Experimental Security Assessment of BMW Cars: A Summary Report

KEEN security lab

1. Introduction

In recent years, more and more BMW cars have been equipped with the new generation of "Internet-Connected" Infotainment system (e.g. HU_NBT/HU_ENTRYNAV) – a.k.a Head Unit – and the Telematics Control Unit (e.g. TCB). While these components have significantly improved the convenience and performance of customers' experience, they have also introduced the opportunity for new attacks.

In our work, we systematically performed an in-depth and comprehensive analysis of the hardware and software on Head Unit, Telematics Control Unit and Central Gateway Module of multiple BMW vehicles. Through mainly focusing on the various external attack surfaces of these units, we discovered that a remote targeted attack on multiple Internet-Connected BMW vehicles in a wide range of areas is feasible, via a set of remote attack surfaces (including GSM Communication, BMW Remote Service, BMW ConnectedDrive Service, UDS Remote Diagnosis, NGTP protocol, and Bluetooth protocol). Therefore, it's susceptible for an attacker to gain remote control to the CAN buses of a vulnerable BMW car by utilizing a complex chain of several vulnerabilities existed in different vehicle components. In addition, even without the capability of Internet-Connected, we are also able to compromise the Head Unit in physical access ways (e.g. USB, Ethernet and OBD-II). Based on our testing, we confirm that all the vulnerabilities would affect various modern BMW models.

Our research findings have proved that it is feasible to gain local and remote access to infotainment, T-Box components and UDS communication above certain speed of selected BMW vehicle modules and been able to gain control of the CAN buses with the execution of arbitrary, unauthorized diagnostic requests of BMW in-car systems remotely.

This report summaries our research findings in the structure below:

- Chapter 2-4 describes our vulnerability findings and attack chains in a controlled technical level without any in-depth technical details that can be utilized.
- Chapter 5 outlines the vulnerable BMW car models.
- Chapter 6 describes the disclosure process.
- Chapter 7 gives a conclusion about the research.

2. Research Description

From a security point of view, modern BMW Cars expose several remote attack surfaces, as well as physical ones. In this paper, we focused on three important vehicular components: Infotainment System (a.k.a Head Unit), Telematics Control Unit and Central Gateway Module, which are susceptible to be compromised from external attacks. Based on our research of BMW Car's in-vehicle network, we found all the three components working very closely with others through physical buses (e.g. USB, CAN Bus, Ethernet).

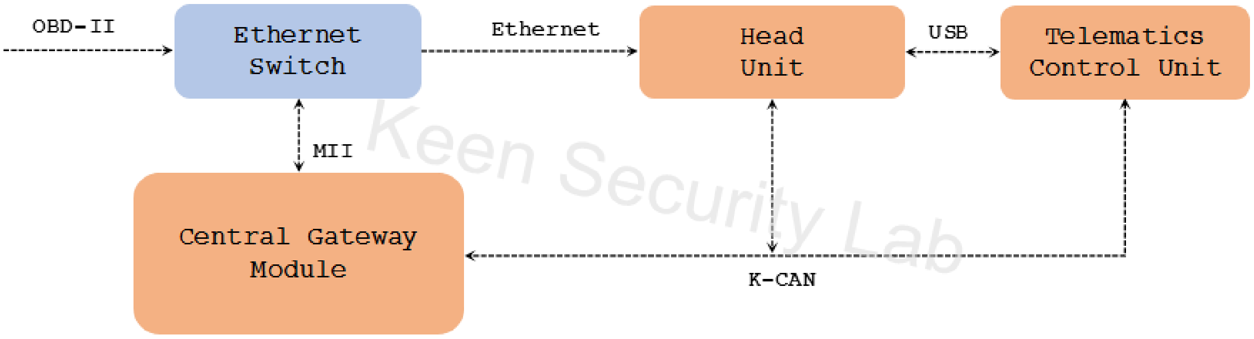


Figure: In-Vehicle Network of BMW Cars

After conducting an in-depth security analysis on firmware, we found 14 vulnerabilities in these vehicular components. All the software vulnerabilities we found can be fixed by online re-configuration and offline firmware update (not Over-The-Air upgrade).

Currently, BMW is in progress working on the mitigation plans, and some high priority countermeasures are already in the roll-out. Therefore we decided to make a brief vulnerabilities disclosure in this paper, instead of a full disclosure which would be considered as irresponsible to BMW users. However, the full technical vulnerabilities report will be released at a proper time in Year 2019.

2.1 Infotainment System

Figure: In-Vehicle Infotainment System of BMW i3

The in-vehicle infotainment system (a.k.a NBT Head Unit) in BWM Cars consists of two parts: hu-intel system and hu-jacinto system.

hu-intel. A QNX system running on the high-layer chip (Intel x86), mainly responsible for the multimedia service and BMW ConnectedDrive service.

hu-jacinto. A QNX system running on the Jacinto ARM chip, which is a low-layer chip for handling power management and CAN-bus communication.

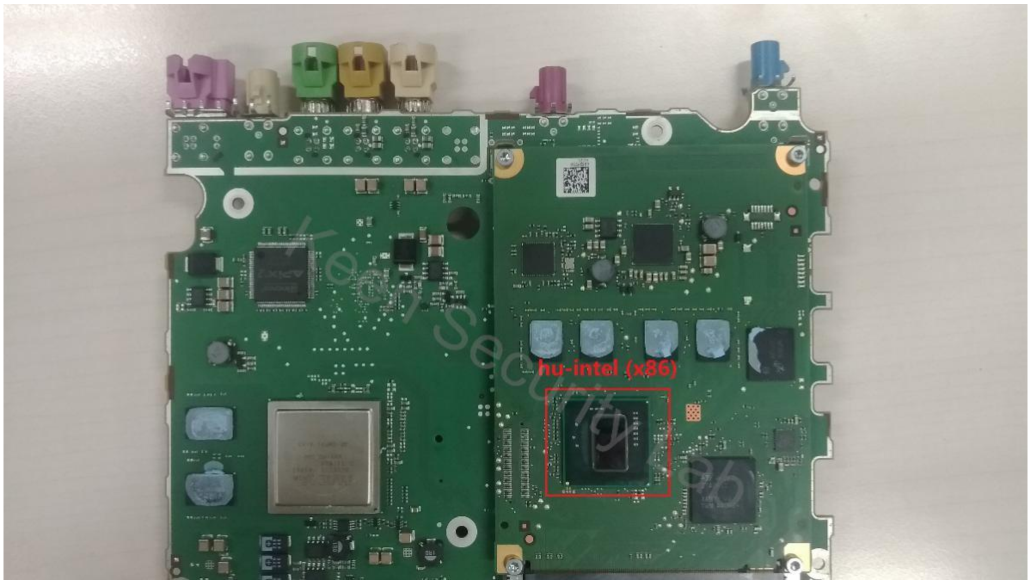


Figure: Mainboard of the High-Layer System (hu-intel) in NBT

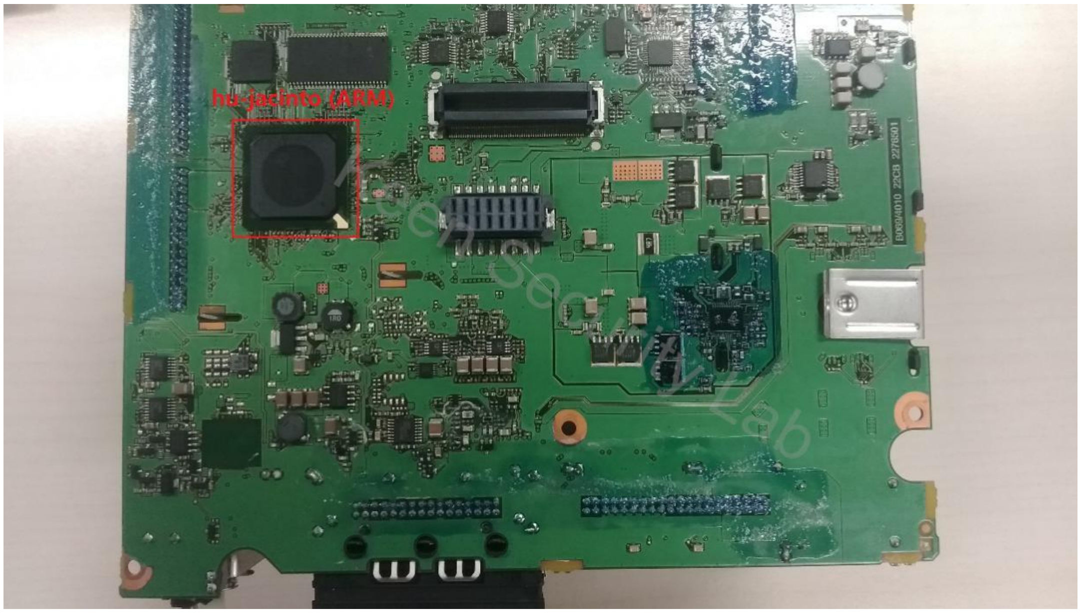


Figure: Mainboard of the Low-Layer System (hu-jacinto) in NBT

Both hu-intel and hu-jacinto can communicate with each other through QNET. The Telematics Control Unit is connected to the hu-intel through USB, where all the communication data between Head Unit and BMW Remote Server will be transmitted. Both hu-jacinto and Telematics Control Unit are connected to K-CAN Bus, which is a dedicated CAN bus for infotainment. And for secure isolation, the Ethernet connection from hu-intel to the Central Gateway Module is blocked by Ethernet Switch, in the newer BMW cars (e.g. BMW I3), both Central Gateway Module and Ethernet Switch are integrated into Body Controller Module (BDC/FEM).

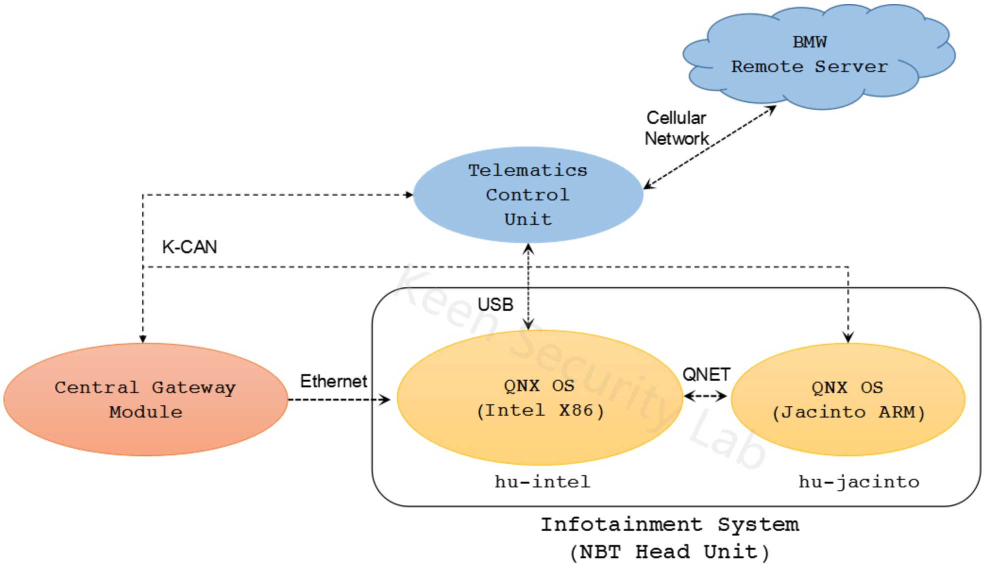
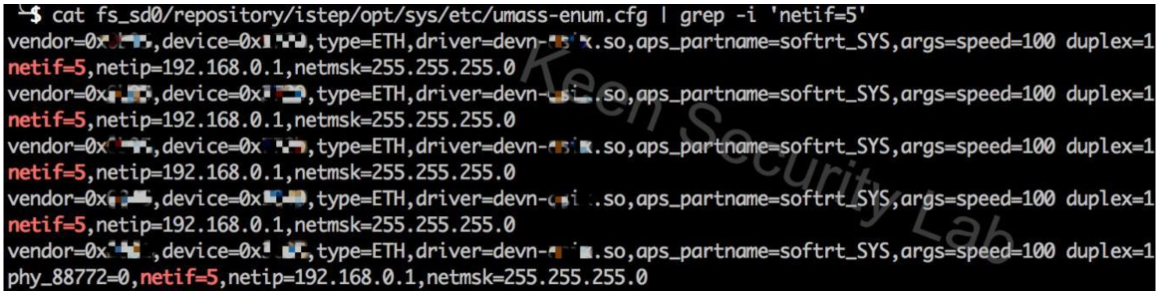


Figure: Architecture of NBT

2.1.1 USB Interface

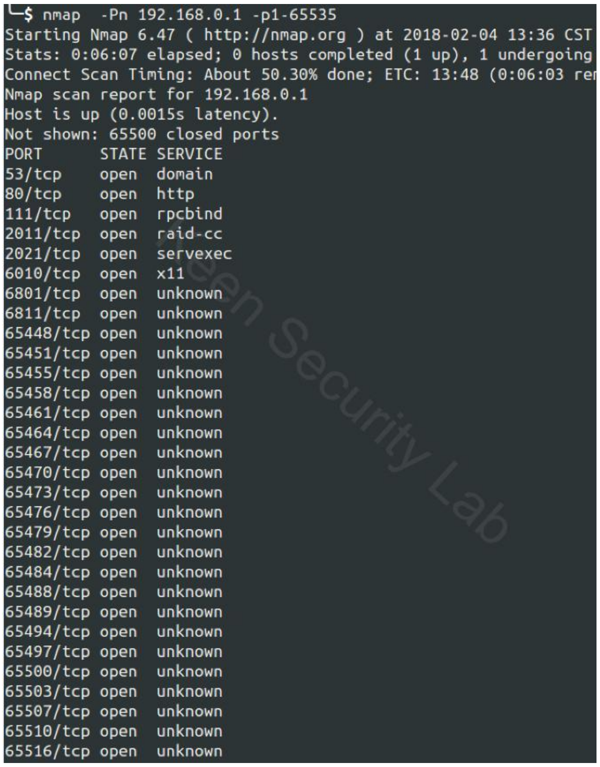
NBT provides some built-in io-pkt network drivers in QNX OS to set up an Ethernet network over USB interface. According to the USB configuration file (/opt/sys/etc/umass-enum.cfg) in hu-intel system, it supports some specific USB-to-ETHERNET adapters by default.



```
└$ cat fs_sd0/repository/istep/opt/sys/etc/umass-enum.cfg | grep -i 'netif=5'
vendor=0x[redacted],device=0x[redacted],type=ETH,driver=devn-[redacted].so,aps_partname=softrt_SYS,args=speed=100 duplex=1
netif=5,netip=192.168.0.1,netmsk=255.255.255.0
vendor=0x[redacted],device=0x[redacted],type=ETH,driver=devn-[redacted].so,aps_partname=softrt_SYS,args=speed=100 duplex=1
netif=5,netip=192.168.0.1,netmsk=255.255.255.0
vendor=0x[redacted],device=0x[redacted],type=ETH,driver=devn-[redacted].so,aps_partname=softrt_SYS,args=speed=100 duplex=1
netif=5,netip=192.168.0.1,netmsk=255.255.255.0
vendor=0x[redacted],device=0x[redacted],type=ETH,driver=devn-[redacted].so,aps_partname=softrt_SYS,args=speed=100 duplex=1
netif=5,netip=192.168.0.1,netmsk=255.255.255.0
vendor=0x[redacted],device=0x[redacted],type=ETH,driver=devn-[redacted].so,aps_partname=softrt_SYS,args=speed=100 duplex=1
phy_88772=0,netif=5,netip=192.168.0.1,netmsk=255.255.255.0
```

Figure: USB-Ethernet Network Configuration

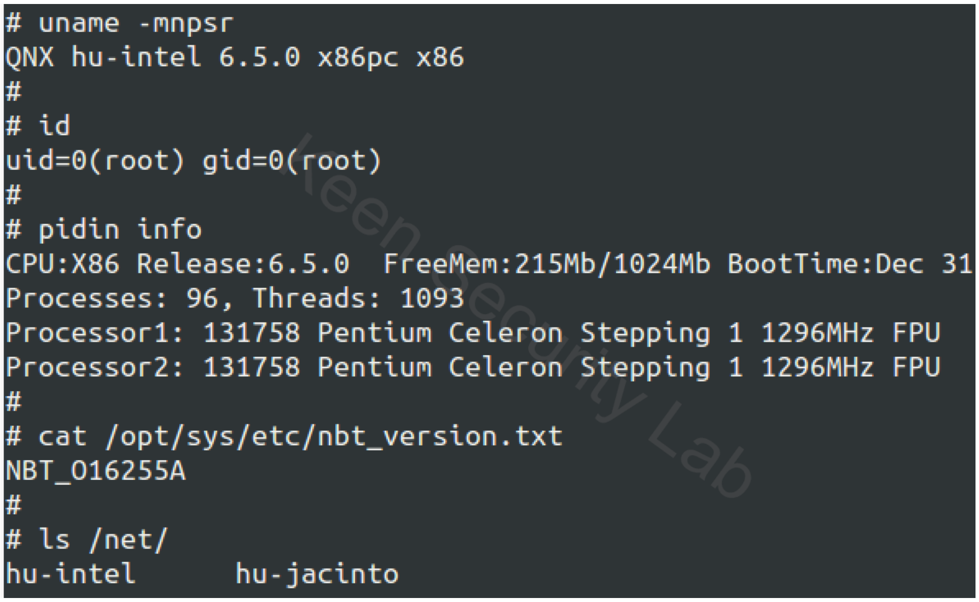
With network drivers, the USB-Ethernet network will be enabled when a USB dongle with some specific chipsets plugged in. NBT head unit will act as a network gateway with a fixed IP address (192.168.0.1). What’s worse, there aren’t any security restrictions to such USB Ethernet Interface, which makes it possible to obtain access to the internal network of the head unit, and then detect many exposed internal services through port scanning.



```
└$ nmap  -Pn 192.168.0.1 -p1-65535
Starting Nmap 6.47 ( http://nmap.org ) at 2018-02-04 13:36 CST
Stats: 0:06:07 elapsed; 0 hosts completed (1 up), 1 undergoing
Connect Scan Timing: About 50.30% done; ETC: 13:48 (0:06:03 re
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
65482/tcp open  unknown
65484/tcp open  unknown
65488/tcp open  unknown
65489/tcp open  unknown
65494/tcp open  unknown
65497/tcp open  unknown
65500/tcp open  unknown
65503/tcp open  unknown
65507/tcp open  unknown
65510/tcp open  unknown
65516/tcp open  unknown
```

Figure: Port Scanning in the Internal Network

Local Code Execution. There are several update services running in hu-intel system (e.g. Navigation Update / Software Update) and monitoring the USB stick. With the expected update content provided in the USB stick, NBT will fall into certain upgrade stage. Some content is signed by BMW private keys, while some are not, which gives us a chance to prepare our malformed content in the USB stick and leverage some vulnerabilities existed in the update service to gain control of hu-intel system with root privilege.



```
# 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
```

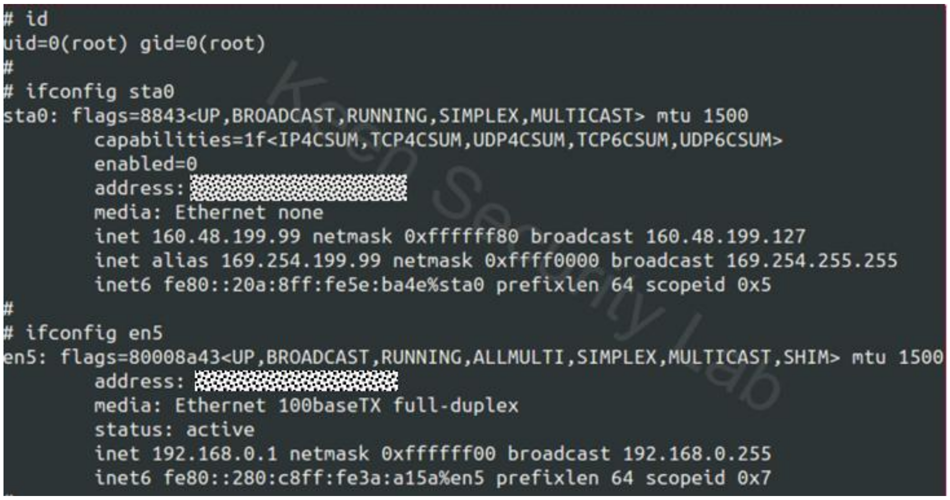
Figure: Root Shell from NBT

There's another way to get a root shell which will be explained later.

2.1.2 E-NET over OBD-II

The E-NET is an in-vehicle Ethernet network hosted on OBD-II Interface in BMW Cars. Via the E-NET, the automotive engineer can connect to the Central Gateway, and conduct offline diagnoses and firmware update for the head unit. Correspondingly, in hu-intel system, a peer diagnose service is responsible for handling requests from Central Gateway. After reverse-engineering the diagnose protocol between Central Gateway and NBT, we found the vulnerability that can be utilized to bypass the code sign mechanism, and successfully gained a root shell from the hu-intel system.

Furthermore, consider a lower-cost way: by using a USB-Ethernet dongle, hackers can also root the high-layer QNX system (hu-intel) which has a fixed IP address (192.168.0.1) on the en5 interface.

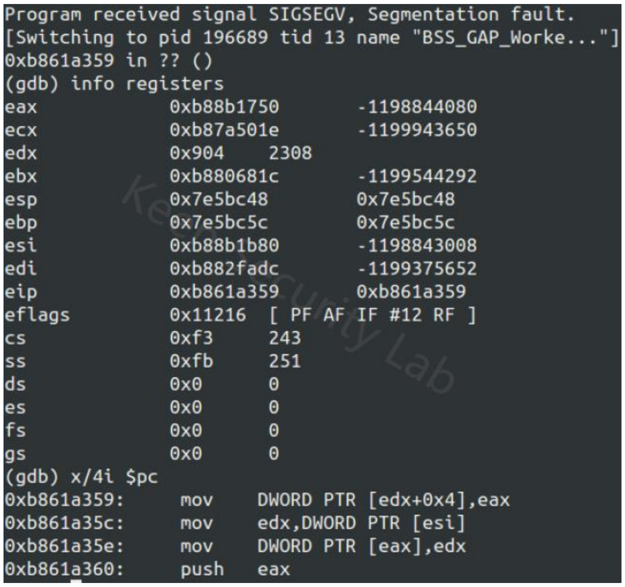


```
# id
uid=0(root) gid=0(root)
#
# ifconfig sta0
sta0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
        capabilities=1f<IP4CSUM,TCP4CSUM,UDP4CSUM,TCP6CSUM,UDP6CSUM>
        enabled=0
        address: [redacted]
        media: Ethernet none
        inet 160.48.199.99 netmask 0xffffff80 broadcast 160.48.199.127
        inet alias 169.254.199.99 netmask 0xffff0000 broadcast 169.254.255.255
        inet6 fe80::20a:8ff:fe5e:ba4e%sta0 prefixlen 64 scopeid 0x5
#
# ifconfig en5
en5: flags=80008a43<UP,BROADCAST,RUNNING,ALLMULTI,SIMPLEX,MULTICAST,SHIM> mtu 1500
        address: [redacted]
        media: Ethernet 100baseTX full-duplex
        status: active
        inet 192.168.0.1 netmask 0xffffff00 broadcast 192.168.0.255
        inet6 fe80::280:c8ff:fe3a:a15a%en5 prefixlen 64 scopeid 0x7
```

Figure: Ethernet Configuration in NBT Head Unit

2.1.3 Bluetooth Stack

With the built-in Bluetooth capability, NBT allows mobile phones to connect to it for hands-free calling. Since we have gained access to the hu-intel system with root privilege, then we identified a particular service responsible for handling Bluetooth functionality. Through reverse engineering, we figured that it uses a third-party Bluetooth stack library, which is an implementation of the management and services component of the Bluetooth stack. After fuzzing the Bluetooth stack by sending malformed packages to the head unit, finally we got a malformed package that can lead to the memory corruption in Bluetooth stack.



```
Program received signal SIGSEGV, Segmentation fault.
[Switching to pid 196689 tid 13 name "BSS_GAP_Worke..."]
0xb861a359 in ?? ()
(gdb) info registers
eax            0xb88b1750       -1198844080
ecx            0xb87a501e       -1199943650
edx            0x904    2308
ebx            0xb880681c       -1199544292
esp            0x7e5bc48        0x7e5bc48
ebp            0x7e5bc5c        0x7e5bc5c
esi            0xb88b1b80       -1198843008
edi            0xb882fadc       -1199375652
eip            0xb861a359       0xb861a359
eflags         0x11216  [ PF AF IF #12 RF ]
cs             0xf3     243
ss             0xfb     251
ds             0x0      0
es             0x0      0
fs             0x0      0
gs             0x0      0
(gdb) x/4i $pc
0xb861a359:     mov    DWORD PTR [edx+0x4],eax
0xb861a35c:     mov    edx,DWORD PTR [esi]
0xb861a35e:     mov    DWORD PTR [eax],edx
0xb861a360:     push   eax
```

Figure: Memory Corruption in Bluetooth Service

As a result, by simply setting NBT into the pairing mode, we can utilize the vulnerability to crash Bluetooth stack in the hu-intel system without PIN code. Consequently, causing the head unit to reboot, due to the internal watchdog mechanism.

2.1.4 ConnectedDrive Service

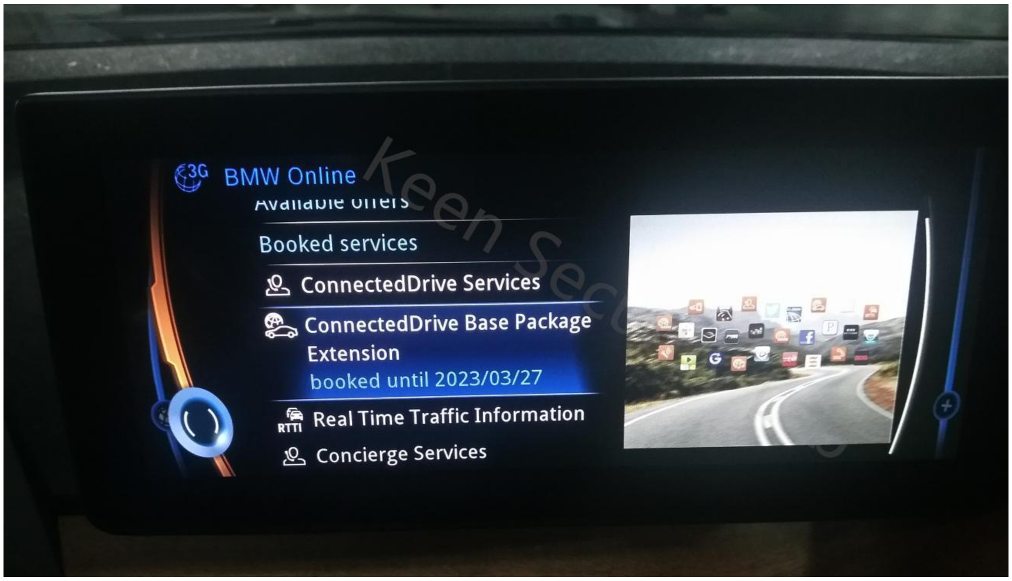
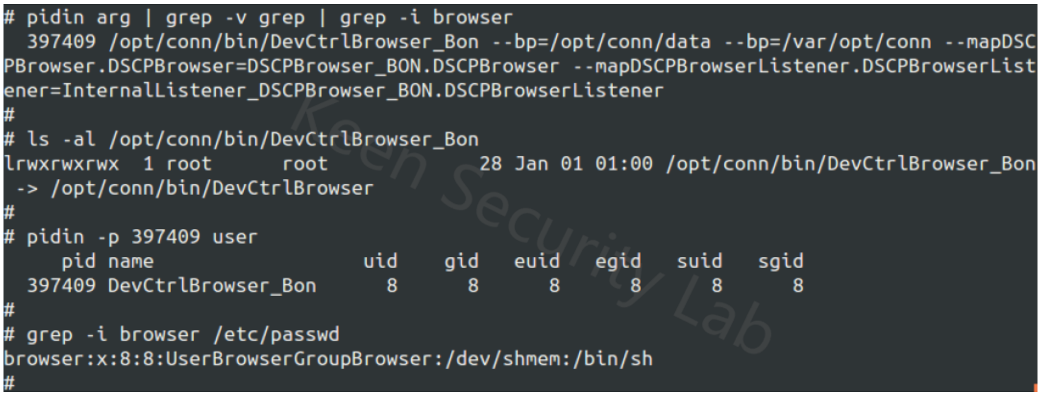


Figure: BMW ConnectedDrive Online Service in NBT Head Unit

BMW ConnectedDrive service in NBT uses a cellular connection via an embedded SIM card built into the Telematics Control Unit to offer customers a wide range of useful online features, including ConnectedDrive Store, TeleServices, Real Time Traffic Information (RTTI), Intelligent Emergency Call, Online Weather and Online News. Most of online features provided by ConnectedDrive service are processed by an in-vehicle browser, so-called "DevCtrlBrowser_Bon" in NBT. The "DevCtrlBrowser_Bon" uses a customized browser engine. It seems to be developed by Harman for in-vehicle infotainment system.



```
# 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
#
```

Figure: Process Information of "DevCtrlBrowser_Bon"

Remote Code Execution. After we implemented a stable GSM network using universal software radio peripheral (USRP) and OpenBTS, all the traffic from ConnectedDrive service were captured, and due to some insecure implementations of ConnectedDrive service in NBT, we also succeeded in intercepting network traffic from ConnectedDrive Service. After that, we were free to find the bugs in “DevCtrlBrowser_Bon”. Then we exploited a memory corruption bug in “DevCtrlBrowser_Bon” and achieved remote code execution in the head unit with browser privilege. In the end, by leveraging the vulnerability mentioned earlier, we achieved root privilege escalation and got a remote root shell from NBT through a different path than 2.1.1.

2.1.5 K-CAN Bus

After accessing to the high-layer QNX System (hu-intel) by exploiting above vulnerabilities, we can also login into the low-level hu-jacinto system via QNET. As mentioned above, hu-jacinto system is running on Jacinto ARM chip and responsible for handling CAN messages. Through in-depth analysis, we figured out two approaches to send arbitrary CAN messages on K-CAN Bus:

(1) Though the datasheet is not open to the public, we can reuse some CAN-bus driver’s source code from BSP project developed by TI to operate the special memory of Jacinto chip to send the CAN messages.

(2) By dynamically hooking the CAN-bus driver’s function which is used to transmit CAN messages, we are also able to stably send arbitrary CAN messages to K-CAN bus.

By chaining the vulnerabilities together, we are able to remotely compromise the NBT. After that, we can also leverage some special remote diagnose interfaces implemented in the Central Gateway Module to send arbitrary diagnostic messages (UDS) to control ECUs on different CAN Buses.

2.2 Telematics Control Unit

Telematics Control Unit provides BMW connected vehicles with telephone functions and telematics service (e.g. E-Call, B-Call, etc.,) via cellular networks as well as BMW remote services (e.g. remote unlock, climate control, etc..). In this section, we target the Telematic Communication Box (TCB) control unit produced by “Peiker Acustic GmbH” from Germany, which is the most widely used Telematics Control Unit that has been equipped with NBT and ENTRYNAV head units in modern BMW cars.

Hardware Architecture. The TCB control unit can be divided into two parts, the high-layer part is the MPU, based on a Qualcomm MDM6200 baseband processor with an AMSS RTOS (REX OS). With an Embedded-SIM card, MDM6200 is responsible for the telematics communication between TCB and BMW remote server. The low-layer part is the MCU, a CAN controller based on Freescale 9S12X, which is directly connected to Central Gateway Module through K-CAN bus. The MPU (MDM6200 baseband) uses UART-based IPC mechanism to communicate with the MCU (Freescale 9S12X).

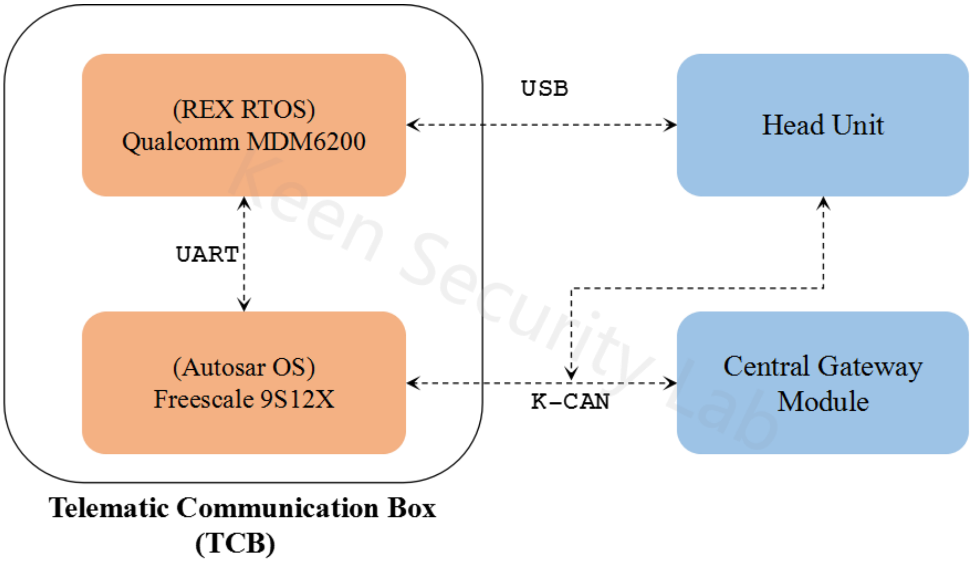


Figure: Hardware Architecture of TCB

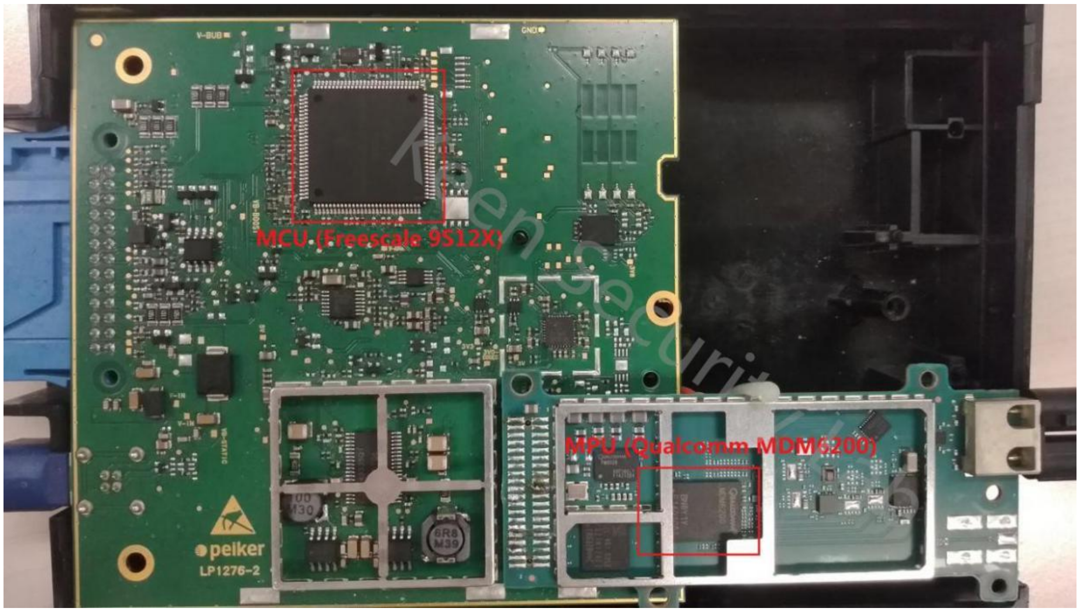
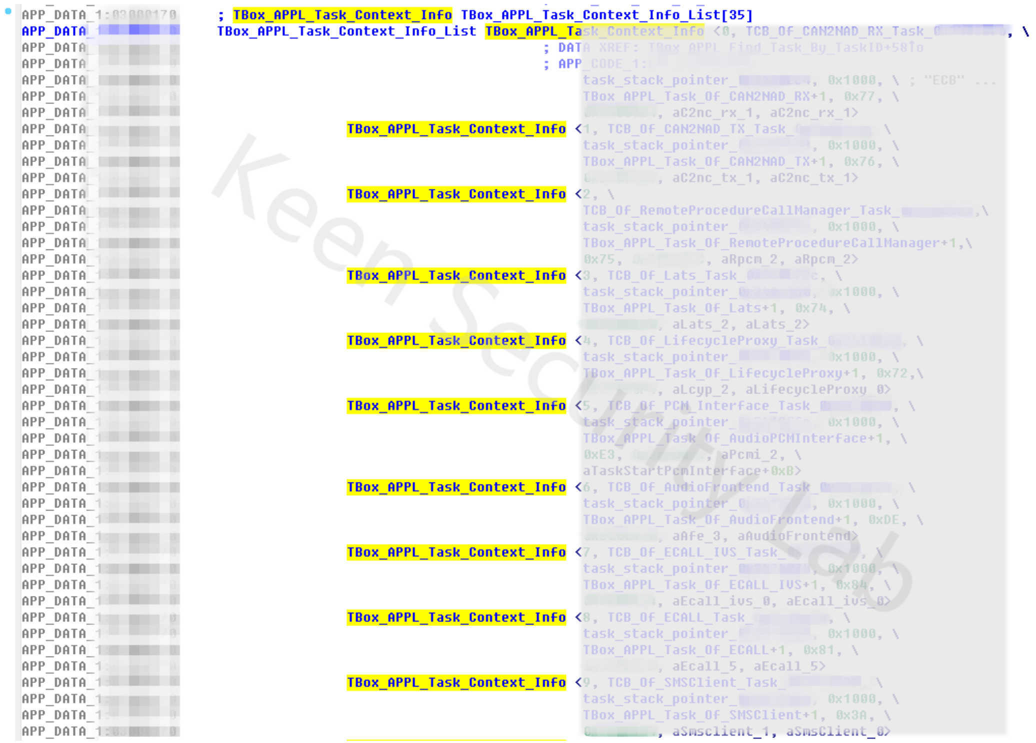


Figure: Mainboard of TCB

Software Architecture. The Telematic Communication Box(TCB) control unit is one of the platforms for the functions of BMW ConnectedDrive System. TCB can establish a connection to the GSM and UMTS networks, during our reverse-engineering the firmware, we found the TCB would support following functionalities:

1. Enhanced emergency call
2. BMW Remote Service (e.g. remote door unlocking, climate control, etc...)
3. BMW TeleService diagnosis, including TeleService help
4. BMW TeleService Call
5. BMW LastStateCall
6. Others

The REX OS (Real-Time Executive Operating System) developed by Qualcomm for the ARM-based Advanced Mode Subscriber Software (AMSS) is running on the TCB's MPU. There are more than 60 system tasks ("CallManager", "Diag_task", "Voice", "GPRS LLC", etc..), as well as about 34 application tasks ("NGTPD", "NAD Diag", "SMSClient", "LastStateCall", etc..) are working for the multiple functions mentioned above.



```
; TBox_APPL_Task_Context_Info TBox_APPL_Task_Context_Info_List[35]
TBox_APPL_Task_Context_Info_List TBox_APPL_Task_Context_Info <0, TCB_Of_CAN2NAD_RX_Task_[redacted], \
                                        ; DATA XREF: TBox_APPL_Find_Task_By_TaskID+58↑o
                                        ; APP_CODE_1:[redacted]
                                         task_stack_pointer_[redacted], 0x1000, \ ; "ECB" ...
                                         TBox_APPL_Task_Of_CAN2NAD_RX+1, 0x77, \
                                         [redacted], aC2nc_rx_1, aC2nc_rx_1>
                TBox_APPL_Task_Context_Info <1, TCB_Of_CAN2NAD_TX_Task_[redacted] \
                                         task_stack_pointer_[redacted], 0x1000, \
                                         TBox_APPL_Task_Of_CAN2NAD_TX+1, 0x76, \
                                         [redacted], aC2nc_tx_1, aC2nc_tx_1>
                TBox_APPL_Task_Context_Info <2, \
                                         TCB_Of_RemoteProcedureCallManager_Task_[redacted],\
                                         task_stack_pointer_[redacted], 0x1000, \
                                         TBox_APPL_Task_Of_RemoteProcedureCallManager+1,\
                                         0x75, [redacted], aRpcm_2, aRpcm_2>
                TBox_APPL_Task_Context_Info <3, TCB_Of_Lats_Task_[redacted], \
                                         task_stack_pointer_[redacted], 0x1000, \
                                         TBox_APPL_Task_Of_Lats+1, 0x74, \
                                         [redacted], aLats_2, aLats_2>
                TBox_APPL_Task_Context_Info <4, TCB_Of_LifecycleProxy_Task_[redacted], \
                                         task_stack_pointer_[redacted], 0x1000, \
                                         TBox_APPL_Task_Of_LifecycleProxy+1, 0x72,\
                                         [redacted], aLcyp_2, aLifecycleProxy_0>
                TBox_APPL_Task_Context_Info <5, TCB_Of_PCM_Interface_Task_[redacted], \
                                         task_stack_pointer_[redacted], 0x1000, \
                                         TBox_APPL_Task_Of_AudioPCMInterface+1, \
                                         0xE3, [redacted], aPcmi_2, \
                                         aTaskStartPcmInterface+0xB>
                TBox_APPL_Task_Context_Info <6, TCB_Of_AudioFrontend_Task_[redacted], \
                                         task_stack_pointer_[redacted], 0x1000, \
                                         TBox_APPL_Task_Of_AudioFrontend+1, 0xDE, \
                                         [redacted], aAfe_3, aAudioFrontend>
                TBox_APPL_Task_Context_Info <7, TCB_Of_ECALL_IVS_Task_[redacted], \
                                         task_stack_pointer_[redacted], 0x1000, \
                                         TBox_APPL_Task_Of_ECALL_IVS+1, 0x84, \
                                         [redacted], aEcall_ivs_0, aEcall_ivs_0>
                TBox_APPL_Task_Context_Info <8, TCB_Of_ECALL_Task_[redacted], \
                                         task_stack_pointer_[redacted], 0x1000, \
                                         TBox_APPL_Task_Of_ECALL+1, 0x81, \
                                         [redacted], aEcall_5, aEcall_5>
                TBox_APPL_Task_Context_Info <9, TCB_Of_SMSClient_Task_[redacted] \
                                         task_stack_pointer_[redacted], 0x1000, \
                                         TBox_APPL_Task_Of_SMSClient+1, 0x3A, \
                                         [redacted], aSmsclient_1, aSmsClient_0>
```

Figure: Code Snippet of Application Tasks in TCB's Firmware

2.2.1 Remote Service with NGTP

The Next Generation Telematics Patterns (NGTP) is a technology-neutral telematics approach that aims to provide greater flexibility and scalability to the automotive, telematics and in-vehicle technology industries to offer better connectivity for drivers, passengers, and the vehicle itself. Functionalities such as BMW remote services, bmwinfo and myinfo in BMW vehicles are provided by NGTP. According to the original design, some NGTP messages should be transferred to TCB by HTTPS. After reverse engineering the firmware, we found that it's valid to directly send arbitrary NGTP messages through SMS to trigger various telematics functionalities as equal as through HTTPS, and the encryption/signature algorithms are known to public, also the encryption keys are hardcoded.

After some in-depth research, we completely restored the NGTP protocol and used USRP and OpenBTS to simulate a GSM network, then suppressed the TSP signals with a signal suppressor to make the BMW vehicle serviced by our rouge base station. Finally, we can directly send arbitrary NGTP messages to the BMW vehicles to trigger BMW Remote Services.

Notes: All these tests were performed in a lab environment for research purpose. DO NOT ATTEMPT THIS IN PUBLIC AREAS.

2.2.2 Remote Diagnosis

In the TCB's firmware, the "LastStateCall" task is responsible for remote diagnosis and diagnostic data gathering. Once the "LastStateCall" task has started, the function "LscDtgtNextJob" is invoked to extract diagnostic CAN messages (UDS) from a global buffer in the firmware, and then send diagnostic messages to the Central Gateway through K-CAN Bus. The Central Gateway will transfer these diagnostic messages to the target ECUs on different CAN Buses, and finally TCB will be notified to upload the corresponding response data from the target ECUs to the BMW remote server through HTTPS.

Remote Code Execution. After some tough reverse-engineering work on TCB's firmware, we also found a memory corruption vulnerability that allows us to bypass the signature protection and achieve remote code execution in the firmware. Till now, we were able to remotely root the TCB without any user interaction and send arbitrary diagnostic messages to control ECUs on CAN Buses like PT-CAN, K-CAN, etc...

2.3 Central Gateway Module

For different design purposes, the Central Gateway Module of BMW cars is integrated into different units (e.g. FEM and BDC). In the older series, ZGW – a standalone gateway ECU – is the Central Gateway Module of the in-vehicle network. In the newer series (e.g. BMW I3), the

Central Gateway Module is integrated into some body controller modules (e.g. BDC and FEM). We chose ZGW and BDC as our research targets which represent two generations Central Gateway Module of BMW cars.

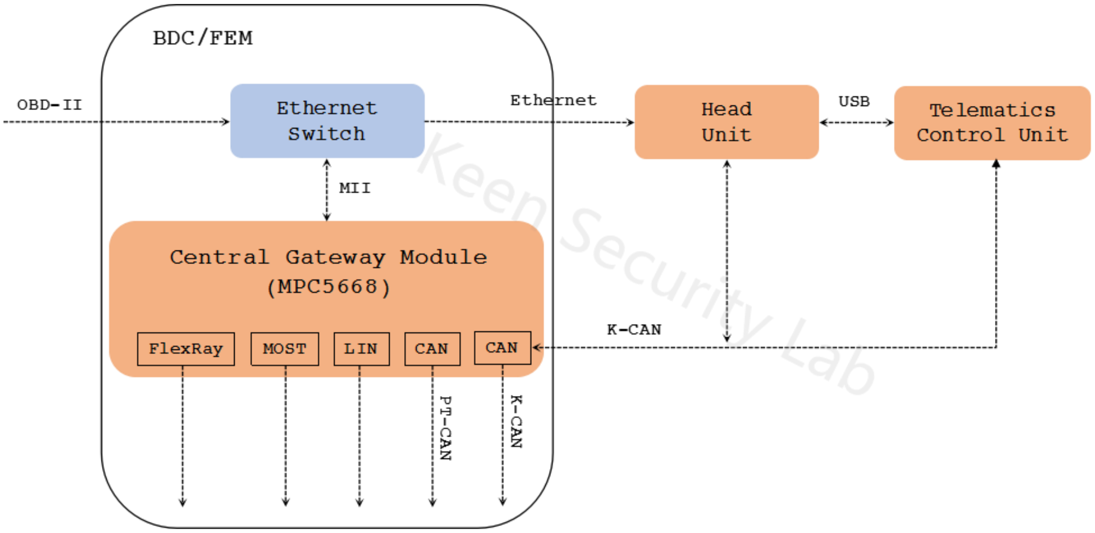


Figure: Central Gateway Module Based on MPC5668 Chip

The Central Gateway Module consists of a customized MPC5668 chip which is the PowerPC architecture. It’s connected to some CAN buses, as well as LIN, FlexRay and MOST buses. The most important feature of Central Gateway is to receive remote diagnostic CAN messages (UDS) from both the Telematics Control Unit and Head Unit, and then transfer diagnostic messages to other ECUs on different CAN buses.

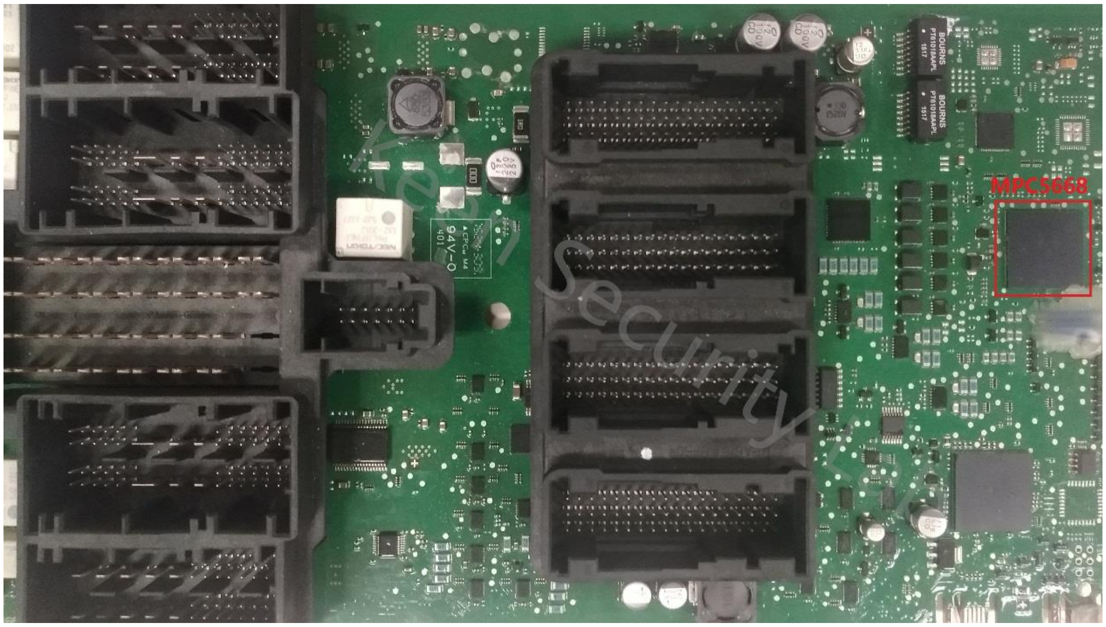


Figure: Mainboard of Central Gateway Module

2.3.1 Cross-Domain Diagnostic Messages

During the research, with the remote diagnostic features in the Central Gateway Module, we could leverage the remote diagnostic feature in the Central Gateway to send UDS messages to other ECUs. While in normal situations there's no danger when the Central Gateway Module processes the legal remote diagnostic messages from Telematics Control Unit or Head Unit, this feature can be a big issue which provides the hacker a potential attack surface to control other ECUs and break the secure isolation of different domains. Considering that we have remotely controlled the Telematics Control Unit and Head Unit, it's easy for us to make the Central Gateway Module transfer controlled diagnostic messages to manipulate ECUs on CAN Buses (e.g. PT-CAN, K-CAN, etc..) at most of the time.

2.3.2 Lack of High Speed Limit on UDS

A secure diagnostic function should be designed properly to avoid the incorrect usage at an abnormal situation. However, we found that most of the ECUs still respond to the diagnostic messages even at normal driving speed **(confirmed on BMW i3)**, which could cause serious security issues already. It will become much worse if attackers invoke some special UDS routines (e.g. reset ECU, etc..).

3. Vulnerability Findings

All the following vulnerabilities and CVEs have been confirmed by submitted by BMW after we provided the full report and collaborated with them on technical details:

No.	Vulnerability Description	Access	Affected Components	Reference
1	All the detail information has been reserved due to security concerns.	Local (USB)	HU_NBT	CVE-2018-9322
2		Local (USB/OBD)	HU_NBT	
3		Remote	HU_NBT	Logic Issue
4		Remote	HU_NBT	Reserved
5		Local (USB)	HU_NBT	CVE-2018-9320
6		Local (USB)	HU_NBT	CVE-2018-9312
7		Remote (Bluetooth)	HU_NBT	CVE-2018-9313
8		Physical	HU_NBT	CVE-2018-9314
9		Physical	TCB	Reserved
10		Remote	TCB	Logic Issue
11		Remote	TCB	CVE-2018-9311
12		Remote	TCB	CVE-2018-9318
13		Indirect Physical	BDC/ZGW	Logic Issue
14		Indirect Physical	BDC/ZGW	Logic Issue

Table: Vulnerabilities and CVEs in Our Research Confirmed by BMW

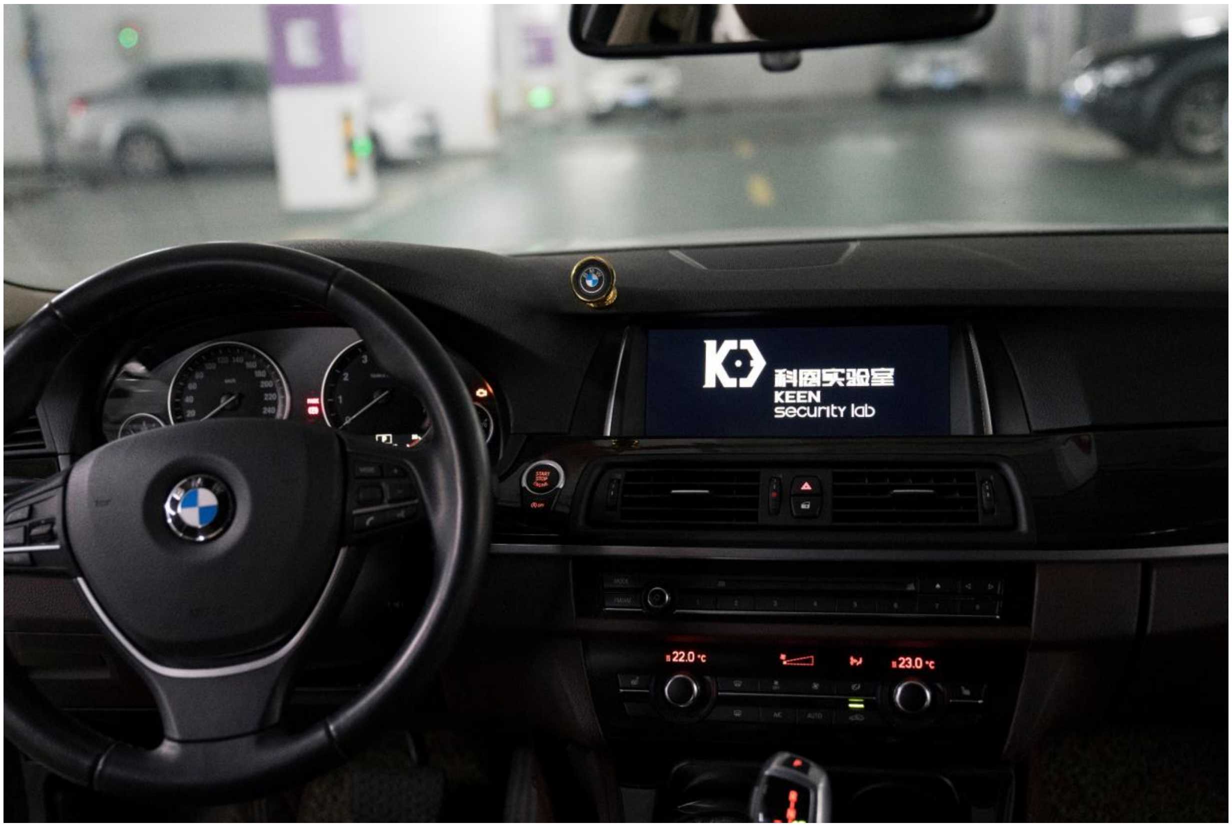


Figure: Example of control of the Infotainment System

4. Attack Chains

After we discovered a series of vulnerabilities mainly in different vehicle components in a modern BMW car, we still want to evaluate the effects of these vulnerabilities in a real-world scenario and try to figure out the potential dangers.

In our research, we have already found some ways to influence the vehicle via different kinds of attack chains by sending arbitrary diagnostic messages to the ECUs.

The attack chains are aimed to implement an arbitrary diagnostic message transmission to other CAN Buses through the Central Gateway Module in order to impact or control the ECUs on different CAN Buses, since we're able to send diagnostic messages in both NBT and TCB.

All the attack chains could be classified into two types: contacted attack and contactless attack. We do believe these attack chains could be utilized by skilled attackers at a very low cost – with enough research. The following subsections will explain the attack chains emphasized on the steps before we are able to send UDS messages on K-CAN bus.

4.1 Contacted Attack

In the real scenes, there is still a lot of situations that people have a chance to touch NBT, so the contacted attack is still a high potential attack method that should be paid attention to.

With the help of serious vulnerabilities over USB interface and OBD-II interface, attackers can easily use them to install the backdoor in the NBT, and then manipulate the vehicle functions through Central Gateway Module.

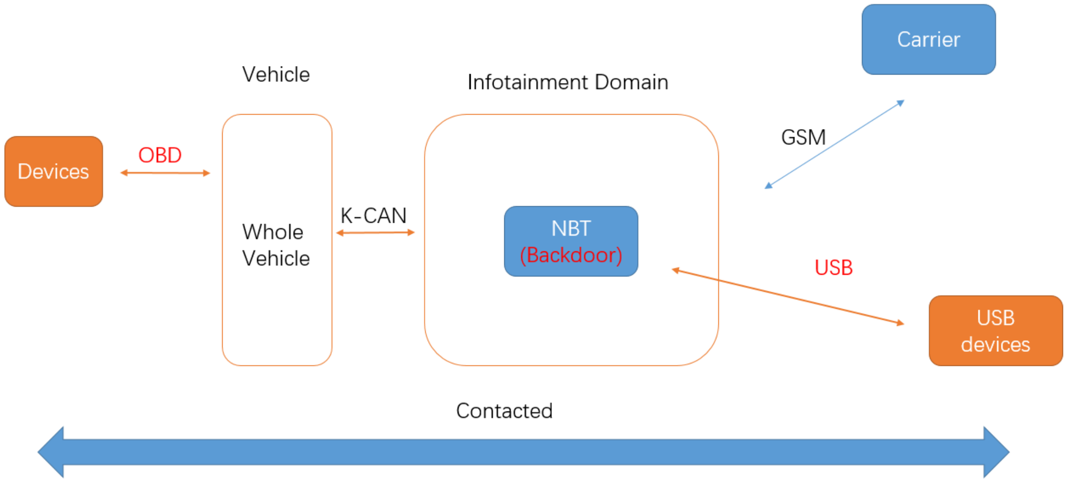


Figure: Attack Chain Based on USB and OBD-II Interfaces

4.2 Contactless Attack

The contactless attack is based on the wireless interfaces of the vehicle. And in such kinds of attack chains, attackers may impact the vehicle remotely. In this part, the attack chains via Bluetooth and Cellular network will be illustrated.

4.2.1 Bluetooth Channel

Bluetooth is a typical short-range communication protocol of NBT in the vehicle. With the vulnerabilities in Bluetooth Stack mentioned earlier, an attacker could affect the availability of the head unit without authentication when the Bluetooth is the pairing mode. However, such an attack could happen only when the attackers are very close to the vehicle and make the NBT work abnormally.

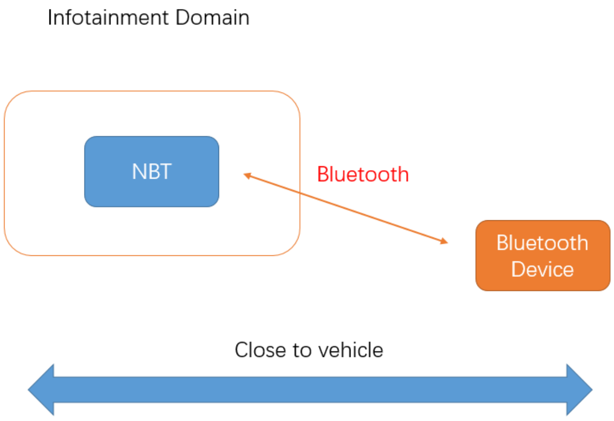


Figure: Attack Chain Based on Bluetooth Channel

4.2.2 Cellular Network

DISCLAIMER: WE CONDUCTED THE RESEARCH AS ETHICAL HACKING IN A CONTROLLED ENVIRONMENT (DO NOT TRY THIS IN THE REAL WORLD MOBILE NETWORK).

If the TCB has fallen into a rouge base station, attackers can extend the attack distance to a wide-range distance with the help of some amplifier devices. Technically speaking it's possible to launch the attack from hundreds of meters even when the car is in the driving mode. Using MITM attack between TSP and the vehicle, attackers could remotely exploit the vulnerabilities existed in both NBT and TCB, leading to backdoors being planted in the NBT and TCB. Typically, a malicious backdoor can inject controlled diagnosis messages to the CAN buses in the vehicle.

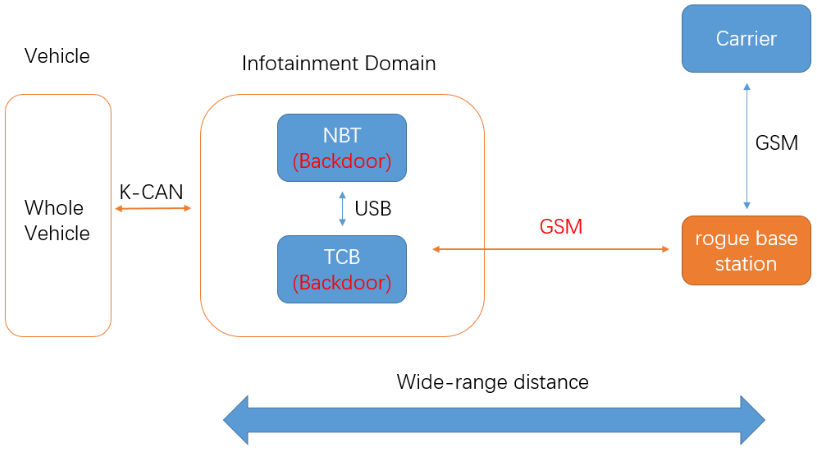


Figure: Remote Attack Chain Based on Cellular Network

5. Vulnerable BMW Models

In our research, the vulnerabilities we found mainly exist in the Head Unit, Telematics Control Unit (TCB), and Central Gateway Module. Based on our research experiments, we can confirm that the vulnerabilities existed in Head Unit would affect several BMW models, including BMW i Series, BMW X Series, BMW 3 Series, BMW 5 Series, BMW 7 Series. And the vulnerabilities existed in Telematics Control Unit (TCB) would affect the BMW models which equipped with this module produced from year 2012.

Table below lists the vulnerable BMW models we've tested during our research and each with its firmware versions of the specific components.

Model	Manufacture Date	Central Gateway	Head Unit	Telematics Control Unit
BMW i3 94(+REX)	2017.02.15	BDC (I01)	HU_NBT (MN-003.013.001 TN-003.013.001)	TCB NAD (003.017.020 APPL [Oct 7 2015 11:54:15])
BMW X1 sDrive 18Li	2016.07.27	BDC (F49)	HU_ENTRYNAV (MV-130.006.007 TV-130.006.007)	TCB NAD (003.017.020 APPL [Oct 7 2015 11:54:15])
BMW 525Li	2016.04.27	FEM (F18)	HU_NBT (MN-003.003.001 TN-003.003.001)	TCB NAD (003.015.022 APPL [Mar 5 2015 13:53:26])
BMW 730Li	2012-10-08	ZGW (F02)	HU_NBT (MN-001.020.022 TN-001.020.022)	TCB NAD (001.014.022 APPL [Mar 8 2012 17:10:58])

Table: Vulnerable BMW Models in Our Test

As different BMW car models may be equipped with different components, and even the same component may have different firmware versions during the product lifecycle. So that from our side the scope of the vulnerable car models is hard to be precisely confirmed. Theoretically, BMW models which are equipped with these vulnerable components can be compromised from our perspective.

BMW confirmed, that the found vulnerabilities are present in the Head Unit and T-Box components mentioned above. Interested customers can check whether their individual car could be affected and whether a SW update is available by following BMW official customer notification.

6. Disclosure Process

The research to BMW cars is an ethical hacking research project. Keen Lab follows the "Responsible Disclosure" practice, which is a well-recognized practice by global manufactures in software and internet industries, to work with BMW on fixing the vulnerabilities and attack chains listed in this report.

Below is the detailed disclosure timeline:

January 2017: Keen Lab kicked off the BMW security research project internally.

February 2018: Keen Lab proved all the vulnerability findings and attack chains in an experimental environment.

February 25th, 2018: Keen Lab reported all the research findings to BMW.

March 9th, 2018: BMW fully confirmed all the vulnerabilities reported by Keen Lab.

March 22nd, 2018: BMW provided the planned technical mitigation measures for the vulnerabilities reported by Keen Lab.

April 5th, 2018: CVE numbers related to the vulnerabilities have been reserved. (CVE-2018-9322, CVE-2018-9320, CVE-2018-9312, CVE-2018-9313, CVE-2018-9314, CVE-2018-9311, CVE-2018-9318)

May 22nd, 2018: This summary report is released to public.

Year 2019: Keen Lab will release the full technical paper.

BMW informed Keen Security Lab that, for all the attacks via cellular networks BMW has started implementing measures in March 2018. These measures are in rollout since mid of April 2018 and are distributed via configuration updates remotely to the affected vehicles. Additional security enhancements are developed by BMW in form of optional SW updates. These will be available through the BMW dealer network.

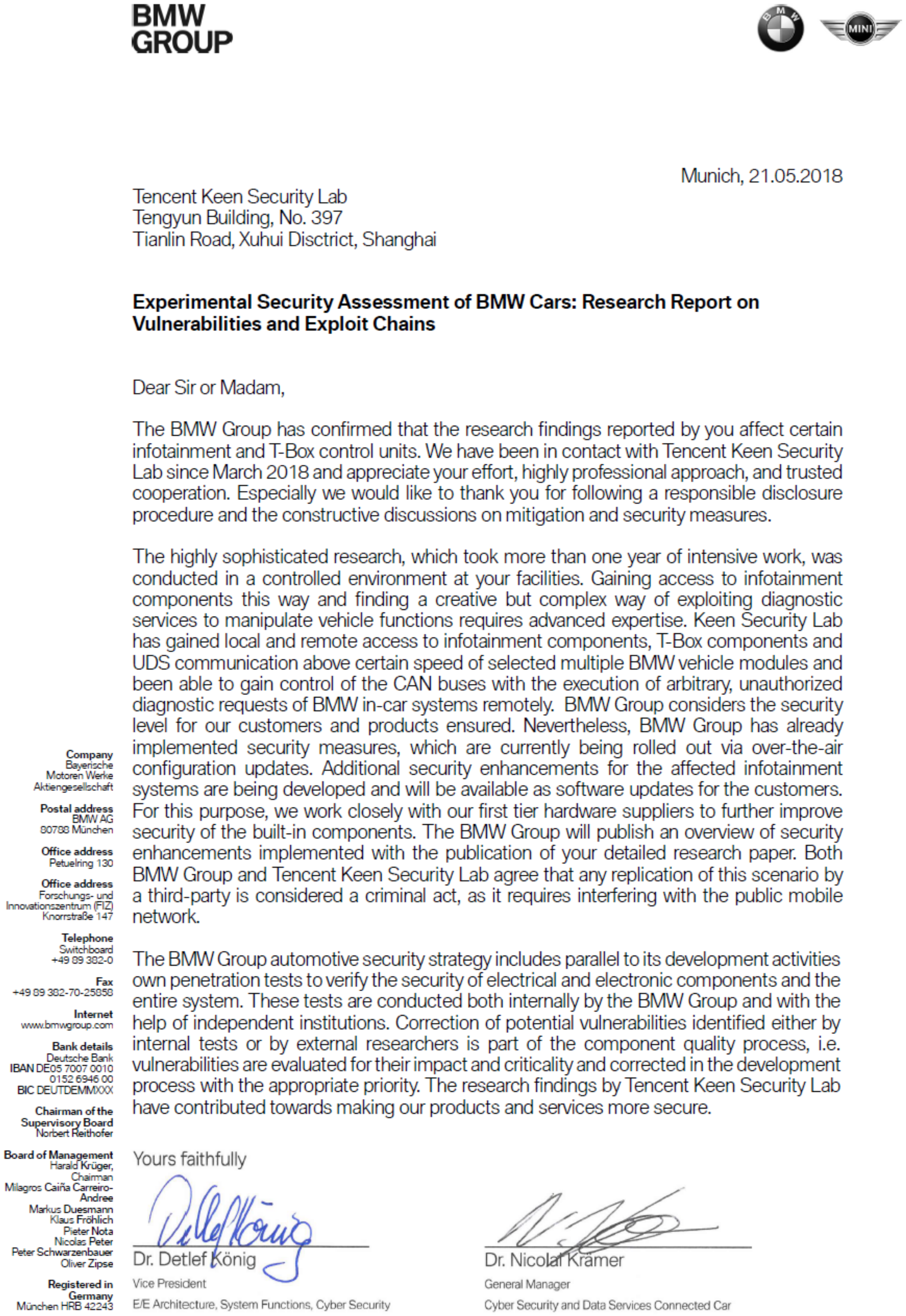
7. Conclusion

In this report, we revealed all the vulnerabilities we found in the Head Unit, Telematics Control Unit and Central Gateway Module. The vulnerabilities can be exploited by an attacker via the vehicle's external-facing I/O interfaces, including USB, OBD-II, and Cellular network. In particular, with the Telematics Control Unit being compromised without any physical access, the attacker can remotely trigger or control vehicular functions over a wide-range distance by sending malicious CAN messages to the BMW vehicle's internal CAN bus, whenever the car is in parking or driving mode.

In conclusion, our research findings have proved that it is feasible to gain local and remote access to infotainment, T-Box components and UDS communication above certain speed of selected BMW vehicle modules and been able to gain control of the CAN buses with the execution of arbitrary, unauthorized diagnostic requests of BMW in-car systems remotely.

Again, this report summaries our research findings in a pure technical viewpoint but without technical details, which may confuse the readers, especially some sophisticated security researchers. Because BMW Group and Keen Lab both agreed that it's not a proper time to publish a full disclosure of our findings or expose the impact to BMW car owners in this report. However, we do have a plan to publish a research paper with full technical details in early 2019.

Letter from BMW Group



BMW GROUP

Munich, 21.05.2018

Tencent Keen Security Lab
Tengyun Building, No. 397
Tianlin Road, Xuhui Disctrict, Shanghai

Experimental Security Assessment of BMW Cars: Research Report on Vulnerabilities and Exploit Chains

Dear Sir or Madam,

The BMW Group has confirmed that the research findings reported by you affect certain infotainment and T-Box control units. We have been in contact with Tencent Keen Security Lab since March 2018 and appreciate your effort, highly professional approach, and trusted cooperation. Especially we would like to thank you for following a responsible disclosure procedure and the constructive discussions on mitigation and security measures.

The highly sophisticated research, which took more than one year of intensive work, was conducted in a controlled environment at your facilities. Gaining access to infotainment components this way and finding a creative but complex way of exploiting diagnostic services to manipulate vehicle functions requires advanced expertise. Keen Security Lab has gained local and remote access to infotainment components, T-Box components and UDS communication above certain speed of selected multiple BMW vehicle modules and been able to gain control of the CAN buses with the execution of arbitrary, unauthorized diagnostic requests of BMW in-car systems remotely. BMW Group considers the security level for our customers and products ensured. Nevertheless, BMW Group has already implemented security measures, which are currently being rolled out via over-the-air configuration updates. Additional security enhancements for the affected infotainment systems are being developed and will be available as software updates for the customers. For this purpose, we work closely with our first tier hardware suppliers to further improve security of the built-in components. The BMW Group will publish an overview of security enhancements implemented with the publication of your detailed research paper. Both BMW Group and Tencent Keen Security Lab agree that any replication of this scenario by a third-party is considered a criminal act, as it requires interfering with the public mobile network.

The BMW Group automotive security strategy includes parallel to its development activities own penetration tests to verify the security of electrical and electronic components and the entire system. These tests are conducted both internally by the BMW Group and with the help of independent institutions. Correction of potential vulnerabilities identified either by internal tests or by external researchers is part of the component quality process, i.e. vulnerabilities are evaluated for their impact and criticality and corrected in the development process with the appropriate priority. The research findings by Tencent Keen Security Lab have contributed towards making our products and services more secure.

Yours faithfully

Dr. Detlef König
Vice President
E/E Architecture, System Functions, Cyber Security

Dr. Nicolai Krämer
General Manager
Cyber Security and Data Services Connected Car

Company
Bayerische
Motoren Werke
Aktiengesellschaft

Postal address
BMW AG
80788 München

Office address
Petuelring 130

Office address
Forschungs- und
Innovationszentrum (FIZ)
Knorrstraße 147

Telephone
Switchboard
+49 89 382-0

Fax
+49 89 382-70-25858

Internet
www.bmwgroup.com

Bank details
Deutsche Bank
IBAN DE05 7007 0010
0152 6946 00
BIC DEUTDEMMXXX

Chairman of the Supervisory Board
Norbert Reithofer

Board of Management
Harald Krüger,
Chairman
Milagros Caiña Carreiro-Andree
Markus Duesmann
Klaus Fröhlich
Pieter Nota
Nicolas Peter
Peter Schwarzenbauer
Oliver Zipse

Registered in Germany
München HRB 42243

About Tencent Keen Security Lab

Figure: Participants of BMW Project at Keen Lab

Tencent Keen Security Lab[1] (in abbreviation "Keen Lab") is a professional security research team, focusing on information security research of both attack and protection techniques, under Tencent Company. In the past years, Keen Lab built security research partnership with global manufactures in software, hardware and internet industries such as Microsoft, Apple, Google, Qualcomm, Samsung, etc.., and achieved a lot of worldwide leading security research results.

Since Year 2015, Keen Lab started research projects in IoT[2] and Connected Vehicle categories and building partnership with manufacturers in IoT and car industries. In the Year 2016 and 2017, Keen Lab published the well-known research globally on "Tesla Model S and Model X Remote Hacking" with leveraging "Responsible Disclosure" practice to report the vulnerabilities and attack chains to Tesla[3,4].

[1] https://keenlab.tencent.com/
[2] https://keenlab.tencent.com/zh/2017/04/01/remote-attack-on-mi-ninebot/
[3] https://keenlab.tencent.com/en/2016/09/19/Keen-Security-Lab-of-Tencent-Car-Hacking-Research-Remote-Attack-to-Tesla-Cars/
[4] https://keenlab.tencent.com/en/2017/07/27/New-Car-Hacking-Research-2017-Remote-Attack-Tesla-Motors-Again/