CAR HACKING ON SIMULATION
BUT WHAT IS CAR HACKING?
A computer hacker remotely unlocks your car, disables the alarm system and steering lock, starts your car, and drives it away.

SCENARIO 1
You are driving along the highway on a fine summer’s day when your windshield fluid sprays onto your windshield, your radio immediately ramps up to full volume, your steering wheel develops a mind of its own, and your car then grinds to a halt in the fast lane.

SCENARIO 2
JEEP CHEROKEE HACK IN 2015 BY CHARLIE MILLER AND CHRIS VALASEK
ABOUT ME

Rohan Aggarwal @nahoragg
Ex Offensive Security Analyst
Independent Security Researcher
Bug Bounty Hunter - Yahoo, Twitter, Goldman Sachs etc.
Security Consultant
AGENDA

- Get started in car hacking - Fast, cheap & easily.
- Demonstrate how vulnerable CANBUS is.
- Encourage more research in this field.
- Encourage automakers to produce more secured system.
| 01 | ATTACK SURFACE |
| 02 | CAR NETWORK BASICS |
| 03 | CANBUS HACKING |
| 04 | DEMONSTRATION |
ATTACK SURFACE

Which one will you choose?
BIRD EYE VIEW OF DIFFERENT ATTACK SURFACE
02
CAR NETWORK BASICS
Let’s start with the basics
“Cars are basically laptop on wheels.”

— ELON MUSK
“Connected Cars are basically laptop smartphone on wheels.”

— ELON MUSK
ANONYMOUS
ECU = Engine Electronic Control Unit
CANBUS - The Network
CANBUS - The Network

 Released in 1986
CANBUS - The Network

>Released in 1986
CANBUS - The Network

Released in 1986
>Broadcasting communication - UDP
CANBUS - The Network

Released in 1986
Broadcasting communication - UDP
>Cheap, lightweight and robust
CANBUS - The Network

Released in 1986
Broadcasting communication - UDP
Cheap, lightweight and robust
> Constant flow of packets
**Arbitration ID (Green)** - Identification & Prioritization | 11 or 29 bit

**CAN Data (Red)** - Tells what function to perform | 8 byte
Lowest ID = Highest Priority

0 = Dominant Bit
1 = Recessive Bit

**Airbag** - Very High Priority - lower arbitration ID

**Door Lock** - Low Priority - high arbitration ID
Lowest ID = Highest Priority

Node 3 will have highest priority.
CANBUS - The Network

- Released in 1986
- Broadcasting communication - UDP
- Cheap, lightweight and robust
- Constant flow of packets
- New Component easily added
CANBUS - The Network Culprit
CANBUS - The Network Culprit

>30 years old, unchanged & still used
CANBUS - The Network Culprit

30 years old, unchanged & still used
> No Segmentation and Boundary Defense
CANBUS - The Network Culprit

30 years old, unchanged & still used
No Segmentation and Boundary Defense
>Lack of Device Authentication
CANBUS - The Network Culprit

30 years old, unchanged & still used
No Segmentation and Boundary Defense
Lack of Device Authentication
>Unencrypted Traffic
SO FAR
SO FAR
SO FAR
SO FAR
OBD-II PORT
OBD-II PORT

- Designed for diagnostic purpose
- Mandatory for new cars after 1996
- Direct interface to a vehicle’s CAN bus
- Usually found underneath the dashboard
OBD-II PINOUT

- Chassis/Signal Ground
- SAE J1850 BUS+
- ISO15765-4 (CAN Bus High)
- ISO9141 (K-Line)
- 12 (Always ON)
- SAE J1850 BUS-
- ISO9141 (L-Line)
- ISO15765-4 (CAN Bus Low)
UNAUTHORIZED CAN ACCESS

http://signalochka.ru/ ©

http://elauto-spb.ru/ ©
03 CANBUS HACKING
How do they do it?
GENERAL METHODOLOGY

- Get access to the CANBUS, somehow
- Sniff the packets
- Reverse Engineer them
- Identify the arbitration ID for an event
- Replay Them
SETTING UP ENVIRONMENT

- Update: `sudo apt-get update`
- Install LibSDL: `sudo apt-get install libsdl2-dev libsdl2-image-dev`
- Install CAN-utils: `sudo apt-get install can-utils`
- Download ICSim: `git clone https://github.com/zombieCraig/ICSim.git`
- Prepare Virtual CAN port: `sh /ICSim/setup_vcan.sh`
 nahor@nahor-VirtualBox:~/ICSim$ ./setup_vcan.sh
[sudo] password for nahor:
nahor@nahor-VirtualBox:~/ICSim$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qLEN 1000
   link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
   inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qLEN 1000
   link/ether 08:00:27:49:6c:a4 brd ff:ff:ff:ff:ff:ff
   inet 10.0.2.15/24 scope global dynamic noprefixroute enp0s3
       valid_lft 81602sec preferred_lft 81602sec
   inet6 fe80::b16d:cd8d:287a:6a9f/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
4: vcan0: <NOARP,UP,LOWER_UP> mtu 72 qdisc noqueue state UNKNOWN group default qLEN 1000
   link/can

 nahor@nahor-VirtualBox:~/ICSim$
RUNNING THE ICSIM SOFTWARE

- /ICSim/icsim vcan0
- /ICSim/controls vcan0
SNIFFING THE CAN PACKETS

- Can-utils
- Wireshark
SNiffing the Can Packets

- candump vcan0
- cansniffer -c vcan0 >
SNIFFING THE CAN PACKETS

- candump vcan0 >
- cansniffer -c vcan0
Delete half of the recording

Play first half

Did the doors unlock? NO

Play second half

Did the doors unlock?

NO

Not in recording

YES

Are you down to one packet?

NO

SUCCESS

YES

Did the doors unlock?
DEMONSTRATION
SUPER EASY ATTACK - DOS

```c
while(1) {
    send_message_with_id(0);
}
```
HACKING THROUGH ERROR MESSAGES

- **ACTIVE**
  - able to send high-priority errors
  - err > 128

- **PASSIVE**
  - could only send low-priority errors
  - err > 256

- **OFF**
CANBUS HACKING SHORTFALL

- Difficult to attack canbus physically through OBD-II.
- Reverse Engineering Arbitration ID’s.
WAYS TO FIND ARBITRATION ID’s

- Reverse Engineer
- Fuzzing
- Open Database
- Buy Somehow
CONCLUSION

- Attack Surfaces
- ECU
- CANBUS
- OBD-II
- Attack Methodology
  - Sniffing
  - Reverse Engineering
  - Replaying
CONCLUSION

- DOS Attack
- Error Message Attack
- CANBUS Shortfall
- Ways To Find Arbitration ID
your data

your obscurity controls

hackers
THANKS!

Does anyone have any questions?

nahoragg@gmail.com
Twitter: @nahoragg