Driving down the rabbit hole

Jesse Michael
Mickey Shkatov
Oleksandr Bazhaniuk
Agenda

- Who are we
- Background
- Picking our battles
- The web vuln
- Intermission
- Telematics
  - What is it
  - Local vulnerabilities discovered
  - Writing a blind exploit
  - Remote vulnerability
- Conclusion
- Public statements
- Questions
After we were done with our previous hackary, we wanted to try something new.

We want to deepen our knowledge and experience with automotive security.

Actual car hacking experience is at 0%.
• Autonomous vehicles
  ○ Tesla Autopilot
  ○ Comma.io
  ○ Google self driving car
  ○ UBER
• Connected cars
  ○ Autonomous
  ○ V2X
  ○ V2V
• Drive by wire systems, how does it work?
Background

If I have seen further than others, it is by standing upon the shoulders of giants.

- Charlie Miller and Chris Valasek
- Troy Hunt and Scott Helme - Nissan web API hack
- Kevin Mahaffey and Marc Rogers (Tesla hack 2015)
- Keen Labs Tesla hack
- And more...
Background

- Budget?

- Where do we start?
- We already pwned an after market IVI, what is next?
  *IVI = In-Vehicle Infotainment System

- Ok, Lets go the wrecking yard and look around
Funny story about the wrecking yard.
  ○ Looking for a late model OEM IVI.
  ○ “What do you have?”
    ■ An F150 that got into a brawl with a wall and lost
    ■ and more squashed cars
● A junk yard != wrecking yard.
Nice car!
Can you spot what caused it to be “Totaled”?
GIMME THAT DASHBOARD!

BACKGROUND

- GIMME THAT DASHBOARD!
Background

- 1 week later
- carception
Background

- A trip to Lowe’s and a few hours later
Once it is fully assembled it kinda works
A “few” errors appear on the instrument panels.
We need to get this thing on the table somewhat functional
NissanConnect℠ EV
NissanConnect℠ EV (formerly known as CARWINGS®) is designed to help you manage your Nissan LEAF® and control a host of convenient features. The best part: you don’t have to be in or even near your car to do it. It all works through your smartphone or computer. [*]

NissanConnect EV is complimentary for three years. You just need to download the companion app to run all the features listed below.

WITH THE NISSANCONNECT℠ EV APP, YOU CAN:

- Find a nearby charging station
- Check on the state of your battery charge
- Remotely start a charging session
- Get notified when your battery is fully charged
- See your estimated driving range
- Heat up or cool down your LEAF® to the comfortable temperature it was when you left it
- Set a reminder to plug in your car

Source: https://www.nissanusa.com/connect/features-app/system-requirements/nissan-connect-ev
Next step, switch owners in the backend
Go ask nicely for the title from wrecking yard, ahh….. No.
Wrecking guy reaction:

- junk title can’t be moved.
- Bill of sale, wrecking yard receipt?
  - ask nissan nicely and you shall receive
Picking our battles
Picking our battles

- We already pwned one in the past, seems like the best place to start.
- Looking at the IVI attack surface:
Picking our battles

- The IVI is running Windows Automotive 7, no source, requires license.
- That's too hard!, we want to hack this but...
- Maybe there is something simpler to hack in our sights, let's keep looking.
Picking our battles

- Getting any kind of info from the IVI
Picking our battles

- Getting any kind of info from the IVI

- Navigation system debug data
- contacts
- way points
- SRAM dump
Picking our battles

- Getting any kind of info from the IVI
  - Navigation system debug data
  - contacts
  - way points
  - SRAM dump
  - Flash dumps
THE WEB VUIN
After running strings on the debug files we discovered this url:

“http://biz.nissan-gev.com/WARCondelivbas/it-m_gw10/”

- Let’s do a WHOIS
- no one owns it, let’s buy it for the lulz!
- setting up an EC2 instance and running a generic honey pot
- Let's see who comes knocking
- The Web vulnerability
- First knock comes from Japan

<table>
<thead>
<tr>
<th>IP Address</th>
<th>150.63.64.10</th>
</tr>
</thead>
<tbody>
<tr>
<td>IP Location</td>
<td>Japan</td>
</tr>
<tr>
<td>IP Reverse DNS (Host)</td>
<td>150.63.64.10</td>
</tr>
<tr>
<td>IP Owner</td>
<td>Nissan Motor Co. Ltd</td>
</tr>
<tr>
<td>Owner IP Range</td>
<td>150.63.0.0 - 150.63.255.255 (92,636 IPs)</td>
</tr>
<tr>
<td>Owner Country</td>
<td>Japan</td>
</tr>
<tr>
<td>Owner Website</td>
<td><a href="http://www.odm.ne.jp">www.odm.ne.jp</a></td>
</tr>
<tr>
<td>Owner CIDR</td>
<td>150.63.0.0/16</td>
</tr>
<tr>
<td>Whois Record Created</td>
<td>17 Jun 1991</td>
</tr>
<tr>
<td>Whois Record Updated</td>
<td>19 Nov 2013</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Web Browser/s on this IP:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Firefox 11</td>
</tr>
<tr>
<td>Chrome 27</td>
</tr>
<tr>
<td>Firefox 15</td>
</tr>
<tr>
<td>Google Chrome 31</td>
</tr>
<tr>
<td>Firefox 27</td>
</tr>
<tr>
<td>Google Chrome 33</td>
</tr>
<tr>
<td>Firefox 32</td>
</tr>
<tr>
<td>Google Chrome 37</td>
</tr>
<tr>
<td>Google Chrome 25</td>
</tr>
<tr>
<td>...</td>
</tr>
<tr>
<td>[see all]</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>OS on this IP:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows 7 x64 Edition</td>
</tr>
<tr>
<td>Windows 8 x64 Edition</td>
</tr>
<tr>
<td>Windows XP</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Browser Agent/s on this IP:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; .NET CLR 1.1.4322; .NET CLR 2.0.50727)</td>
</tr>
<tr>
<td>Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; .NET CLR 1.1.4322; MS-RTC LM 8; .NET</td>
</tr>
</tbody>
</table>
The Web Vulnerability

- First knock comes from Japan
The web yin

- The Web vulnerability
- First knock comes from Japan
- But then we start getting more knocks on the door and these are not your usual automated tools.

POST /WARCondelivbas/it-m_gw10/ HTTP/1.1
Host: biz.nissan-gev.com
Connection: Keep-Alive
User-Agent: NISSAN CARWINGS
Content-Type: application/x-carwings-nz
Content-Length: 614

^@^@^Cä^@^@^BZx<9ØK<81>:6%?Ćsdd^TQ<82><j4A_<87>æL0^Rí^Pi(<81
häsdxςΩβεÇtHûld^H rzwggg×l<92>^\$É»
^?)cÜlx^X<83>Éæó | ¶<9c><Ö:<81>óïïíÝ/ÜOőâ<9c>¼?=<87><23>óÔ:Ô<98><82>ÂA<89>4eS³)
ëÝïÖQN<8349>ôÂTB7F^VÔ4ôuiď´Tv<8b>^P÷<9a><9a>M2<87>ê<WfM<8c>ê^UW^U ßĐ|ÄK]Pi-%UYG^?
Æ²4:gl<89>Rj,ÍOò%cx¶LÖ<93><80>°X.ëÜét^G<8f>B÷ng,µßZ^N^±xcfAW
«˝Qê²δ|A<80>ôdé^B^GjţN^V<94>±E$PDûíEpşÛq@^?<81>^Bl_jâÚjiÖ<96>^H<86><90>ÇA^xNWPç<9b>
<96>^Rë`^B«¶
The web yuIn

- The Web vulnerability
  - First knock comes from Japan
- but then we start getting more knocks on the door and these
  are not your usual automated tools.
The Web YUIN

- The Web vulnerability
  - First knock comes from Japan
- but then we start getting more knocks on the door and these are not your usual automated tools.
The Web vulnerability

- First knock comes from Japan
- But then we start getting more knocks on the door and these are not your usual automated tools.
- We got cars connecting to our server?!!
The Web Yuin

- The Web vulnerability
  - First knock comes from Japan
  - But then we start getting more knocks on the door and these are not your usual automated tools.

```xml
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<carwings version="2.2">
  <aut_inf navi_id="1054********" tel="err" dcm_id="2012********"
dcm_tel="380***********" sim_id="89380***************" vin="1N4A************" user_id="********" password="********"></aut_inf>
  <bs_inf><sftwr_ver navi="041-102-10111000000003010100" map="006" dcm="3NF0000642"></sftwr_ver>
  <vcl spd="0" drc="138.5" sts="stop" rss="5" crr="life:)">
    <crd datum="wgs84" lat="40,00,**.**" lon="-75,01,**.**"></crd></vcl>
  <navi_set t_zone="-8.00" lang="use" dst_d="km" tmp_d="C" e_mlg_d="km/kwh" spd_d="km/h"></navi_set></bs_inf>
  <srv_inf><app name="AP"><send_data id_type="file" id="APUP001.001"></send_data></app></srv_inf>
</carwings>
```
The web Yuin

The cars sent us plenty of data, including location, let's look at one of them a bit closer.
The web yuin

The cars sent us plenty of data, including location, let's look at one of them a bit closer.
The cars sent us plenty of data, including location, let's look at one of them a bit closer.
Who owns this car? we have the VIN, let's google...

Sample shipment record for Ярославович П - Т Ет Від 19.02.2011Р. Коломийським Рв Умвс

<table>
<thead>
<tr>
<th>BILL OF LADING</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Recipient</strong></td>
</tr>
<tr>
<td>Ярославович П - Т Ет Від 19.02.2011Р. Коломийським Рв Умвс</td>
</tr>
<tr>
<td>Печеніжин, Вул. Прикарпатська</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Cargo Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>1. МОТОРНИЙ ТРАНСПОРТНИЙ ЗАСІБ ДЛЯ ПЕРЕВЕЗЕННЯ ПАСАЖИРІВ ПО ДОРОГАХ ЗАГАЛЬНОГО КОРИСТУВАННЯ: ЛЕГКІВ ОБОВ'ЯЗ Комуністичний МОДЕЛІ LEAF, НОМЕР КУЗОВА ТИП ДВИГУНА Е</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>HS CODE</th>
<th>ARRIVAL DATE</th>
<th>WEIGHT</th>
<th>PRICE</th>
<th>DECLARATION NO.</th>
<th>CURRENCY RATIO</th>
<th>CURRENCY NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2016-03-25</td>
<td>1493.00</td>
<td>26777.20</td>
<td></td>
<td>26.25</td>
<td>840</td>
</tr>
</tbody>
</table>

View more
The web yuin
Who owns this car? we have the VIN, lets google...

Sample shipment record for Futter 1234567 Y. P - T Et Vid 19.02.2011R. Kolomiisky District Police

Y. P - T Et Vid 19.02.2011R. District Police Kolomiisky imports from Not Available

<table>
<thead>
<tr>
<th>BILL OF LADING</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Recipient</strong></td>
</tr>
<tr>
<td>Y. P - T ET VID 19.02.2011R. KOLOMIYSKY DISTRICT POLICE</td>
</tr>
</tbody>
</table>

Cargo Description

VEHICLE 1 MOTORYY FOR PASSENGER TRANSPORT ON PUBLIC ROADS. PASSENGER CARS MODEL NISSAN LEAF, BODY ISSUE , TYPE MOTOR ED "

View more
Why is this happening?

- Owner replacing the SIM card in their car.
- The Jasper network.
INTERMISSION
Continental made Telematics Control Unit (TCU)
Used as the conduit for the car to connect to the backend.
Older model, buy it on eBay for cheap.
Telematics

- Uses a cellular 2G modem
● Uses a cellular 2G modem
● Yes
● Uses a cellular 2G modem
● Yes
● 2G
Telematics

Connected to the rest of the car like this
Telematics

Gathering Intel from the board
- Exploring the TCU - TOP
Telematics

Gathering Intel from the board
- Exploring the TCU - Bottom
Exploring the TCU

- Small connector (USB)
- Freescale chip CAN
- UART CAN
- Big connector
- 2G Cellular complex USB
- ANT
Exploring the TCU

- Small connector (USB)
- Big connector
- ANT
- UART
- CAN
- Freescale chip
- 2G Cellular complex
- USB
Gathering Intel from the board

- Freescale chip debug header, lets get firmware
Gathering Intel from the board

- Its USB right? lets mitm it! CAR<-usb->LAPTOP<-usb->TCU
Gathering Intel from the board

- Its USB right? Let's MITM it! CAR<-usb->LAPTOP<-usb->TCU
- This looks familiar...

```
2.C.o.m.n.e.o.n... 2.C.D.C.
```

Delta time: 0:15:13.6262.566 Transferred length: 235 bytes (-0.015)
Gathering Intel from the board

- Its USB right? let's MITM it! CAR<-usb->LAPTOP<-usb->TCU
- This looks familiar...
Gathering Intel from the board

- Its USB right? lets mitm it!
- This looks familiar...
- Oh, look at that!
- I know this chip! Do you?
Telematics

- Here are a few hints
Infineon PMB 8876
Functional diagram
Gathering Intel from the board

- It’s a USB system. We know this...
- Let’s connect to it and explore

```
root@atr-1t01:~/leaf# ./leaf.py
AT
OK

AT+CGMI
+CGMI: Continental Automotive Systems
OK

AT+CGMM
+CGMM: "GSM900","GSM1800","GSM1900","GSM850","MODEL=SGOLD2 NAD"
OK

AT+CGMR
+CGMR: "06.42R_51R_V26"
OK

AT+CIMI
310650701614947
OK
```
Telematics vulnerabilities

- Ok, now that we have gathered our senses together, let's check for known vulnerabilities...

```
AT+CIMI
310650701614947
OK

AT+XLOG
+XGENDATA: "cas2_21.41.23:NOVANTO_NAD_51R    dows_NT"
OK

AT+XAPP="AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"
```

Traceback (most recent call last):
 File "./leaf.py", line 32, in <module>
   dev.write(2, "\%s\r" % command)
 File "/usr/lib/python2.7/dist-packages/usb/core.py", line 948, in write
   self.__get_timeout(timeout)
 File "/usr/lib/python2.7/dist-packages/usb/backend/libusb1.py", line 824, in bulk_write
   timeout)
 File "/usr/lib/python2.7/dist-packages/usb/backend/libusb1.py", line 920, in __write
   _check(retval)
 File "/usr/lib/python2.7/dist-packages/usb/backend/libusb1.py", line 595, in _check
 raise USBError(_strerror(ret), ret, _libusb_errno[ret])
usb.core.USBError: [Errno 5] Input/Output Error
root@atr-lt01:~/leaf#
Telematics vulnerabilities

- Ok, now that we have gathered our senses together, let's check for known vulnerabilities...
Telematics vulnerabilities

- Ok, now that we have gathered our senses together, let's check for known vulnerabilities...

```
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

+XLOG: Exception Number: 2
Trap Class: 0xBBBB (HW PREFETCH ABORT TRAP)
System Stack:
- 0x41414141
- 0x41414141
- 0x41414141
- 0x41414141
- 0x00414141
- 0x00000001
- 0xA024111C
- 0x00000001
- 0xB013FDA8
- 0xB026FF14
- 0xB00325FC
- 0x00000001
- 0x00000001
- 0xA013216D
- 0x00000001
- 0xB00546EC
- 0xB026FF14
- 0xA01081AD
- 0xB026F164
Telematics vulnerabilities

- Ok, now that we have gathered our senses together, let's check for known vulnerabilities...
Telematics vulnerabilities

- Ok, now that we have gathered our senses together, let's check for known vulnerabilities...
Telematics vulnerabilities

- Ok, now that we have gathered our senses together, let's check for known vulnerabilities.
- confirmed local vector
  - AT+STKPROF
  - AT+XAPP
  - AT+XLOG
  - AT+FNS
Telematics vulnerabilities

- After confirming the local vulns, let’s check for remote ones…
- oh wait!
- Thanks to the amazing Dr. Ralf-Philipp Weinmann we know this baseband FW is vulnerable to an Over-The-Air TMSI buffer overflow.
Telematics vulnerabilities

- Confirming the TMSI vulnerability
  - The good book has PoC code in it, yay!
  - OpenBTS has moved on from testcall functionality ("security" reasons)
  - this will take a while, better get a faraday cage
Telematics vulnerabilities

- Confirming the TMSI vulnerability
Telematics vulnerabilities

- Confirming the TMSI vulnerability
- After many many days of attempts and trying to get OpenBTS to work, Jesse confirms remote buffer overflow!
  - Thank you Jared Boone!

![Image of a computer screen showing register values with hexadecimal numbers, including dates: 01.01.2004, time: 00:24, and various register values like r0, r1, r2, r3, r4, r5, etc., with values such as 0x00000000, 0xB00B0B9D, 0xFFFF231C, etc.]

![Image of a person standing outdoors with one arm raised, smiling.]
Telematics vulnerabilities

- Exploiting
  - We don’t have a copy of the firmware, how do we fix this?
  - Getting the firmware out of the device requires semi-blind exploitation
  - It’s not quite that bad, we have some basic exception logging that includes:
    - Register state at time of crash
    - 178 dwords of stack values upwards from SP at time of crash
  - We can work with that
Telematics vulnerabilities

- Exploiting
  - No DEP
  - No ASLR

A high five doesn’t even cut it.

HIGH SIX!
Telematics vulnerabilities

- Exploiting
  - We’ll just use AT command buffer overflow to inject payload to:
    - Write tag to signify start of data block
    - Copy 512 bytes from arbitrary location into stack frame
    - Write tag to signify completed copy of data block
    - Jump to hardcoded invalid location to force a crash at specific location
    - Wait for device to reboot
    - Read exception log using AT+XLOG and extract data from between tags in stack dump
    - … and then do it again 13 thousand times …
Telematics vulnerabilities

- Once firmware is accessible we can work on reversing and jumping from the baseband to the CAN bus
Conclusion
Thank you!

Questions?