A Model-Driven Methodology for Automotive Cybersecurity Test Case Generation

SRCNAS/STRIVE WS @ IEEE EURO S&P�much 21

September 6, 2021

Stefan Marksteiner, Peter Priller

The Need for Industrialized Automotive Cybersecurity Testing

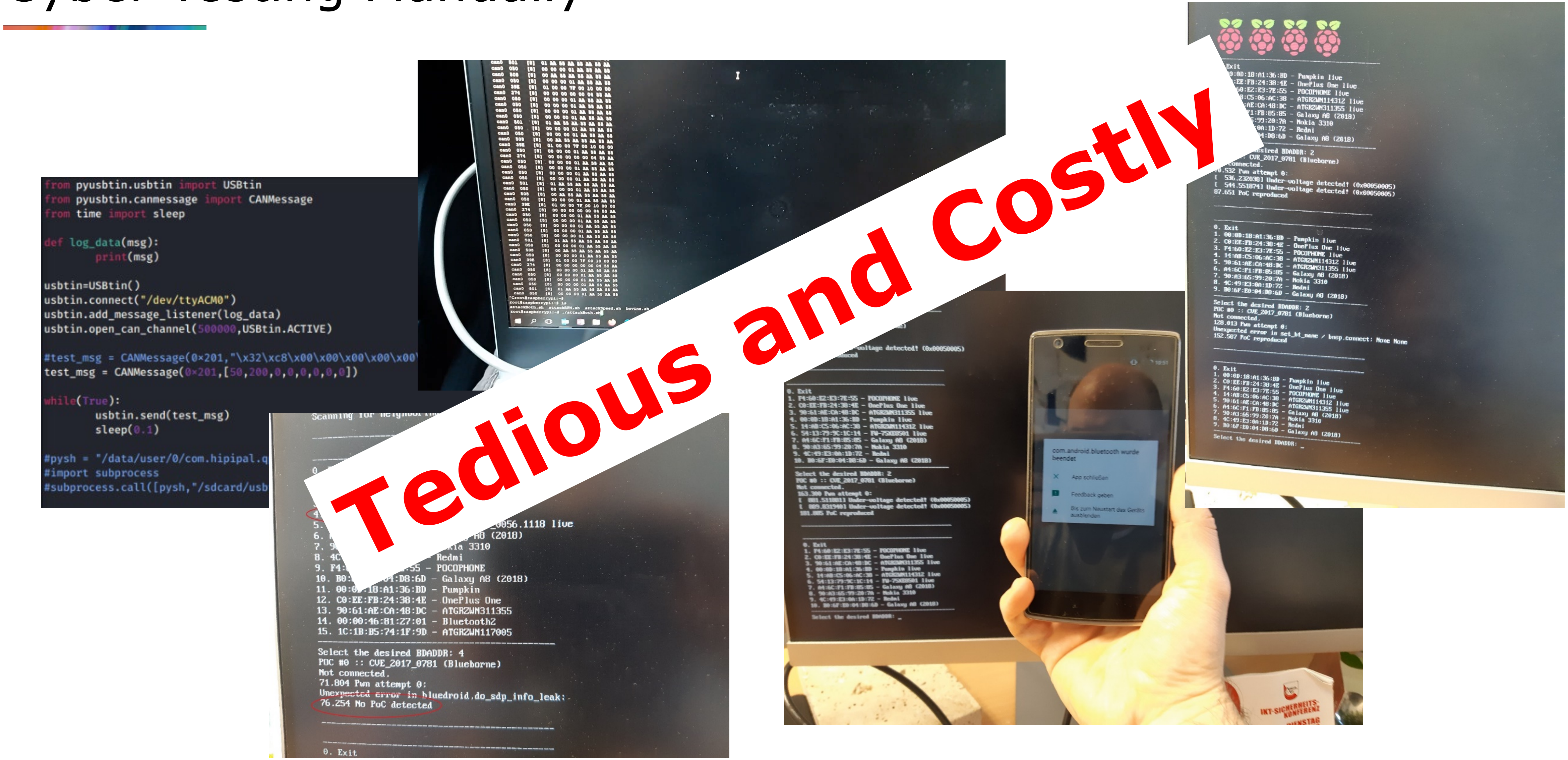
- UNECE
  - Regulation R.155
  - Mandates cybersecurity and cybersecurity management
  - Requires testing of measures
  - Adopted in EU, Japan and Korea
  - Effective in EU for new types 2022 and for all new vehicles 2024
- ISO/SAE 21434
  - Cyber security management system for automotive systems
  - Risk-based approach
  - Also demands testing, however, does not specify details
  - To be supplemented for testing by ISO PWI 8477 (V&V) and ISO/SAE PWI 8475 (CAL &TAF)

=> Need for automated testing

Why Black Box Testing?

- Providing an attacker’s view
- Long supply chain – source might not be available
- Unwillingness (or inability) to disclose internals

Cyber Testing Manually



Tedious and Costly

```
from pyusbtin.usbtin import USBtin
from pyusbtin.canmessage import CANMessage
from time import sleep

def log_data(msg):
        print(msg)

usbtin=USBtin()
usbtin.connect("/dev/ttyACM0")
usbtin.add_message_listener(log_data)
usbtin.open_can_channel(500000,USBtin.ACTIVE)

#test_msg = CANMessage(0×201,"\x32\xc8\x00\x00\x00\x00\x00
test_msg = CANMessage(0×201,[50,200,0,0,0,0,0,0])

while(True):
        usbtin.send(test_msg)
        sleep(0.1)

#pysh = "/data/user/0/com.hipipal.q
#import subprocess
#subprocess.call([pysh,"/sdcard/usb
```

```
Scanning for neighboring
-------------------------------------------
-------------------------------------------
0
4
5.                              _0056.1118 live
6.                                  A8 (2018)
7. 9                             kia 3310
8. 4C                            Redmi
9. F4:          :55 - POCOPHONE
10. B0:      04:D8:6D - Galaxy A8 (2018)
11. 00:0  :18:A1:36:BD - Pumpkin
12. C0:EE:FB:24:38:4E - OnePlus One
13. 90:61:AE:CA:48:DC - ATGRZWN311355
14. 00:00:46:81:27:01 - Bluetooth2
15. 1C:1B:B5:74:1F:9D - ATGRZWN117005
-------------------------------------------
Select the desired BDADDR: 4
POC #0 :: CVE_2017_0781 (Blueborne)
Not connected.
71.804 Pwn attempt 0:
Unexpected error in bluedroid.do_sdp_info_leak:
76.254 No PoC detected
-------------------------------------------
-------------------------------------------
0. Exit
```

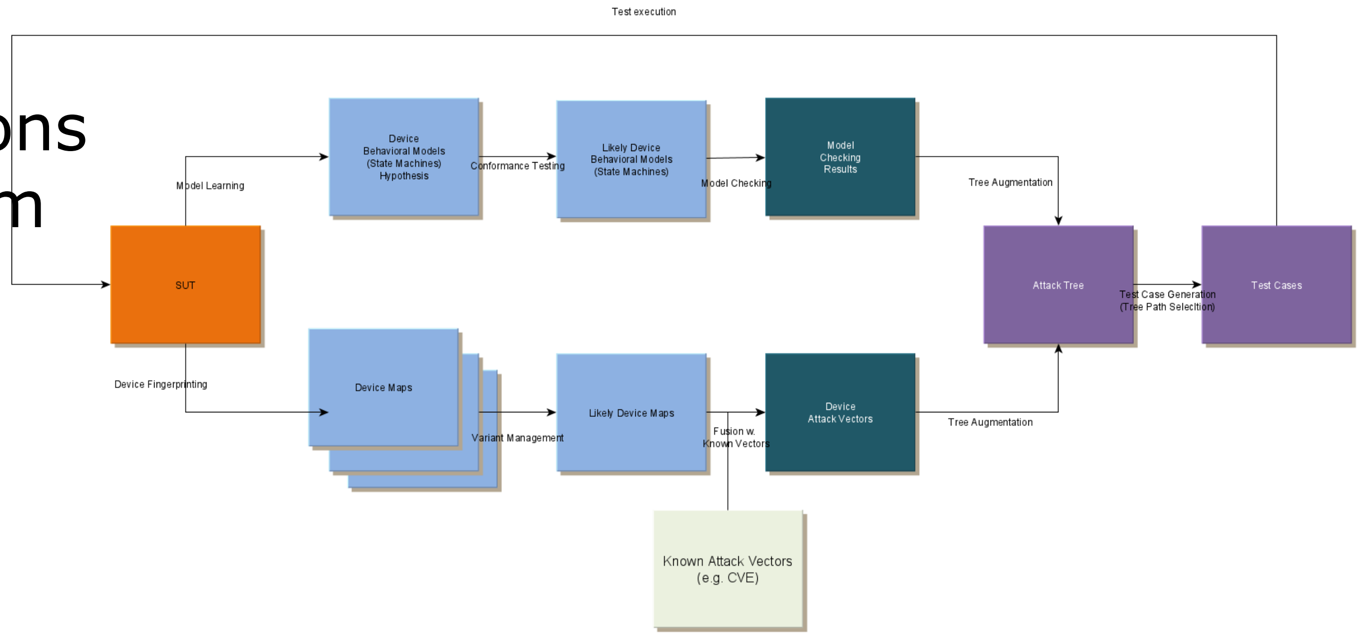
```
-------------------------------------------
                    -voltage detected! (0x00050005)
                    duced
-------------------------------------------
-------------------------------------------
0. Exit
1. F4:60:E2:E3:7E:55 - POCOPHONE live
2. C0:EE:FB:24:38:4E - OnePlus One live
3. 90:61:AE:CA:48:DC - ATGRZWN311355 live
4. 00:0D:18:A1:36:BD - Pumpkin live
5. 14:AB:C5:06:AC:38 - ATGRZWN114312 live
6. 54:13:79:9C:1C:14 - FW-75XE8501 live
7. A4:6C:F1:FB:85:85 - Galaxy A8 (2018)
8. 90:A3:65:99:20:7A - Nokia 3310
9. 4C:49:E3:0A:1D:72 - Redmi
10. B0:6F:E0:04:D8:6D - Galaxy A8 (2018)
-------------------------------------------
Select the desired BDADDR: 2
POC #0 :: CVE_2017_0781 (Blueborne)
Not connected.
163.300 Pwn attempt 0:
[  881.511881] Under-voltage detected! (0x00050005)
[  889.831940] Under-voltage detected! (0x00050005)
181.885 PoC reproduced
-------------------------------------------
-------------------------------------------
0. Exit
1. F4:60:E2:E3:7E:55 - POCOPHONE live
2. C0:EE:FB:24:38:4E - OnePlus One live
3. 90:61:AE:CA:48:DC - ATGRZWN311355 live
4. 00:0D:18:A1:36:BD - Pumpkin live
5. 14:AB:C5:06:AC:38 - ATGRZWN114312 live
6. 54:13:79:9C:1C:14 - FW-75XE8501 live
7. A4:6C:F1:FB:85:85 - Galaxy A8 (2018)
8. 90:A3:65:99:20:7A - Nokia 3310
9. 4C:49:E3:0A:1D:72 - Redmi
10. B0:6F:E0:04:D8:6D - Galaxy A8 (2018)
-------------------------------------------
Select the desired BDADDR: _
```

com.android.bluetooth wurde beendet

```
-------------------------------------------
 Exit
 0:0D:18:A1:36:BD - Pumpkin live
  :EE:FB:24:38:4E - OnePlus One live
  60:E2:E3:7E:55 - POCOPHONE live
  B:C5:06:AC:38 - ATGRZWN114312 live
   :AE:CA:48:DC - ATGRZWN311355 live
   F1:FB:85:85 - Galaxy A8 (2018)
   5:99:20:7A - Nokia 3310
    0A:1D:72 - Redmi
    4:D8:6D - Galaxy A8 (2018)
-------------------------------------------
   esired BDADDR: 2
   : CVE_2017_0781 (Blueborne)
 connected.
 0.532 Pwn attempt 0:
[  536.232038] Under-voltage detected! (0x00050005)
[  544.551874] Under-voltage detected! (0x00050005)
87.651 PoC reproduced
-------------------------------------------
-------------------------------------------
0. Exit
1. 00:0D:18:A1:36:BD - Pumpkin live
2. C0:EE:FB:24:38:4E - OnePlus One live
3. F4:60:E2:E3:7E:55 - POCOPHONE live
4. 14:AB:C5:06:AC:38 - ATGRZWN114312 live
5. 90:61:AE:CA:48:DC - ATGRZWN311355 live
6. A4:6C:F1:FB:85:85 - Galaxy A8 (2018)
7. 90:A3:65:99:20:7A - Nokia 3310
8. 4C:49:E3:0A:1D:72 - Redmi
9. B0:6F:E0:04:D8:6D - Galaxy A8 (2018)
-------------------------------------------
Select the desired BDADDR: 2
POC #0 :: CVE_2017_0781 (Blueborne)
Not connected.
128.013 Pwn attempt 0:
Unexpected error in set_bt_name / bnep.connect: None None
152.587 PoC reproduced
-------------------------------------------
-------------------------------------------
0. Exit
1. 00:0D:18:A1:36:BD - Pumpkin live
2. C0:EE:FB:24:38:4E - OnePlus One live
3. F4:60:E2:E3:7E:55 - POCOPHONE live
4. 14:AB:C5:06:AC:38 - ATGRZWN114312 live
5. 90:61:AE:CA:48:DC - ATGRZWN311355 live
6. A4:6C:F1:FB:85:85 - Galaxy A8 (2018)
7. 90:A3:65:99:20:7A - Nokia 3310
8. 4C:49:E3:0A:1D:72 - Redmi
9. B0:6F:E0:04:D8:6D - Galaxy A8 (2018)
-------------------------------------------
Select the desired BDADDR: _
```

Holistic Testing

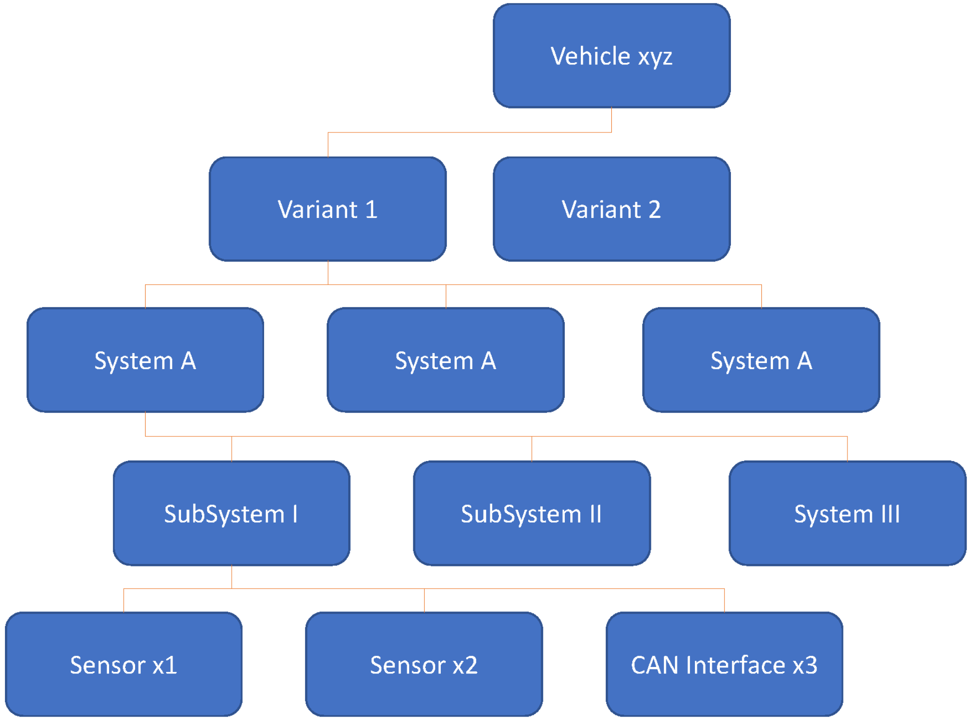
- System level
- Including architecture
- Conducted on the communications networks inside an actual system



System Reconnaissance

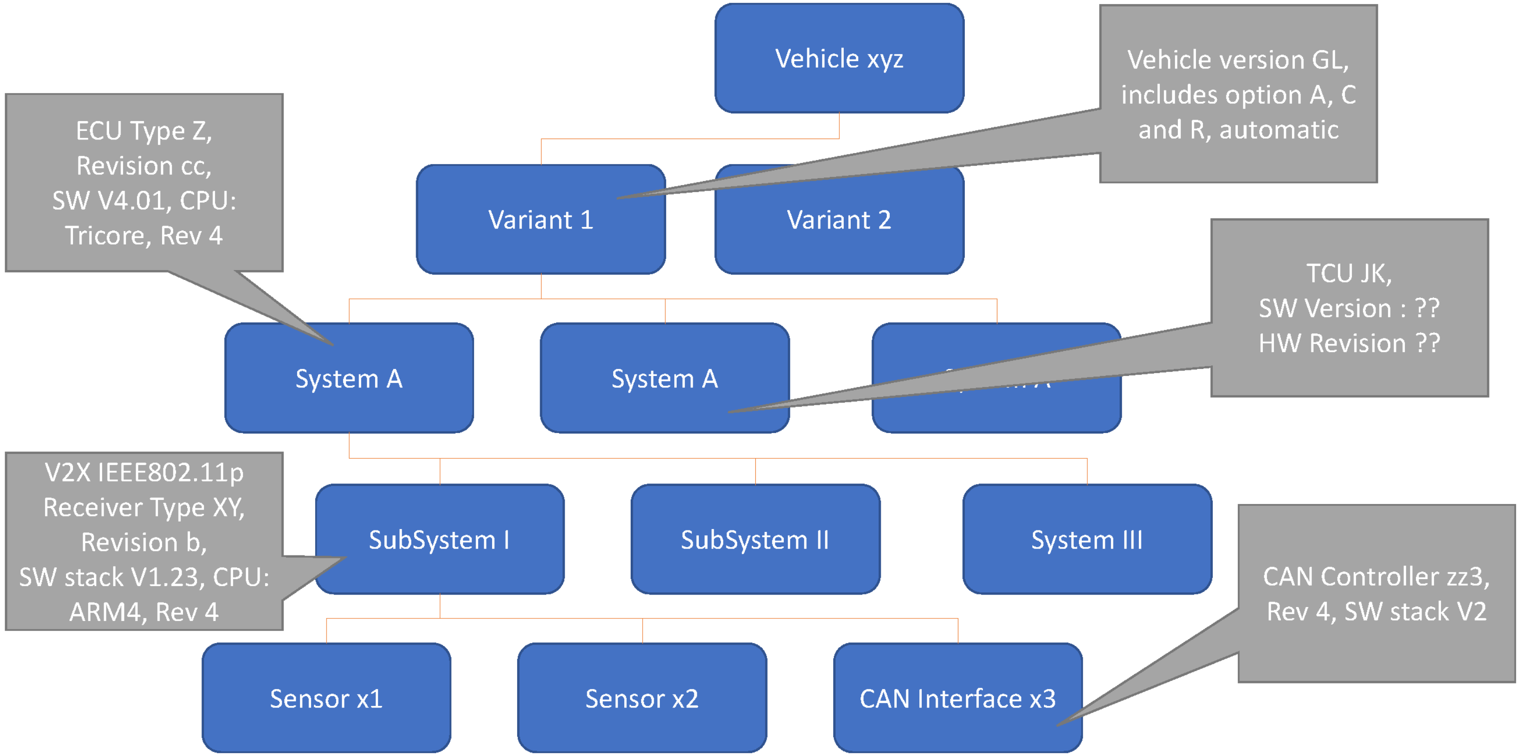
- Use a variety of interfaces
  - Wireless UIs (WiFi, BlueTooth,..)
  - Wired UIs (USB,)
  - Diagnostic (OBD)
  - Wiretapping (CAN, LIN)
- Active (sending messages)
- Passive (listening only)

- More complete picture of the SUT
- Ability to segment attacks



Fingerprinting

- Passively:
  - Deviation
  - Kurtosis
  - Clock skew
  - …
- Actively:
  - Sending (CAN) messages
    - Well formatted
    - Malformed
- Attribute a component according to the detecting interface



Vehicle xyz

Vehicle version GL, includes option A, C and R, automatic

Variant 1

Variant 2

ECU Type Z, Revision cc, SW V4.01, CPU: Tricore, Rev 4

System A

System A

TCU JK, SW Version : ?? HW Revision ??

V2X IEEE802.11p Receiver Type XY, Revision b, SW stack V1.23, CPU: ARM4, Rev 4

SubSystem I

SubSystem II

System III

CAN Controller zz3, Rev 4, SW stack V2

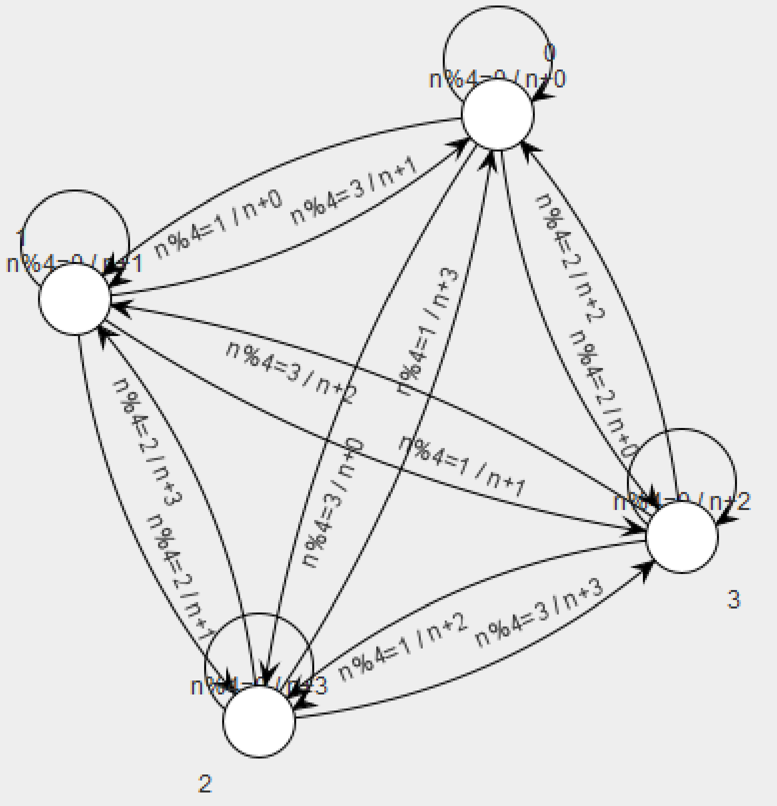
Sensor x1

Sensor x2

CAN Interface x3

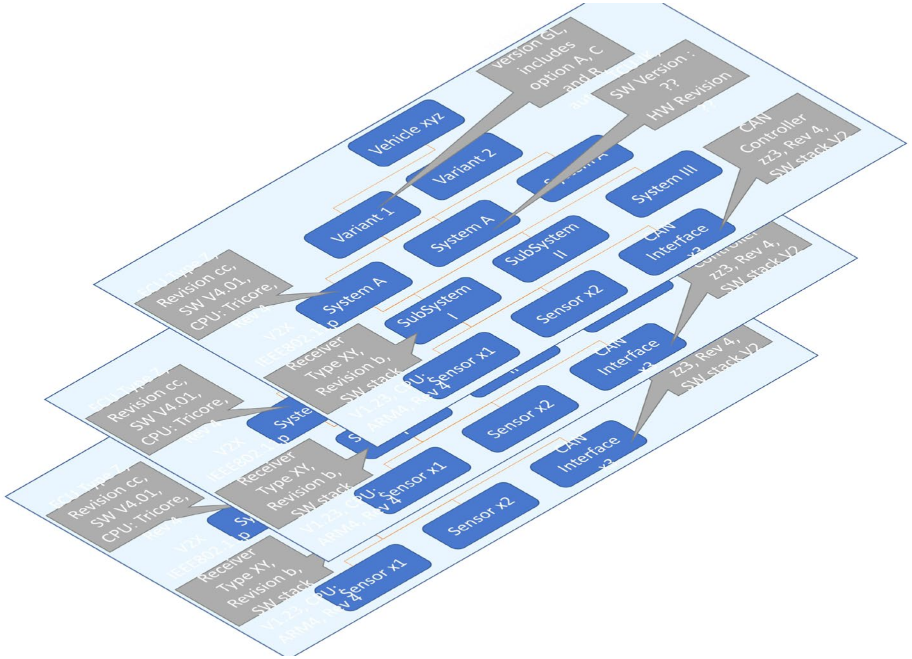
Model Generation

- Use (abstract) automata learning to learn a *behavioral* model
- Use model checking for test case generation



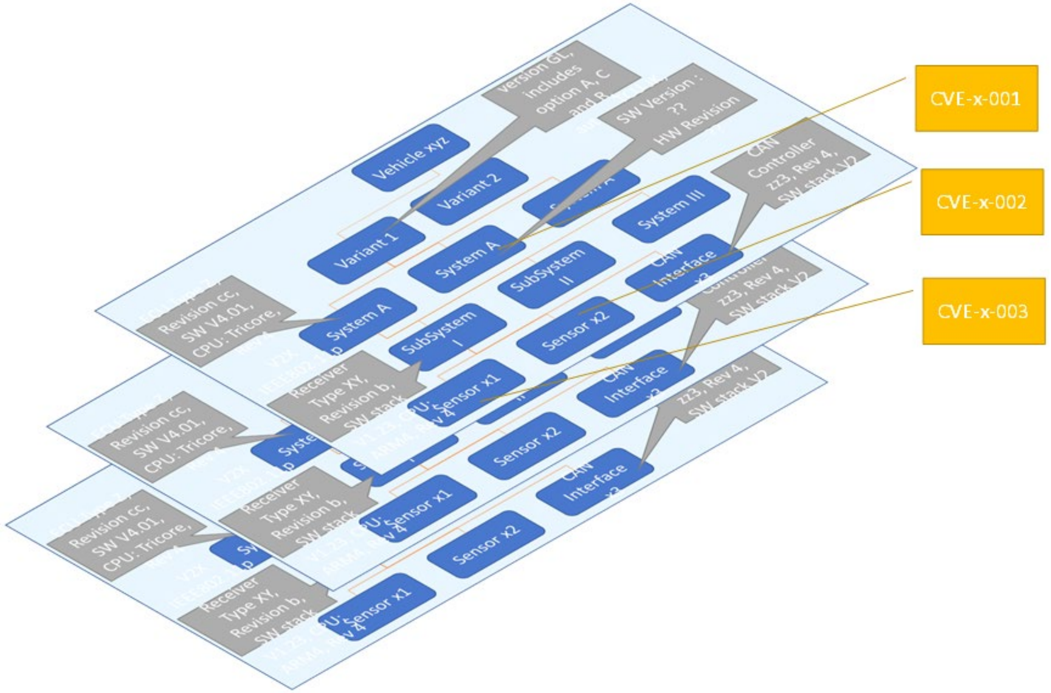
Variant Management

- Without a priori knowledge, a plethora of candidate models is available
- This set is narrowed down with every piece of information
- Each test case touches a number of assumed components, allowing for gathering data for fingerprinting
- Test cases will not only be chosen according to a potential attack vector, but also considering pivot elements to exclude or verify an optimal number of candidate models


Vehicle xyz
Variant 1
Variant 2
System A
System A
System III
SubSystem I
SubSystem II
Sensor x1
Sensor x2
CAN Interface x3
version GL, includes option A, C and B
SW Version : ?? HW Revision ??
CAN Controller zz3, Rev 4, SW stack V2
ECU Type Z, Revision cc, SW V4.01, CPU: Tricore, Rev 4
V2X IEEE802.11p
Receiver Type XY, Revision b, SW stack V1.23, CPU: ARM4, Rev 4

Attack Model

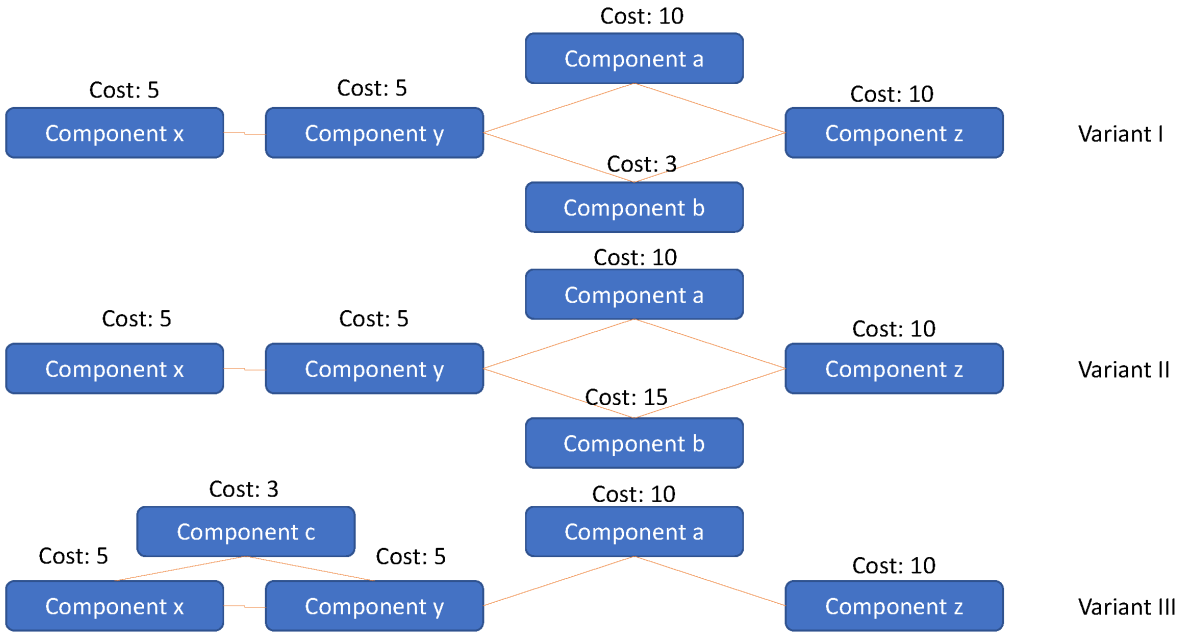
- Augmenting the system model with attack information
  - Using CVE information
  - Using other public sources (Auto-ISAC, research, darknet)
  - Analysis - see previous presentation of this workshop ;)
- Should occur both component and function-wise



Vehicle xyz
Variant 1
Variant 2
System A
System III
SubSystem II
SubSystem I
Sensor x1
Sensor x2
CAN Interface x2
CAN Controller zz3, Rev 4, SW stack V2
Revision cc, SW V4.01, CPU: Tricore,
Receiver Type XY, Revision b, SW stack
SW Version: ?? HW Revision
CVE-x-001
CVE-x-002
CVE-x-003

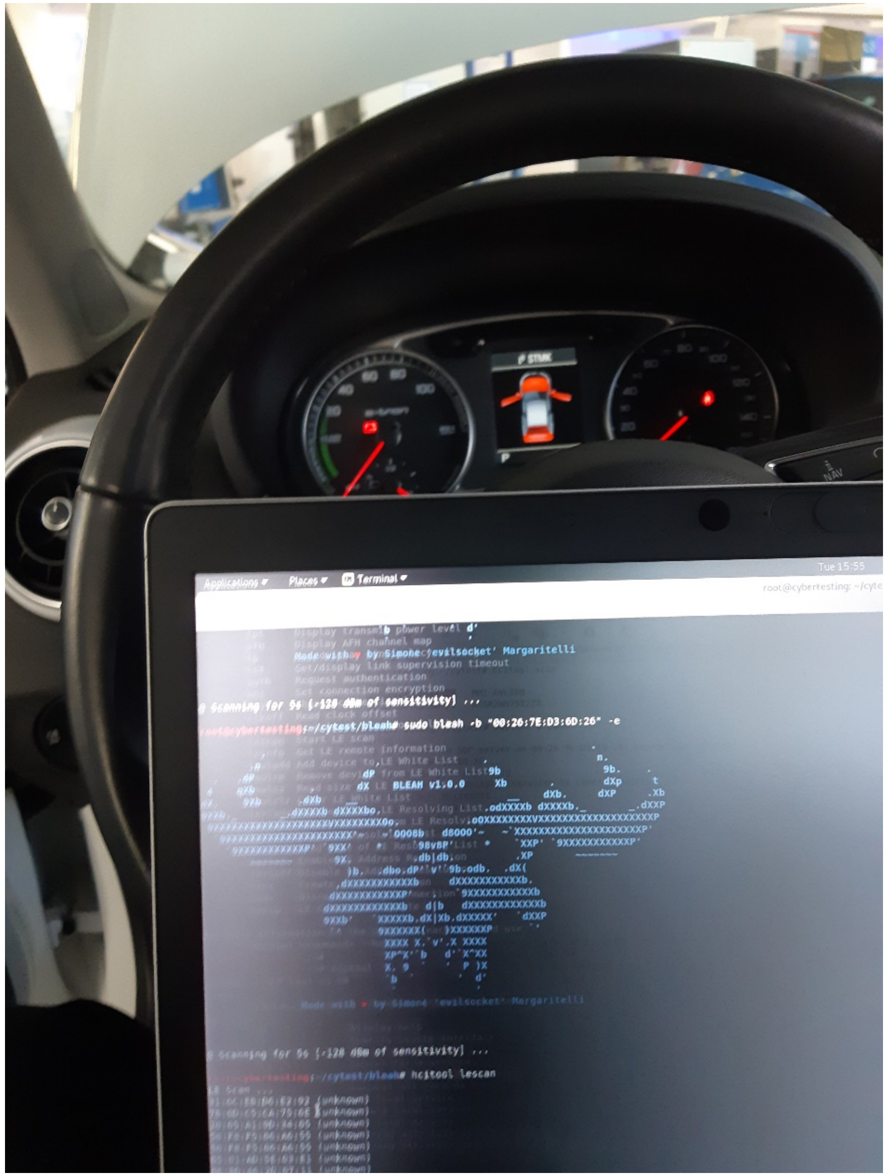
Attack Tree

- Assign cost to attacks on a specific component
- Shortest path (per cost) => most feasible attack
- Shortest path will be tested first, in conjunction with variant management considerations
- Test pass if the cost of the shortest path is above a certain threshold => *sufficiently secure*



Conclusion

- Concept for holistic zero-knowledge testing of automotive systems
- Combining fingerprinting and attack trees for test case generation
- Coping with variants that result from fuzziness



```
Made with by Simone 'evilsocket' Margaritelli
@ Scanning for 5s [-128 dBm of sensitivity] ...
root@cybertesting:~/cytest/bleah# sudo bleah -b "00:26:7E:D3:6D:26" -e
BLEAH v1.0.0
@ Scanning for 5s [-128 dBm of sensitivity] ...
root@cybertesting:~/cytest/bleah# hcitool lescan
LE Scan ...
```

Thank you for your attention!

Thanks!

Stefan Marksteiner[1], Peter Priller[2]

[1] Senior Technology Scout Cyber Security, AVL List Gmbh, stefan.marksteiner@avl.com

[2] Principal Technology Scout Embedded Systems, AVL List Gmbh, peter.priller@avl.com

This research has received funding from the program "ICT of the Future" of the Austrian Research Promotion Agency (FFG) and the Austrian Ministry for Transport, Innovation and Technology under grant agreement No. 867558 (project TRUSTED) and within the the ECSEL Joint Undertaking (JU) under grant agreement No. 876038 (project InSecTT). The JU receives support from the European Union's Horizon 2020 research and innovation programme and Austria, Sweden, Spain, Italy, France, Portugal, Ireland, Finland, Slovenia, Poland, Netherlands, Turkey. The document reflects the author's view only and the Commission is not responsible for any use that may be made of the information it contains.