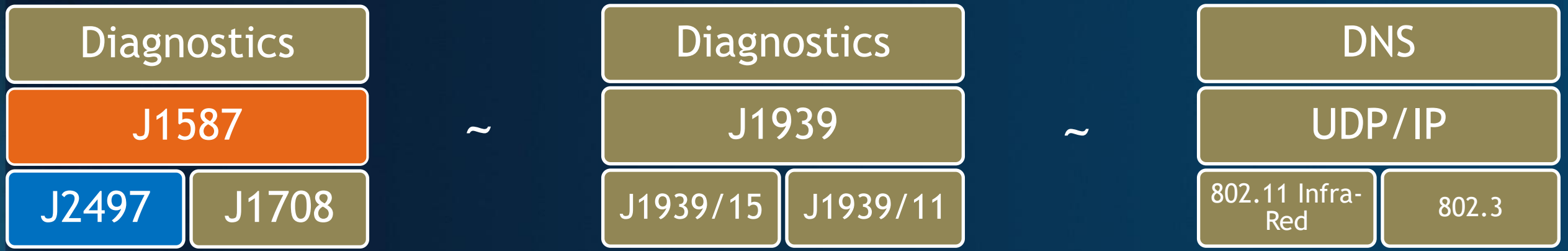




Truck (and Trailer) Vehicle Networks: J2497

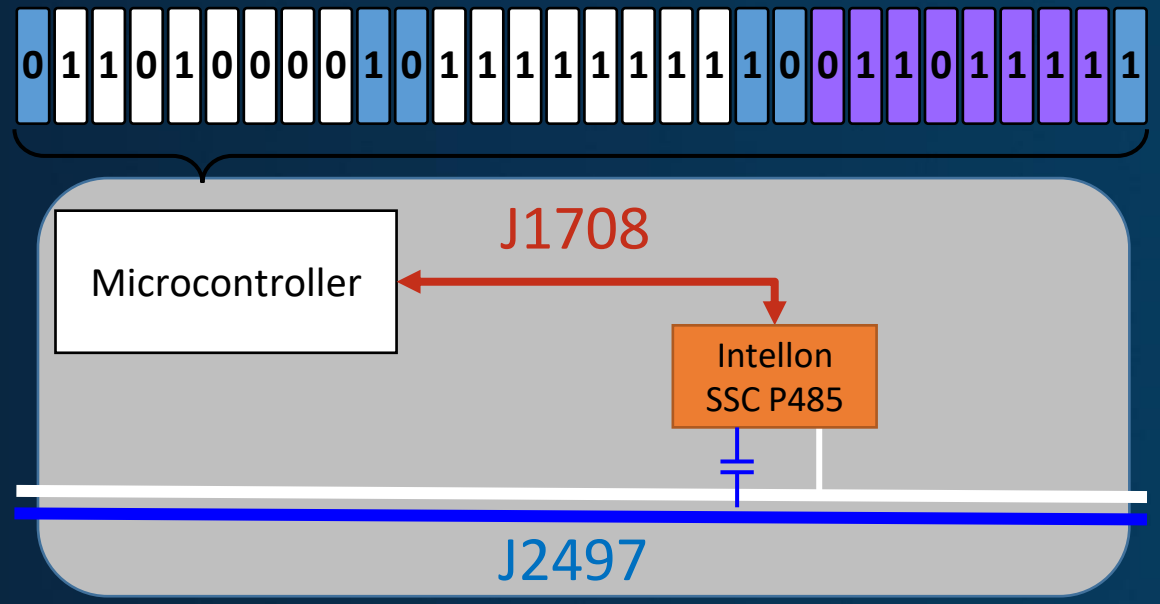
J2497 Specifics (1/2)

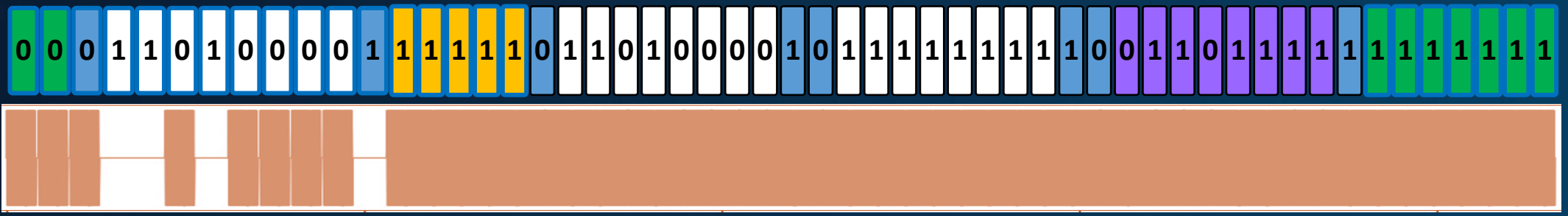
- Roughly speaking, it is "J1708 over trailer power lines"
- a.k.a. *PLC4TRUCKS*
- Again by analogy:



J2497 Specifics (2/2)

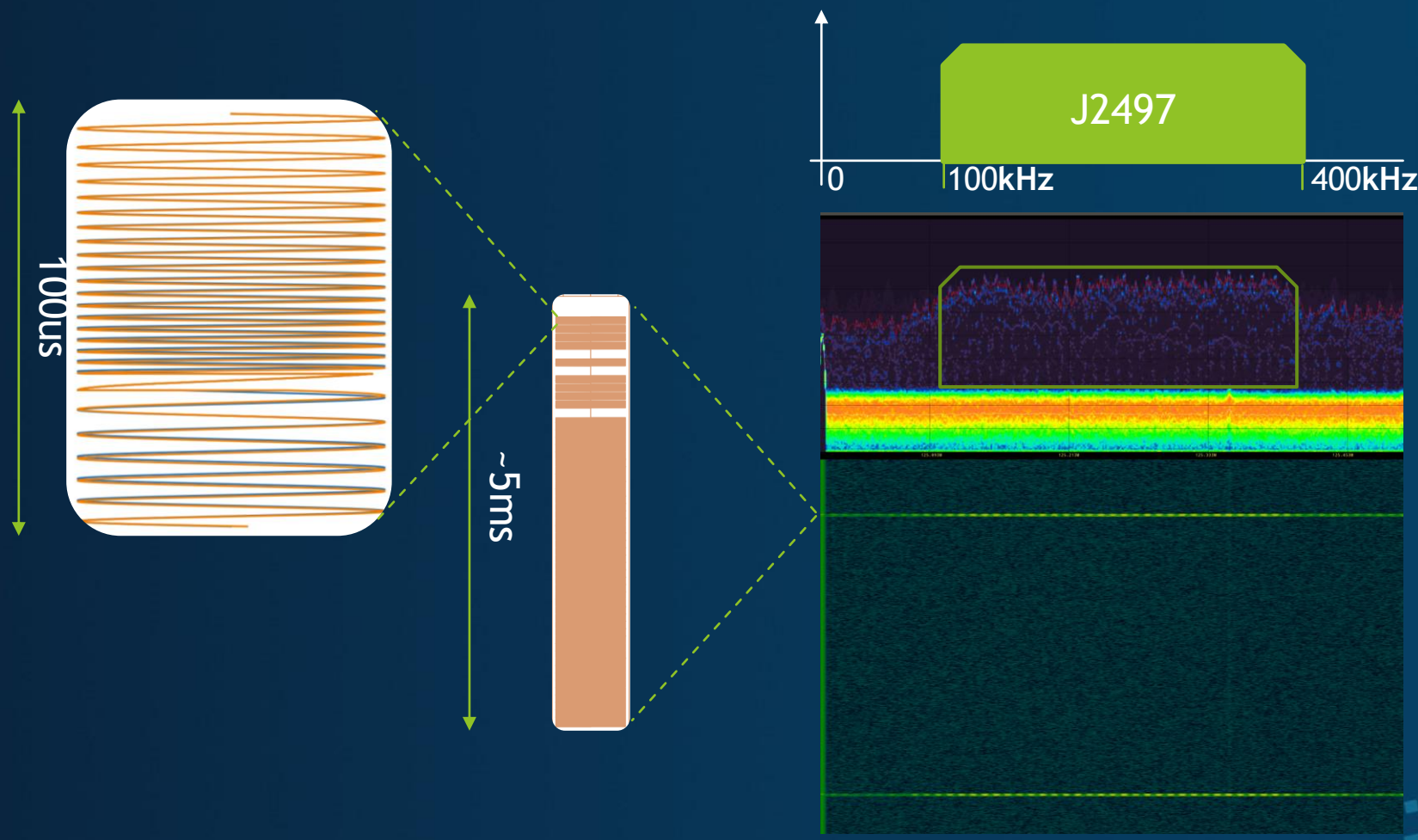
- J1708 ↔ J2497
  - Implemented almost exclusively by the Intellon SSC P485 chip





J2497 Specifics (3/4)

- The chirps are 100us in duration, between 2.5 and 7V peak-peak
- The chirps sweep from 203KHz through 400KHz (63us) then to 100KHz (4us) and back to 203KHz (33us) to finish



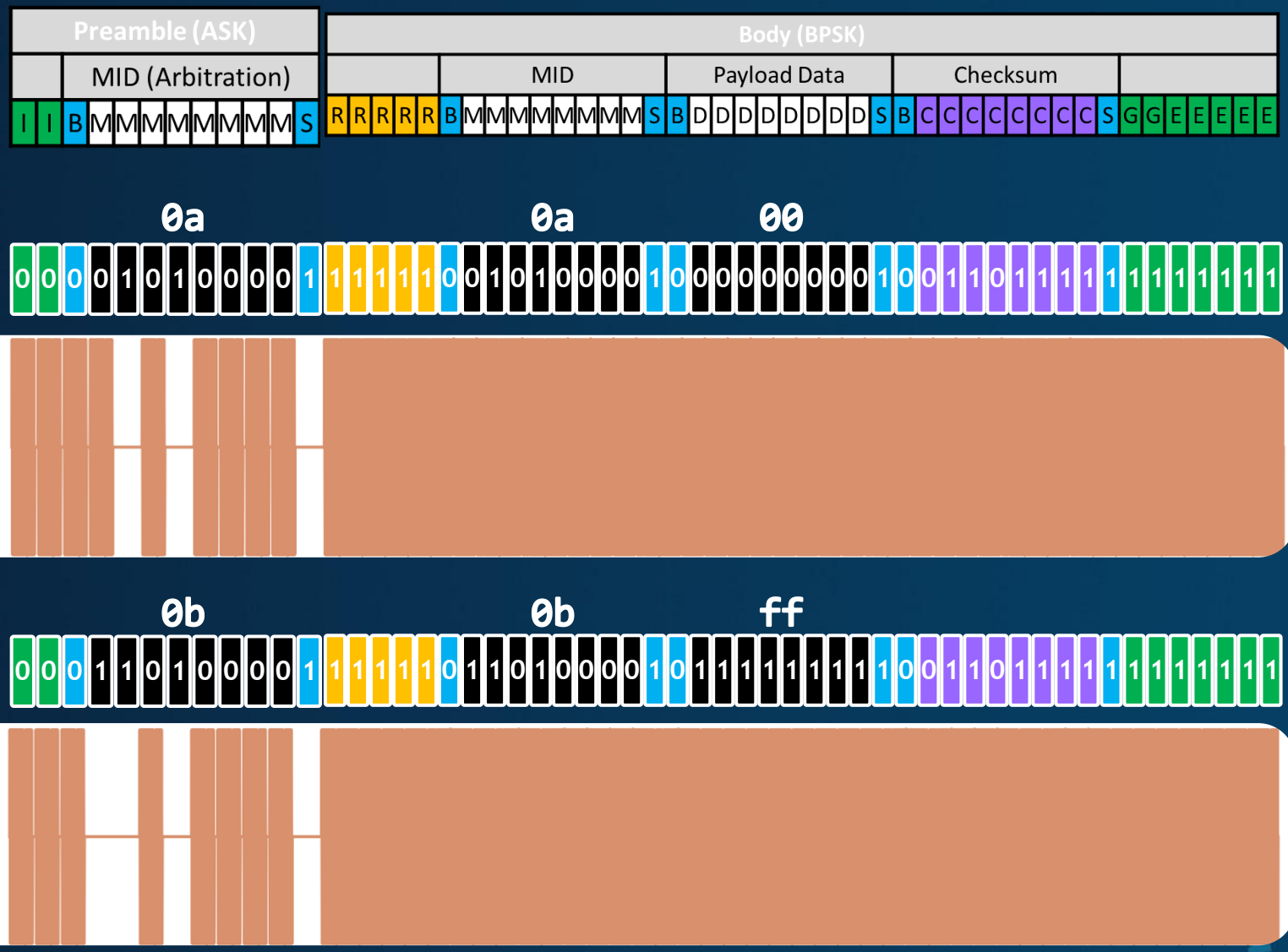
J2497 Specifics (4/4)

Preamble

- Amplitude Shift Keying (ASK)
- Bit time 114us (14us silence after 100us chirp)
- Logic ‘0’ = chirp present
- Initial symbols (1-2 logic ‘0’)
- Start bit (logic ‘0’)
- MID bits (duplicated in body)
- Stop bit (logic ‘1’)

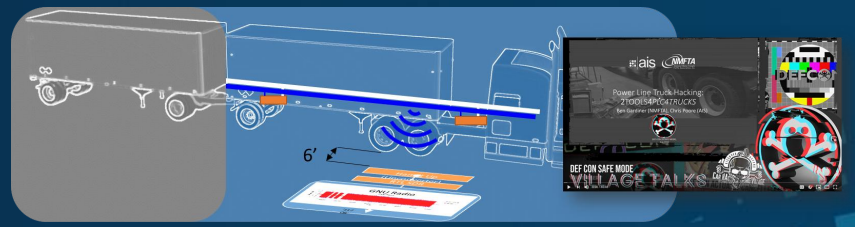
Body

- Phase Shift Keying (PSK), 180deg difference
- Bit time 100us
- Logic ‘0’ symbol is arbitrary per device, determined by the symbol transmitted in the preamble
- Sync symbols (5 logic ‘1’)
- J1708 Body Bytes. MID followed by Data
  - Start bit (logic ‘0’)
  - Data bits (8)
  - Stop bit (logic ‘1’)
- J1708 Checksum Byte
  - Start bit (logic ‘0’)
  - Checksum bits (8)
  - Stop bit (logic ‘1’)
- Gap (0-4 logic ‘1’) & End symbols (5 logic ‘1’)



J2497 Features

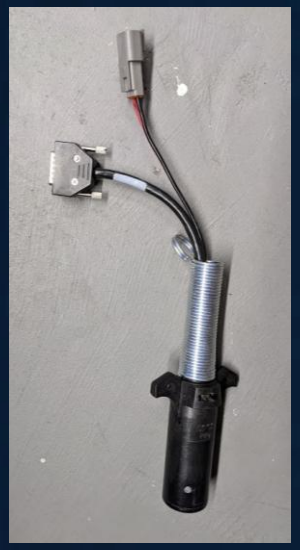
- Primary purpose is for 0a00 and 0bff LAMP ON and LAMP OFF messages. But *there's more*:
- Has all the feature of J1708/J1587 plus:
  - dynamic address (MID) claim (PID 4)
  - data transfer *bridging* (PIDs 204 and 460)
- Trailer brake diagnostic functions such as ABS air pressure valve cycling and ECU reconfiguration
- Some trailer brake ECUs have scripting languages programmable over J2497
- because of the added preamble/MID byte it is possible to create J2497 frames that override bus arbitration
  - e.g. a J2497 priority of maximum 00 and a J1708 priority of minimum ff which overrides all J2497 traffic but is received as MID ff
- Radiates enough energy to be read remotely at 6ft from trailer



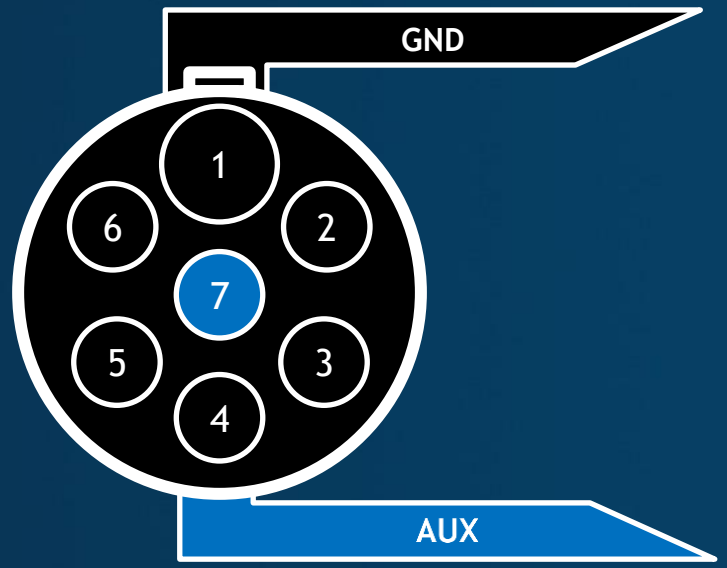
Finding J2497 (1/5)

- Will always be on the power pin (AUX) of the trailer J560 connector 👉 (at back of tractor / front of trailer)

CC BY-SA MobiusDaXter

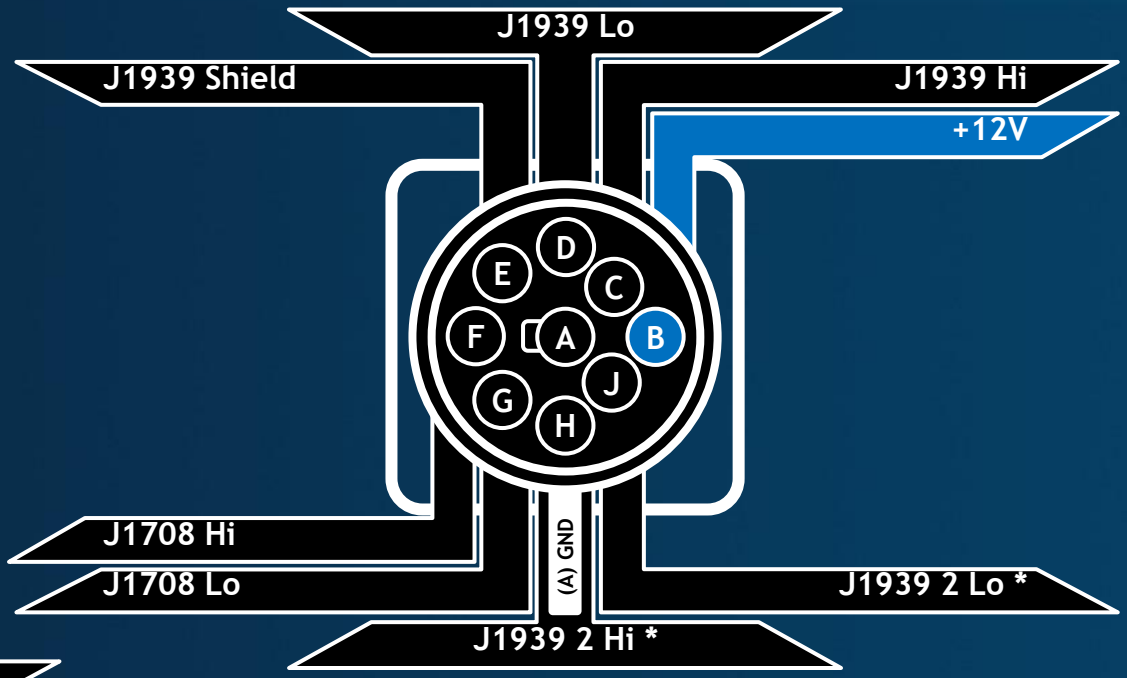


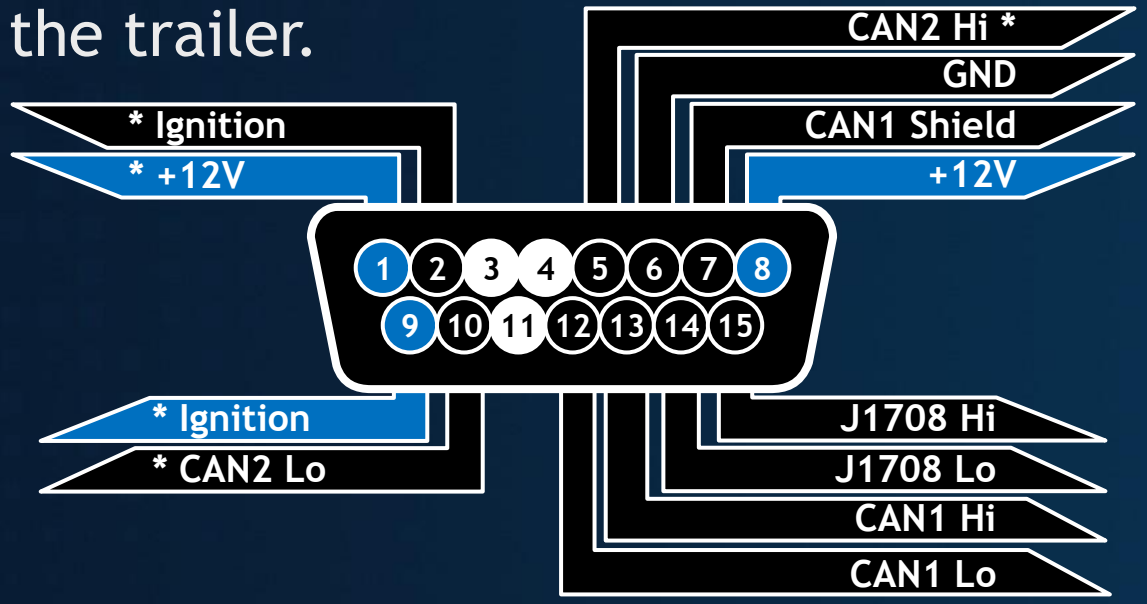
www.ebay.ca/itm/Bendix-ABS-Trailer-Remote-Diagnostic-Unit-TRDU-PLC-Adapter-9-pin-Connection



Finding J2497 (2/5)

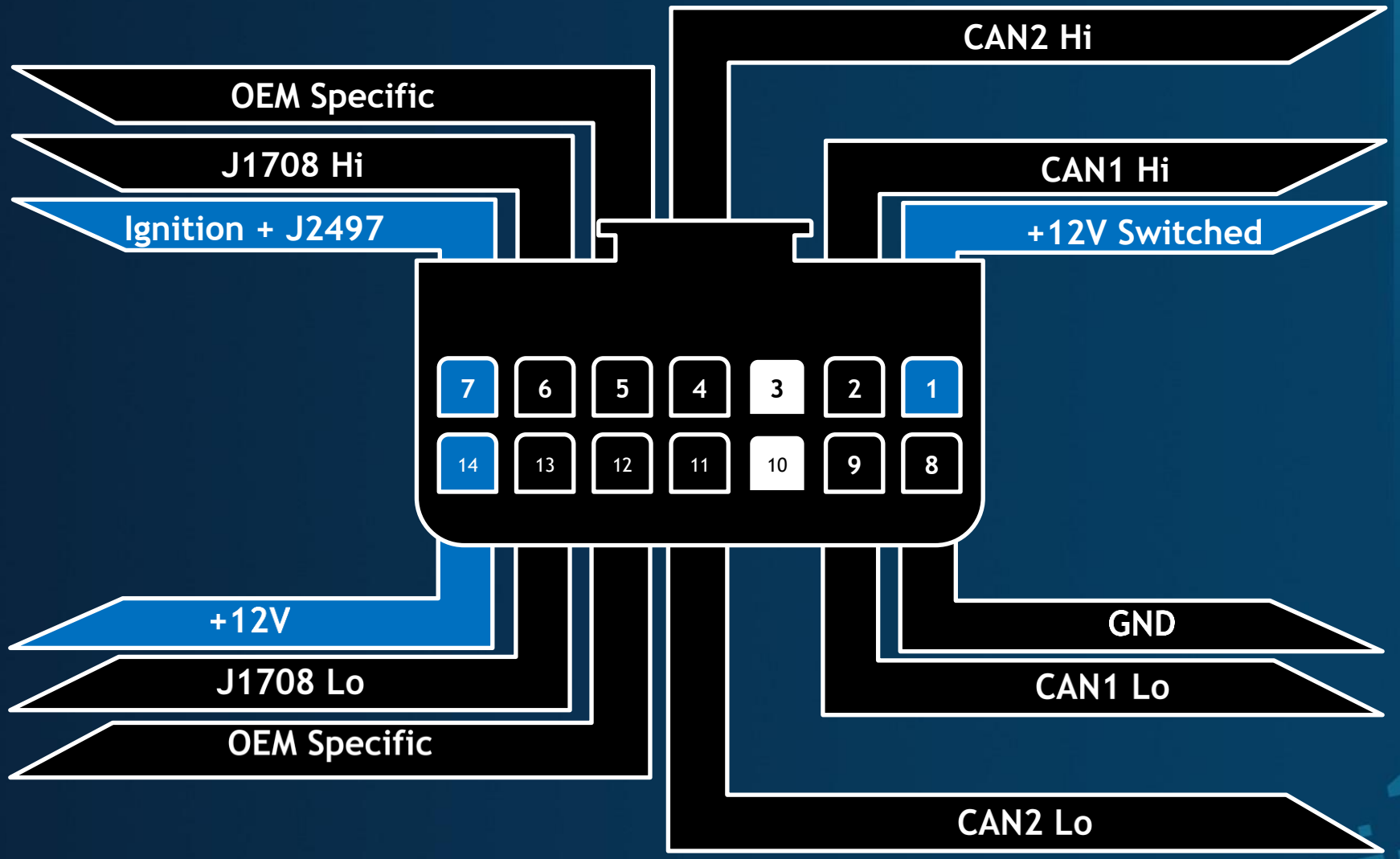
- Might be on the power pins of the diagnostics connector
- What you find could be filtered/segmented from the trailer.





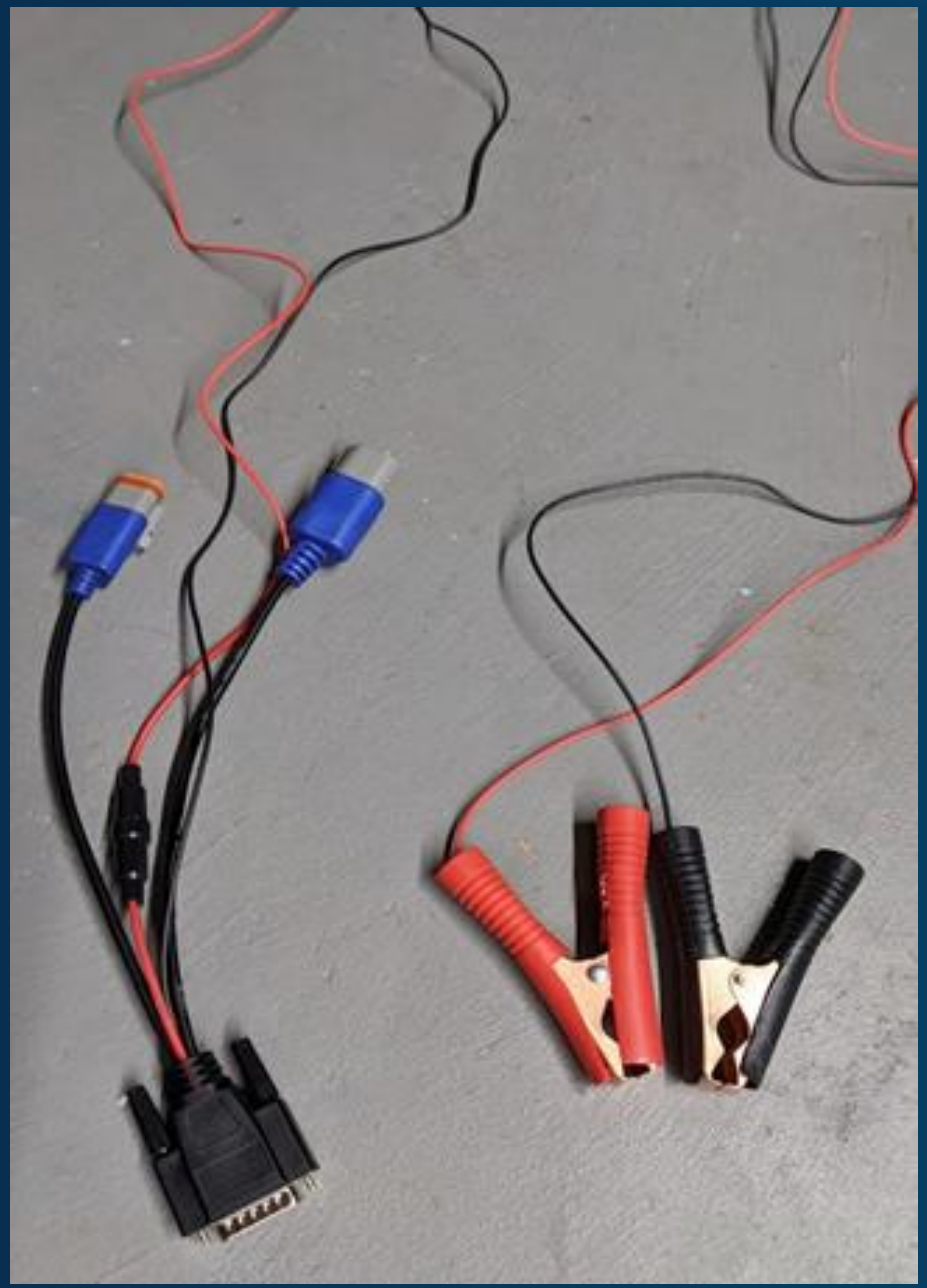
Finding J2497 (3/5)

- Should be on the RP1226 (Aftermarket/ Telematics) Connector



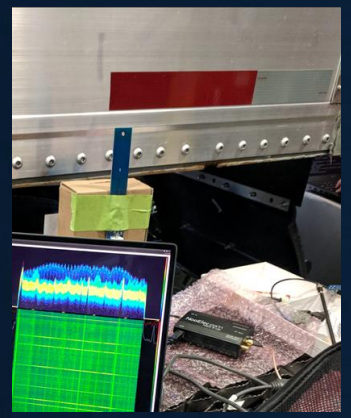
Finding J2497 (4/5)

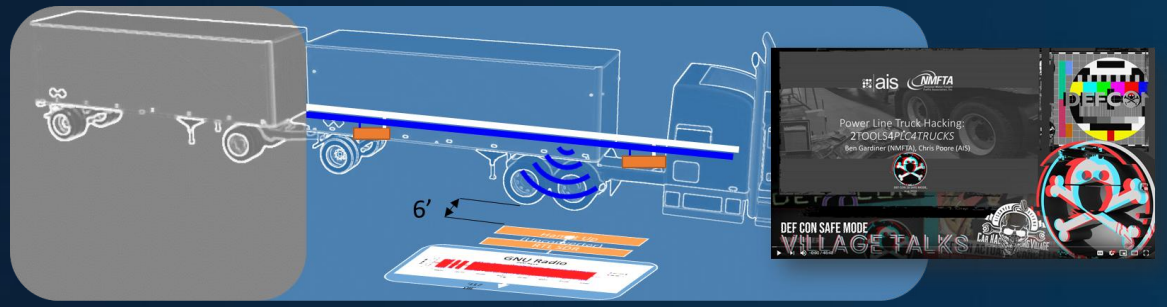
- Might be on the battery terminals -- but what you find could be filtered/segmented from the trailer.



Finding J2497 (5/5)

- Might just radiate away from the trailer. ICSA-20-219-01





Chris Poore & Ben Gardiner. *Power Line Truck Hacking: 2TOOLS4PLC4TRUCKS*

- Might be writable via RF. ICSA-22-063-01

Chris Poore & Ben Gardiner. *Trailer Shouting*, DEF CON 30 :

Truck Vehicle Networks: Automotive Ethernet

Automotive Ethernet Specifics

- Comes in different speeds/flavors:
  - 10BASE-T1S, 100BASE-T1, 1000BASE-T1
    (and these each have their own 802.3xy)
  - BroadR-Reach pioneered these and eventually became 100BASE-T1
- Point-to-point connections; except 10BASE-T1S which can be ‘multi-drop’ (like CAN)
- After you have an adapter and are connected: It is *pretty much* 802.3 Ethernet (like on the back of your WiFi AP at home)
  - Except what’s in the packets of course
- MUCH MUCH MORE in Ivan’s class tomorrow

Automotive Ethernet Features

- Will carry one or both of: Diagnostics, Mission Time
- For Diagnostics:
  - Could be Diag. over IP (DOIP)
  - Could be Service Oriented Vehicle Diagnostics (SOVD)
  - Could be ‘normal’ J1939 or UDS diagnostics packed into the ‘mission time’ vehicle traffic
- For Mission Time:
  - Could put CAN right on top of ethernet frames
  - Could mix with audio/video information
  - Could stick it in Time Sensitive Network (TSN) messages on top of UDP
  - Could stick it into MQTT
  - But somehow, vehicle data (maybe even CAN data) is packed in there…
- MUCH MUCH MORE in Ivan’s class tomorrow

Other Truck Vehicle Networks

Vehicle Networks: More

- LIN
- CAN-FD, CAN-HG
- FlexRay
- MOST
- Much More Wireless

Section Summary: Vehicle Networks Overview

- There are several different networks available to you during assessment
  - You know now what they are
  - You know now where to find them
- Spoofing proprietary space can be powerful (because that's where the control loops mainly rely today)

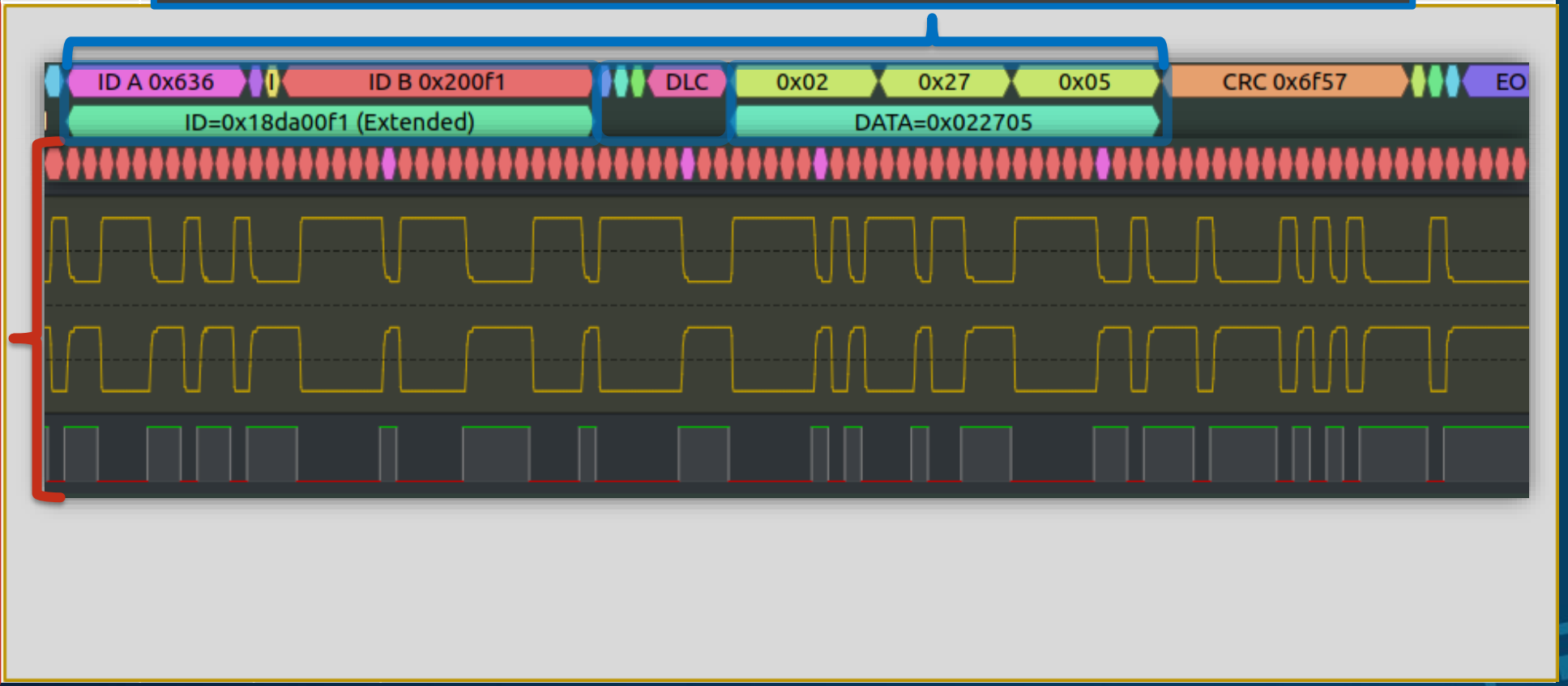
Publicly Disclosed Vehicle Network Cyber-Physical Attack Paths

CAN Attack Methods (below J1939)

	ALL e.g. *Socket CAN*
Bus Flood	Y
(Simple) Spoofing	Y

```
while True:
  sock.send(b'\x00\x00\x00\x00\x01\x00')
```

```
sock.send(b'\x18\xda\x00\xf1\x03\x02\x27\x05')
```



?

CAN Attack Methods (below J1939)

	ALL e.g. *Socket CAN*	*CAN Hack*	*CANT*	*CANHack* by Dr. Ken Tindell @ **CANIS** *CANT* by b1tbane & ehntoo @ **GRIMM** --- **Notes:**
Bus Flood	Y			
(Simple) Spoofing	Y			← the means for nearly all the attacks discussed next
Bus-Off / Bus Killer		Y	Y	target an ECU and destroy its frames repeatedly with selective bit override Cho et. Al & Maggi, F. c.f. also ICS-ALERT-17-209-01
(ASAP) Spoofing		Y		takes advantage of bitbanging to ensure attack frame is entered into arbitration ASAP after the target frame
Double Receive		Y		make a transmitter double-send a frame, error is only visible to transmitter and every other node receives same frame twice
Freeze / Overload		Y	Y	send a number of overload frames after a target frame
Error Passive Spoofing / Data Replacer		Y	Y	put a target into error passive mode then put an attack frame in front of a target frame aka "*bit smashing*"
Janus [Tindell]		Y		create a custom bitstream for two sampling point values so that receivers configured to those sampling points will receive different frames
Bus Short			Y	(cyber paperclip mode): shorts CAN_H+CAN_L **(requires analog switch)**
NACK			Y	clobber ACK bit by asserting a recessive state on the bus **(requires analog switch)**
(Improved) **Data Replacer**			Y	" but can clobber also dominant bits **(requires analog switch)**

What can you do on Heavy Vehicle Networks (1/2)?

- Some examples that have been made public
- Each result is true only on specific model year builds of trucks

What can you do (2/2)? Summary

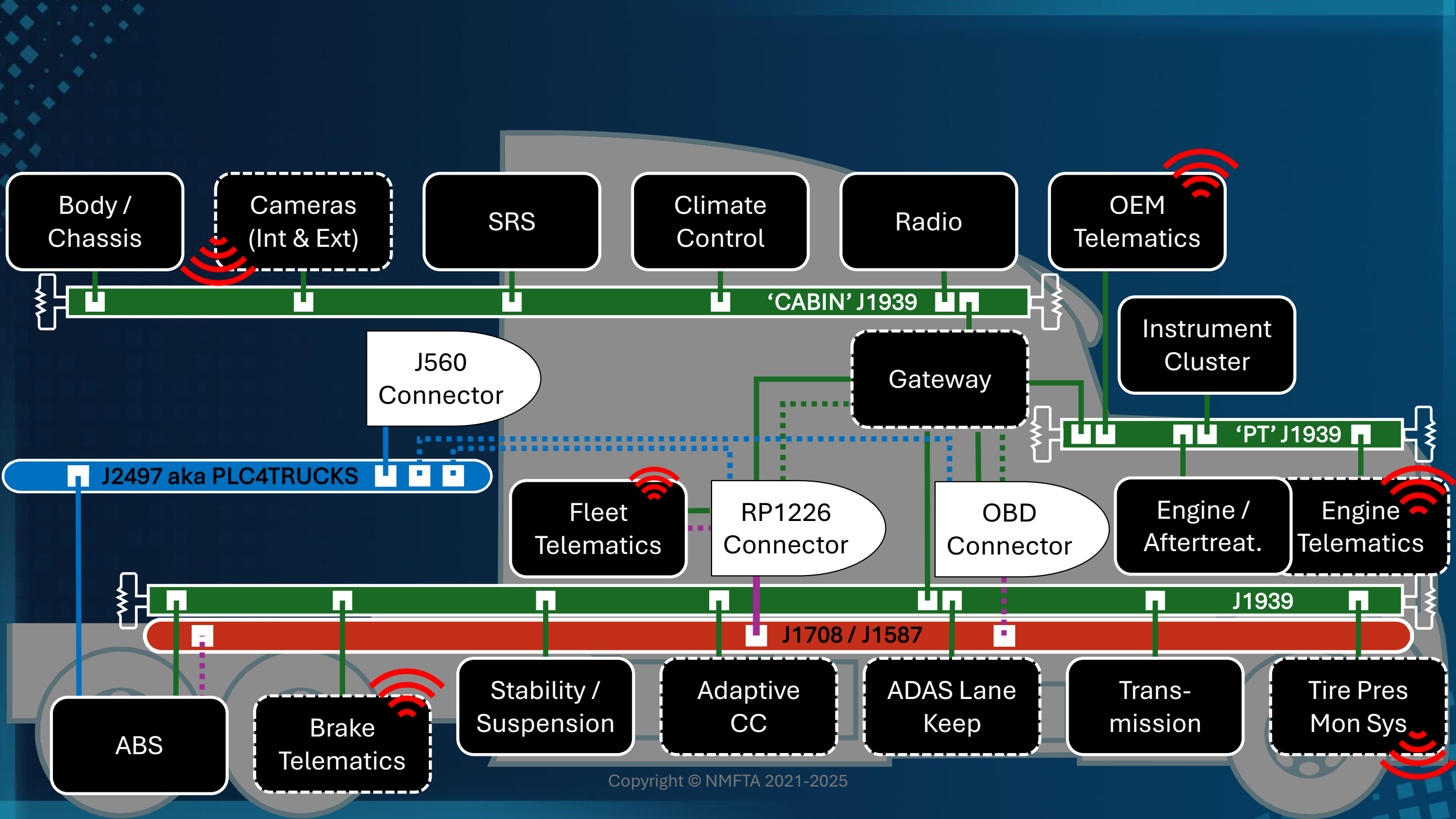
Network	'Hacking'	*Who*
J1939	Vehicle disable/limp by DEF message manipulation	Jonson (NMFTA)
J1939	Denial of ECUs	Mukherjee et. al.
J1939	Vehicle disable/limp by de-rate message manipulation	Leale (CanBusHack)
J1708/J1587	Malicious misconfiguration of truck ECM	Haystack et. al.
J1939	Instrument Cluster override	Burakova et. al.
J1939	RPM control and engine brake disable	Burakova et. al.
J1708/J1587	Disable engine cylinders	Burakova et. al.
J1708/J1587 / J2497	Cycle ABS air release valves	Burakova et. al.
J2497	Remote cycle trailer ABS air release valves	Gardiner (NMFTA) et. al.

Section Summary: Public Attacks

- There are many ways to cause a de-rate
- There are protocol attacks that are possible beyond ‘simple’ spoofing

Vehicle Network Segments / Gateways

Trustworthy or Not



Untrustworthy Network Domain (UND)

A **UND** is a network domain (or segment) that contains **any** of the following:

- wireless communication or reachable wirelessly (e.g J2497, ISO 15118, BlueTooth, WiFi, TPMS, ZigBee)
- an interface for aftermarket devices or operator access (e.g. an OBD port, RP1226 connector)
- a telematics device that does not satisfy the HD VCR requirements
- a multi-network device (e.g. intended or unintended gateway) that does not satisfy the HD VCR requirements

Trustworthy Network Domain (TND)

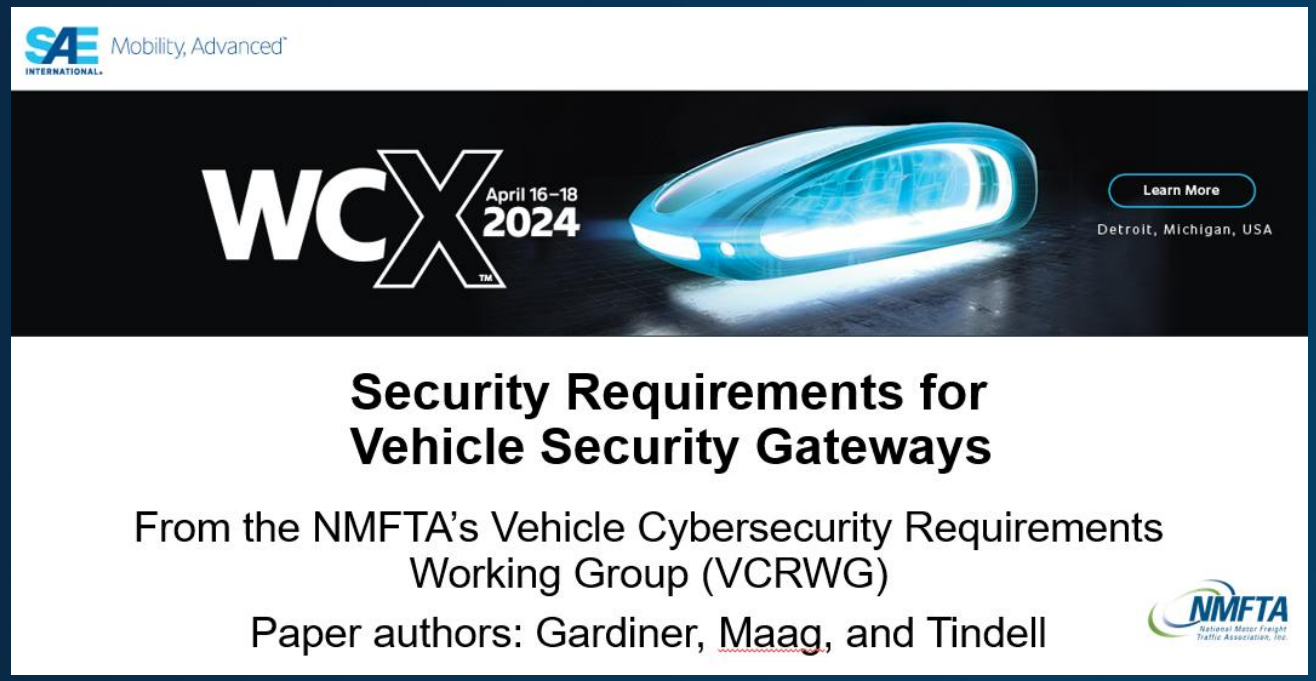
A **TND** is a network domain (or segment) that does not contain any of the above.

More Details

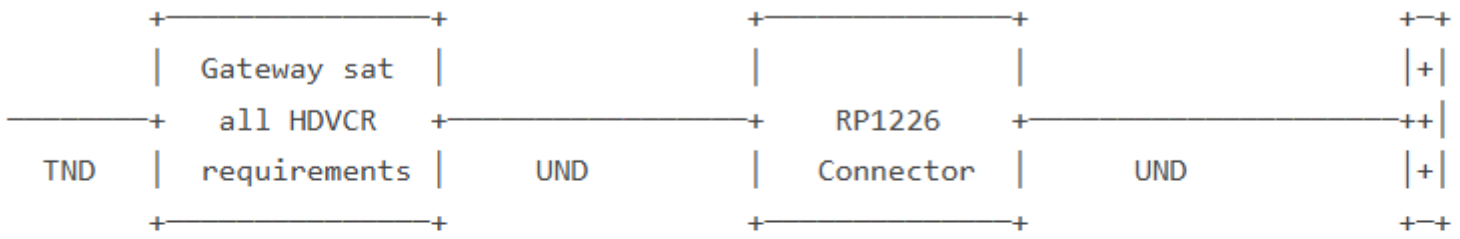
See

https://nmfta-repo.github.io/vcr-experiment/

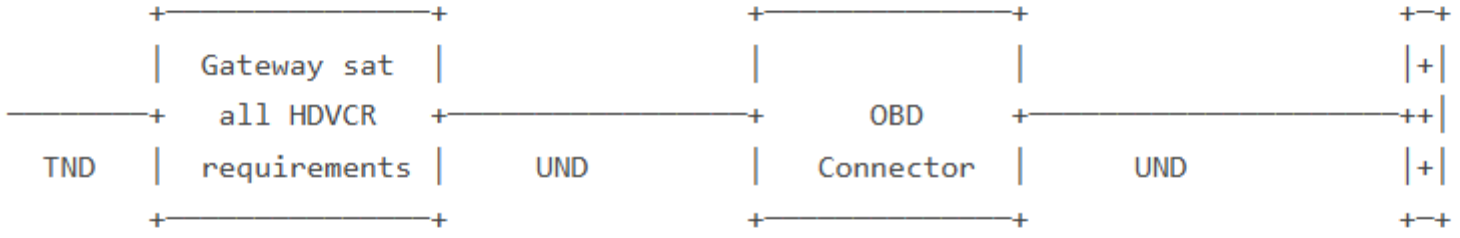
for the paper and browsable requirements



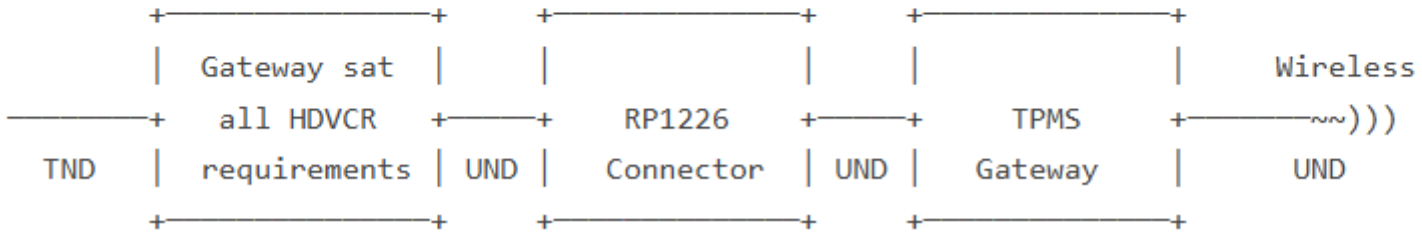
Examples



The OBD connector is _the_ operator access port; therefore the databuses there are UNDs.



The TPMS gateways on RP1226 databus connect to a wireless medium and are on the aftermarket bus; therefore the databuses there are UNDs.



Gateway Devices

(Intended) Gateways
Unintended Gateways

(Intended) Gateways

"If it walks like a duck and talks like a duck then it is a duck"

←duck typing

- We define gateways as any device that
  - Has multiple network segment connections AND
  - Does any of the following between UND and TND segments:
    - Transports
    - Translates
    - Filter/Drop/Rate-Limit
    - Encapsulate

A Gateway is as a Gateway Does

- **Transports**: These devices transport/'move' information between two separate network 'domains,' bi-directionally.
- **Translates**: These devices translate/transform the information between the separate network domains but intentions of the data are preserved. 3
- **Filter/Drop/Rate-Limit**: These devices select which information is transported based on rules matching content or metadata before (ingress) or after (egress) processing steps in preparation for transport/translate/encapsulate or other forwarding actions. These devices select which information is transported in a time varying fashion for the purposes of limiting the rate at which the information is put on a network domain.
- **Encapsulates**: These devices encapsulate information as it is transported and/or translated between the network domains.

Why Duck Type?

A: Because attackers don’t care how engineering defined the ECU

i.e. Any one of those functions is enough to have a security impact on TND segments.

Unintended Gateways

- A Device which
  - Has multiple network segment connections AND
  - does not perform any of
    - Transports
    - Translates
    - Filter/Drop/Rate-Limit
    - Encapsulate

BUT these devices *could* perform any of those if compromised by e.g. remote code execution;
therefore, there are security requirements to consider for these as well

Not a Gateway: Interior Devices

A Device which

- Has connections to one or more vehicle network segments AND
- None of the segments connected are UNDs

None of the gateway security requirements apply; however, there will be generic security requirements for these e.g.

- ISO/SAE 21434
- UNECE WP.29
- The Truck Matrix (more later)

Gateway Device Security Requirements

e.g.

2.2.1 Won't Transport

UID:	NGW-S-002
CRITICALITY:	High

These devices SHALL NOT transport/'move' information between two separate network 'domains,' in either bidirection.

Parents:

- [NGW-S-001] *Security Assurance*

2.2.2 Won't Translate

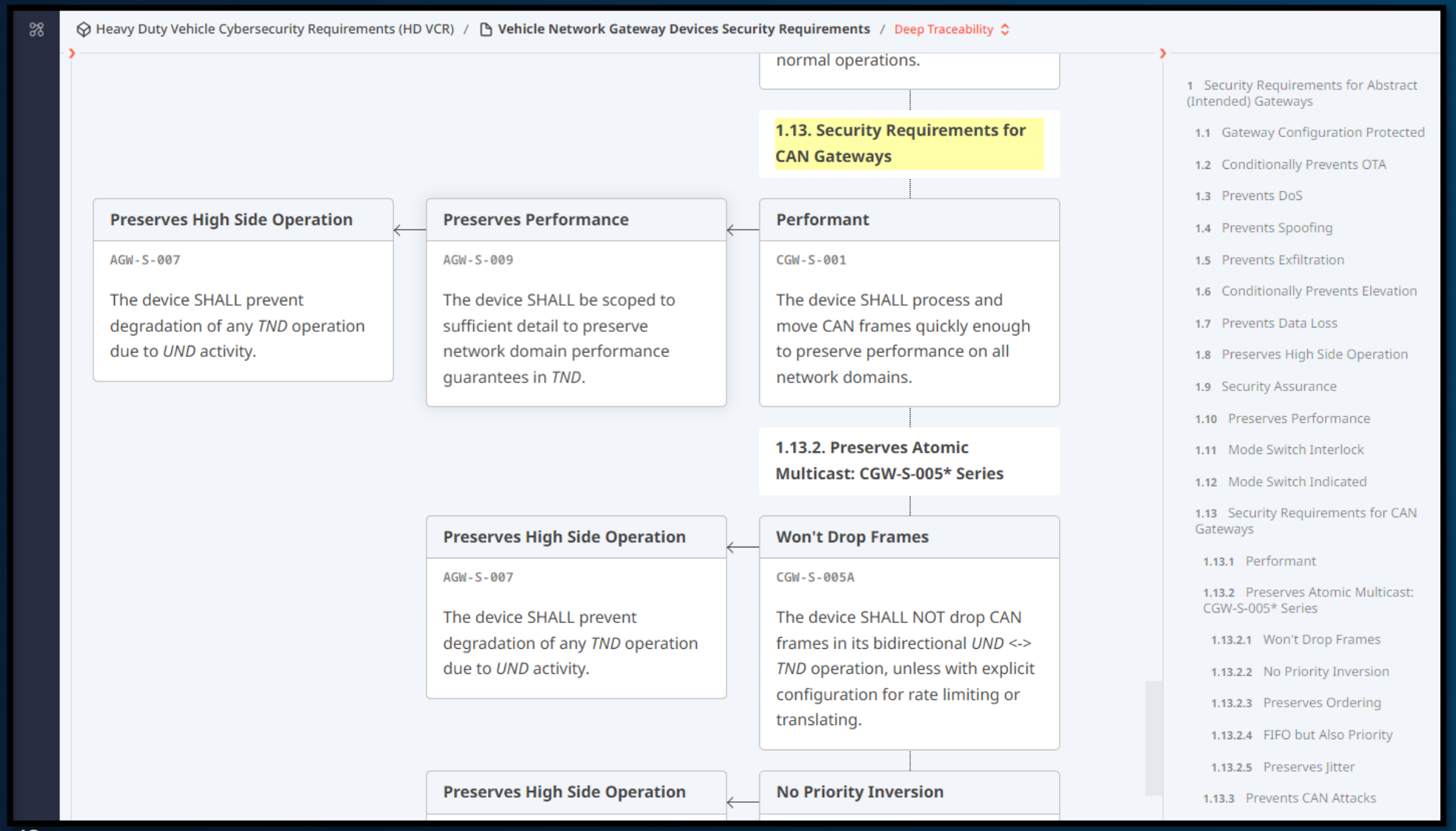
UID:	NGW-S-003
CRITICALITY:	High

These devices SHALL NOT translate/transform the information between the separate network domains.

Parents:

- [NGW-S-001] *Security Assurance*

Feature: Browsable



Feature: Fleet Acceptance Tests

- The Security Requirements include acceptance tests that can ideally be performed by the fleets
- There are cases where a fleets couldn’t reasonably be expected to execute the acceptance test and in these cases third-party testing is recommended (this is the same as with the TSRM)

More: The Truck Matrix

Just Piece of the Puzzle

- The (extensive) security requirements for Security Gateways are just a part of the Truck Matrix; both:
  - What is presently available as Truck Matrix resources AND
  - What is being developed for the Truck Matrix

https://github.com/nmfta-repo/nmfta-vehicle_cybersecurity_requirements

Network Topology Survey and Risk Analysis

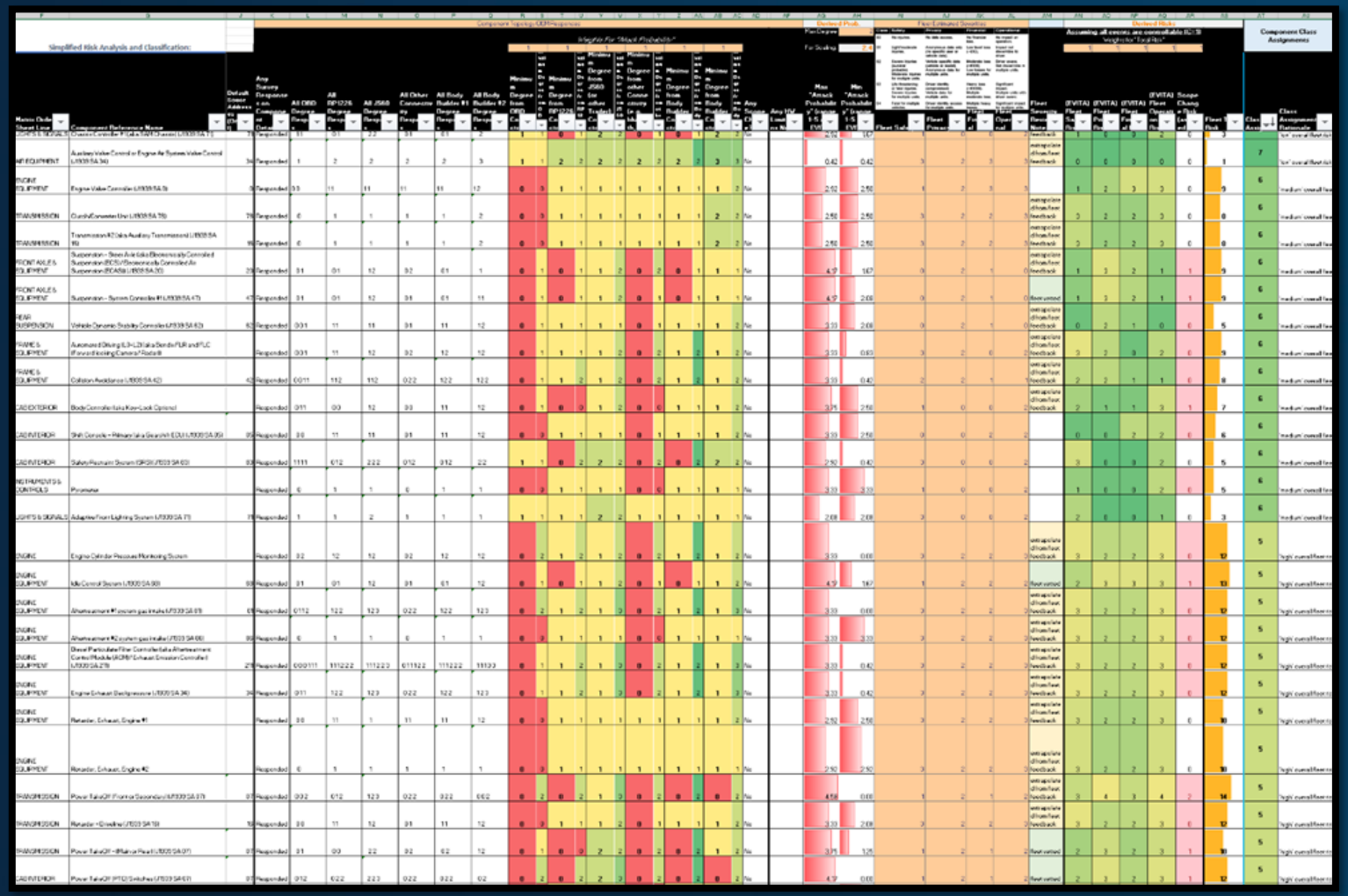
All the survey results from OEMs collected

Basic **risk analysis** and device classification

Could be improved too:

- More impact input from fleets
- More vehicle survey results from OEMs

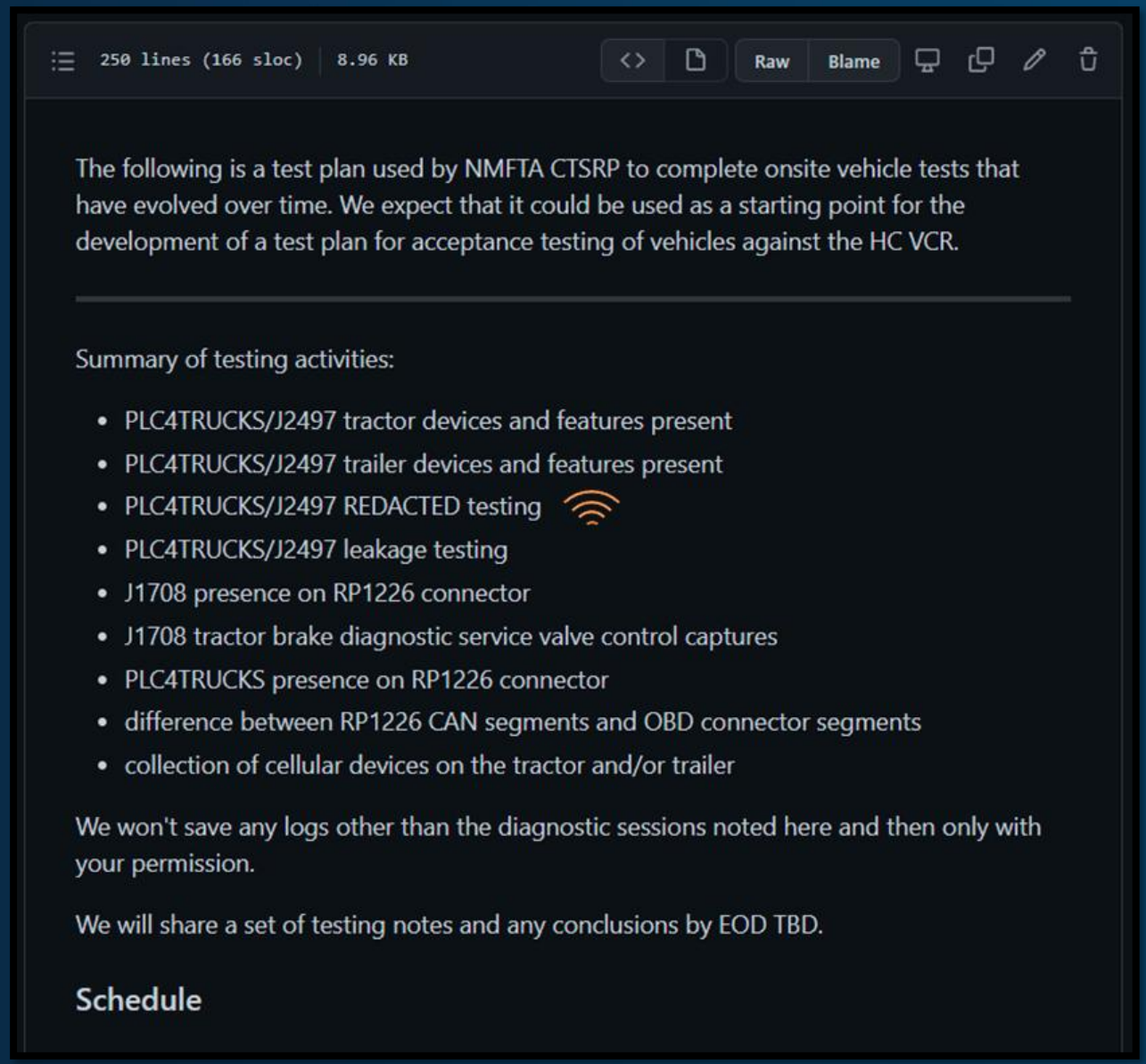
Note: can be used to run analysis of a particular vehicle to compare against this classification



https://github.com/nmfta-repo/nmfta-vehicle_cybersecurity_requirements/blob/main/resources/Component_Class_Assignment_v18_DRAFT.xlsx

Other Resources: Truck Testing Plan

We have committed our heavy vehicle testing plan to the repo to serve as a seed for hosting the eventual test plan corresponding to the vehicle security requirements.



https://github.com/nmfta-repo/nmfta-vehicle_cybersecurity_requirements/blob/main/resources/heavy_vehicle_testing_plan.md

Section Summary: Vehicle Network Segments and Gateways

Many resources that could be useful to you:

- existing definitions for trustworthy (and not) segments
- existing security requirements with acceptance tests for intended (and unintended) gateways
- network survey excel sheet with risk estimates and could be used to assess a particular vehicle's segmentation-based risk
- bonus: truck testing plan used by NMFTA

Replay Attacks

Replay Attack

- Repeating the captured traffic – without modification (but perhaps of timing)
- Some protocols are vulnerable to replay attacks.

e.g.

- A diagnostic function doesn't require seed-key first
- A session key can be installed repeatedly
- A instrument cluster displays whatever it receives

How do you test for susceptibility?

How do you do find the minimal sequence to replay in a large log file?

Optimum Search: Bisect

- Computer Science / Information Theory result: the most efficient way to search for something located randomly in a collection is to iteratively divide in half.
- Method detailed in the Car Hacker's handbook https://archive.org/details/car-hackers-handbook-the-craig-smith

Bisect with linux can-utils

- You can use the script proposed in an unmerged can-utils pull request: https://github.com/linux-can/can-utils/pull/17
- ⚠ canplayer (used by that script) has a bug: it does not respect timestamps and will play things back as-fast-as-possible⚠

Bisect with anything else

- Do it manually:
- Divide the log into A and B
- Test A, test B.
- Whichever half gives a result: repeat with that half.

Section Summary: Replay

- The bisection search is the most efficient way to search for susceptibility to replay
- You can definitely roll your own bisect – or do it manually

Enumerating ‘Controller Applications’

What is this ‘Controller Application’ (CA)

- It is the J1939 term for an addressable piece of software running on a host (i.e. an ECU)
- Analogy: IP addresses on a server – there can be multiple

- BUT USUALLY: a CA is an ECU

How to Passively Enumerate CAs

This method relies on assuming the ECUs have not changed their address (recall that ECUs can change addresses dynamically)

Look at traffic logs; the source and destination addresses are in the CAN Arbitration IDs.

1. For each arbitration ID:
    1. Record the last byte of the ID (the source address) as a CA address
    2. If the PGN < 0xF000:
       Also record the second-last byte (the dest address) as a CA address
2. Lookup the NAME Function for each CA address (assuming default)

Look-up of NAME Functions

- You can use the Digital Annex
- You can use other things google-able on the internet
- You can use the NMFTA Truck Matrix Component Class Assignment spreadsheet too

  https://github.com/nmfta-repo/nmfta-vehicle_cybersecurity_requirements/blob/main/resources/Component_Class_Assignment_v18_DRAFT.xlsx

  - BONUS: comes with fleet-risk estimates for each CA – useful in security assessment prioritization

📝 Hands-On: Passively Enumerate CAs

Using the cybertruckchallenge.org truck challenge vehicle log file and the NMFTA Truck Matrix Component Class Assignment spreadsheet answer these questions:

A. How many CAs are broadcasting?

B. Are there other known CAs?

(now assume all CAs have their default address)

C. Which CA has the highest fleet risk?

D. Which CA has the lowest fleet risk?

1. Look at the candump format log file at https://www.cybertruckchallenge.org/participate/truck-challenges/
2. Collect all the source (and destination addresses you find there)

   e.g. by opening the logfile with wireshark and add 'decode as' : J1939
3. Answer A. and B.
4. Download and open the spreadsheet at https://github.com/nmfta-repo/nmfta-vehicle_cybersecurity_requirements/blob/main/resources/Component_Class_Assignment_v18_DRAFT.xlsx
5. Answer C. and D.

How to Actively Enumerate CAs

- Send (broadcast) a PGN request (PGN 0x00EA00) for address claimed PGN (0x00EE00)

For each response of PGN 0x00EE00:

1. Record the source address
2. Decode the payload as NAME
3. Lookup NAME Function

This can be scripted pretty easily. OR…

📝 Hands-on: Actively Enumerating CAs with TruckDevil

Using TruckDevil's ecu_discovery module, actively enumerate the controller applications on the benchtop ECU

1. Open a terminal, cd to the TruckDevil git checkout
2. cd (again) into the truckdevil folder
3. Run truckdevil with `python truckdevil.py`
4. Run `add_device socketcan can0 50000` to add a device (or an equivalent python-can add on windows 💪)
5. Run `run_module ecu_discovery`
6. Run `active_scan`
7. Run `view_ecus`

Section Summary: Enumerating CAs

- CAs are roughly/probably the ECUs
- You can passively ‘read’ the CAs present from the PGNs (e.g. in a traffic log)
  - IF you assume the CAs took their default addresses
- You can actively ‘scan’ for CAs using PGN request for the address claimed PGN
  - You can re-use the TruckDevil `ecu_discovery.active_scan` module function for this
- CA default addresses and NAME Functions can be looked up in various ways

CyberTruck Challenge™ Assessment Tips

Targets I've Noticed

- TODO onsite
- And of course tractor brake service valve tests

Things I Brought That you can BORROW

- CHV Badge Automotive Ethernet Adapter
- Various J42497/PLC tools and connectors
- SIM Card reader
- Induction J1939 and J1708 adapters
- A bunch of USB OTG adapters
- RP1226 cable for modification
- LANTap
- CANHack bitbang protocol attack tool
- SDR things: hackrf, amplifiers, high gain and directional antennas
- 😳 ~~Pwnagotchi (little bettercap platform for web-interface wireless hacking)~~ 😭
- BTLEJack BLE sniffer (3x / all-band) also works with Mirage
- USB-PD 5V-15V variable power supplies
- Some soldering tools (incl hotplate) and chipquick solder (great for DE-soldering)

Fleet Context to Keep in Mind

- Where you connect on the vehicle matters
  - OBD: Where fleets connect their maintenance laptops _often_
  - RP1226: Where fleets _should_ connect their telematics
  - J560 / TT CAN: Where fleets trade trailers; and left accessible in public spaces
  - Internal Segments: Not often connected-to BUT are there OEM/Supplier telematics there?
- Fleet Impact schedule
  1. AVAILABILITY
  2. CONFIDENTIALITY
  3. INTEGRITY (except where it can affect 1. in turn)

Conclusions

Conclusions

- Trucks have 3 main types of vehicle networks:
  J1939, J1708/J1587, and J2497
  - Two (J1939 and J2497) are on all trucks in North America
  - You've learned all about J1939 from Dr. Daily
  - There is also Auto. Ethernet – you'll learn all about this from Ivan Granero
  - There is also LIN, MOST, FlexRay and more – not really covered at CTC™
- Proprietary Messages can be juicy
- There are many ways to de-rate
- Vehicle network gateways come in both intended and unintended forms and can highly impact security
- Controller Applications are (roughly) ECUs and can be discovered both passively and actively
- There are free resources to consult at github.com/nmfta-repo
- There are a host of free tools for interacting with vehicle networks
- Binary search is king (don't worry about trinary et. al.)