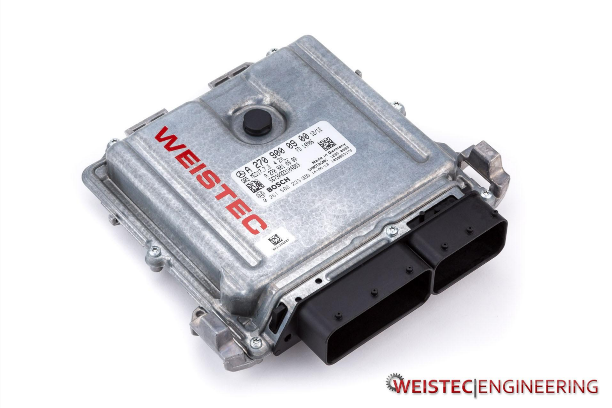
Automotive Attack Surfaces

UCSD and University of Washington

Current Automotive Environment

- Modern cars are run by tens of ECUs comprising millions of lines of code
- ECUs are well connected over internal buses (mainly CAN buses) to enable both critical safety and convenience features
  - Engine control, brakes, steering
  - Phone connectivity, bluetooth, cellular functionality
- Need connectivity for alerting the driver
  - Oil pressure warnings, engine efficiency information
  - Voice control of car functions, etc.



Previous Research

- Previous research has focused on vulnerabilities in the car’s internal network
    - What can be accomplished by sending packets on the car’s internal networks?
- By exploiting the connectivity and lax restrictions, an adversary can circumvent all control systems for complete access
- However, these approaches are restrictive

Previous Research (the problem)

- The threat model of previous research assumes access to the car's internal buses (unrealistic)
- An adversary with this access can carry out physical attacks for a lower cost than ECU exploits
- Instead of controlling the ECU for brakes, cut the brake lines
- Similarly: damage the steering column, remove the radiator, etc. (Really, any kind of physical tampering you can imagine)

Goals of this Paper

- This paper aims to evaluate the external attack surface of modern, mass-produced automotive systems
  - Also study how, where and why vulnerabilities arise
- Unlike previous work, it analyzes the remote attack surface provides a basis for the feasibility and practicality of attacks
  - First to study the external attack surface of modern cars

This Paper: Contributions

- **Threat model characterization**: external attack access vectors and delivery
- **Vulnerability analysis**: practical vulnerabilities for access vectors
- **Threat assessment**: utility of vulnerabilities to an attacker

Automotive Threat Model

- **Technical capabilities:** adversary's knowledge of its target and the ability to develop malicious inputs
  - Assumed that he/she has access to the automobile model being targeted (information)
  - Can not brute force cryptography or solve computationally hard problems

- **Operational capabilities**: adversaries ability to deliver malicious inputs
  - Three main categories:
    - Indirect physical access
    - Short-range wireless access
    - Long-range wireless access

Indirect Physical Access

- Access that comes from a physical connection without the presence the attacker
- Leave it to the user to create the connection
    - Compromise other devices OOB
- Cars provide several interfaces for connection to the internal network
    - For both convenience and safety features connected on CAN

Indirect Access: Disc, USB, iPod

- Malicious payload could be encoded on a CD
  - Exploit audio decoding and parsing software
- Many such systems are CAN bus connected
  - Even entertainment access can achieve complete compromise
- A phone that is compromised OOB can deliver malicious payload
  - Malicious trojan applications have been seen on the app store

- Compromised USBs provide direct physical connection to the car

Indirect Access: OBD-II Port

- Included in all modern cars
- Directly communicates on the CAN bus

- Mainly accessed by personnel during maintenance and ECU programming
    - Personnel use manufacturer scan tools like Toyota’s Diagnostic Tester
    - Similar tools provided by other car manufacturers too

- PassThru device allows clients to connect (TCP) to CAN bus wirelessly through API
    - Compromised PC could deliver payload to compromise PassThru

Short-Range Wireless Access

- Reasonable threat model where attacker is in the vicinity of the target vehicle
- Communicating remotely prevents detection and is more realistic
- Often less complex than getting indirect physical access
    - Requiring OOB compromising of devices like smartphones and diagnostic tools is cumbersome
- Access through wireless interfaces like: Bluetooth, RFID, WiFi

Short-Range Wireless Access

- **Bluetooth**: available in most cars with a range that can be extended beyond 10m
- **Keyless Entry**: RF communication that can control lights, locks and even ignition
- **WiFi**: car acts as a hotspot connection to the internet
- **Dedicated Short-Range Comm. (DSRC)**: cars can communicate with others nearby
- Vulnerabilities in the ECU for any of these allows an adversary to deliver the payload

Long-Range Wireless Access

- Access to a vehicle at great distances (on the scale of miles)
- Adversary with this access can compromise vehicles from anywhere
- **Broadcast Channels**: undirected channels that receivers tune into like GPS, digital radios
  - Part of media system that is connected to other key ECUs

Long-Range Wireless Access

- **Addressable Channels**: directed, unlike broadcast channels
- Often use cellular voice and data networks and can be accessed over arbitrary distances
- High bandwidth and two-way to meet consumer need for data exfiltration (Onstar, vehicle assistance, phone calls)

Vulnerability Analysis

Vulnerability Analysis: Setup

- Experiments conducted on a late-model, mass-produced sedan
  - Representative of the average consumer vehicle
- Around 30 ECUs for all critical and convenience functionality
- Equipped to expose interfaces like OBD-II, Bluetooth, GPS, telematics unit
  - Telematics provides voice/data through cellular networks and is connect to all CAN buses

Vulnerability Analysis

- First determined how to control important ECUs over CAN bus
- Reverse engineering firmware for each ECU
  - Created native debuggers for some components
  - UART interfaces also used
- Observed normal behavior to determine correct operation
- With I/O control, were able to rewrite ECU firmware, modify memory and can control the entire car by compromising only one ECU
  - Used mostly available debugging and diagnostic tools

Vulnerabilities: Indirect Physical Access

Media Player: audio disc player accepts formats like MP3 and WMA

1. *Undocumented* feature allows automatic reflashing of unit with properly formatted disc
   a. Cryptic message is the only way for user to prevent this

2. The file parsers make strong assumptions about the length of inputs
   a. Access only to BSS segment, not the stack
   b. Created debugger to find important pointers that give stack access
   c. Careful encoding prevents detection even when played on a PC

(Underscores need the for formal specification/verification of software)

Vulnerabilities: Indirect Physical Access

OBD-II: PassThru device allows WiFi access to the CAN bus

1. Communication between client apps and PassThru is unauthenticated
   a. Input validation bugs in API allows bourne shell access
   b. Telnet, ftp and nc already exist allowing trivial access and means of payload transmission

- Implanted malicious code in the PassThru device allows CAN access to <u>every</u> car that it plugs into
  - Created a worm that finds and infects other PassThru devices
  - Attack can be fully automated

Vulnerabilities: Short-Range Wireless

Bluetooth: connected to the telematics unit with custom implementation

1. Over 20 calls that strcpy onto the stack were identified and none of them were properly secured
   a. Buffer overflow allows arbitrary code execution
   b. No stack defenses
   c. Any paired bluetooth device can carry out attack

- Indirect wireless attack and direct wireless attack

Bluetooth - Indirect

- Hard for adversary to pair with target car
- Exploit smartphones that will connect via Bluetooth
- Applications and web sites are capable of installing and acting as Trojans that find telematics units

Bluetooth - Direct

MAC address and PIN number needed to pair with in-car Bluetooth

1. The MAC address is readily available as it’s broadcast every time a device attempts to find known bluetooth devices (sniff phone broadcast)
2. Car Bluetooth will respond to pairing requests without any user input
   a. PIN number can be brute forced
   b. However, takes about 10 hours to brute force
   c. Is this really a practical scenario?

Vulnerabilities - Long-Range Wireless

Telematics: Airbiquity aqLink software modem is used to communicate voice and data over cellular service in most North American cars

- Tone-based signaling used to switch between cellular and data
- “Stealth” mode hides any evidence of communication when call is a pure data call
  - Avenue for attacker to create connection to car telemetry without detection

Vulnerabilities: Telematics (connectivity)

- Reversed engineering aqLink protocol by observing audio signals during call
- Debug flags/methods creates ground truth binary log for packet identification
    - Debug tools/flags not removed in production
- Mismatch in assumptions in the “glue” connecting aqLink and command program allows for buffer overflow (packet size)
    - Protocol is low-level and circumvents higher-level authentication checks
    - However, this approach is not practical on its own because protocol ends call before entire payload is sent

Vulnerabilities: Telematics (authentication)

- A challenge response protocol is used to authenticate data calls to the car
- However, there are implementation errors

1. The random challenge is not really random
    a. The same constant seeds the RNG when the system is restarted
    b. Attacker can replay a sniffed response packet (very easy)
2. A bug allows invalid response to be accepted
    a. Approximately 1 out of 256 challenges accept
    b. Once accepted, payload can be transmitted without any indication to the driver
        i. Pure data calls are used by manufacturer to update software

Vulnerabilities: Telematics (authentication)

Exploit Implementation:

1. aqLink compatible software calls car until response is accepted
   a. Changes timeout of call so that buffer overflow has enough time to transmit payload

2. Exploit can also be accomplished blindly (without sniffing packets)
   a. Encoded audio file played over the call executes buffer overflow to compromise the car

Threat Assessment

Threat Assessment: Theft

- The naive adversary can use these exploits to steal a car
    - Easy to do with complete access
- However, a more clever adversary can compromise many cars in order to maximise profit
    - Track cars through GPS and identify through VIN
    - Only take advantage of ECU control on a specific target car
    - Can find cars that he/she is interested in the most

Theft

- Better yet, follow the desktop computer model: sell capability as a service
    - Provide compromised cars for sale like infected PCs
    - Customers come with car requests

- Researchers implemented this theft technique by providing an accomplice with a car
    - Attacker remotely unlocks car doors, disengages shift lock and spoofs startup protocol to start the engine
    - Accomplice drives away in their desired car

Threat Assessment: Surveillance

- By compromising a car's telematics, an adversary can record and exfiltrate the audio in the cabin

- In addition, the location of a car can be tracked through GPS

- Find targets of interest through mass exploitation and sifting
    - For example, an expensive car in a Google parking lot going home to an rich neighborhood is indicative of a person of interest.
    - With enough knowledge of a target person, their car can be identified efficiently

Suggested Solutions

- The common trait in most of the exploits identified comes through:

1. Unauthenticated interfaces that are open to unsolicited communication
    a. There is no reason to keep telnet, ftp or nc binaries to exist in production ECUs
    b. The car should not allow brute force bluetooth pairing attempts without any indication
    c. Data calls should give some indication of activity to the user (even for manufacturer calls)
2. “Security 101”
    a. Safe usage of strcpy
    b. At least use basic buffer overflow countermeasures like ASLR or stack canaries
    c. Don’t provide debugging flags or information in production ECUs

Suggestion Solutions

3. Learn from the development of computer security
    a. Don’t wait for high-profile attacks to happen before considering security
    b. Move away from the physical-access threat model because it encourages laziness

4. Better documented functionality and interfaces
    a. Most exploits arise in assumptions made by the “glue” connecting different components
        i. Manufacturers unaware of CD reflashing capability
        ii. ECUs from different manufacturers are implemented completely differently
    b. Outsourcing implementation to other companies leads to misinformation

Contributions

- The first to study to experimentally and systematically study the externally-facing attack surface of a car
- Demonstrate vulnerabilities in commonly used components and practical exploits to gain full control of all ECUs in a car
- Show that most vulnerabilities arise in the interfaces between different components

Discussion

1. How can we reconcile the need for some outsourcing of components with the information that a manufacturer needs to develop secure interfaces?
    a. For example, assumptions of packet size or undocumented features like CD reflashing.
2. With cars coming close to resembling the functionality of personal computers, why is there such a lack of foresight on the part of car manufacturers?
3. Are attacks like Bluetooth PIN brute force really feasible? Can an adversary assume short-range wireless access to the target vehicle for 10 hours?

Discussion

4. What are some limitations of the approaches in this paper?