The Car Hacker’s Handbook – Complete Learning Guide

Compiled for hands-on learning and research. Use only on vehicles and equipment you own or have explicit permission to test. Educational content adapted for practical study.

Table of Contents

1. Chapter 1: Understanding Threat Models
2. Chapter 2: Bus Protocols
3. Chapter 3: Vehicle Communication with SocketCAN
4. Chapter 4: Diagnostics and Logging
5. Chapter 5: Reverse Engineering the CAN Bus
6. Chapter 6: ECU Hacking
7. Chapter 7: Building and Using ECU Test Benches
8. Chapter 8: Attacking ECUs and Other Embedded Systems
9. Chapter 9: In-Vehicle Infotainment Systems
10. Chapter 10: Vehicle-to-Vehicle Communication
11. Chapter 11: Weaponizing CAN Findings
12. Chapter 12: Attacking Wireless Systems with SDR
13. Chapter 13: Performance Tuning
14. Appendix A: Tools of the Trade
15. Appendix B: Diagnostic Modes and PIDs
16. Appendix C: Create Your Own Open Garage
17. Practical Lab Setup

Chapter 1: Understanding Threat Models

Key Concepts

- Attack surface: all ways data enters a vehicle (external and internal).
- Threat modeling levels: Level 0 (bird’s-eye inputs), Level 1 (receivers), Level 2 (receiver internals).
- Trust boundaries: more boundaries crossed → higher risk.

Process

1. List all inputs (cellular, Wi-Fi, Bluetooth, key fob, TPMS, USB, OBD-II, sensors).
2. Map receivers (IVI/infotainment, immobilizer, ECUs, gateways).
3. Break down complex receivers (kernel-space vs user-space paths).

Risk Rating

- DREAD (Damage, Reproducibility, Exploitability, Affected users, Discoverability).
- CVSS as an alternative framework for standardized scoring.

Exercise: Draw Level 0/1/2 diagrams for a specific car. Brainstorm threats and score them with DREAD.

Chapter 2: Bus Protocols

Core Protocols

- CAN (11-bit/29-bit IDs, 2-wire differential, 120Ω termination).
- ISO-TP for multi-frame messages (diagnostics, large transfers).
- GMLAN (GM: low-speed single-wire + high-speed dual-wire).
- J1850 (PWM/VPW), KWP2000/K-Line (ISO 14230/9141-2), LIN (master/slave), MOST (multimedia), FlexRay (time-critical), Automotive Ethernet.

Finding Buses

- OBD-II pins: 6 (CANH), 14 (CANL), 2/10 (J1850), 7 (K-line), 15 (L-line), 16 (battery+).
- Voltage heuristics: CAN ~2.5V rest ±1V; LIN ~12V; PWM 5V; VPW 7V.

Exercise: Measure OBD-II voltages to identify active protocols. Map protocol-to-system usage from wiring diagrams.

Chapter 3: Vehicle Communication with SocketCAN

Setup

- Drivers: built-in CAN controllers, USB/serial CAN adapters.
- Virtual interfaces: `vcan` for safe practice.
- Utilities: `ip link`, `candump`, `cansend`, `cansniffer`, `canplayer`, `isotptun`.

Workflow

1. Bring up interface (real or vcan).
2. Sniff: record baseline traffic.
3. Replay: verify action reproducibility.
4. Filter: isolate IDs/bits for behaviors.

Exercise: Create `vcan0`, use `cansend` to transmit frames, capture with `candump`, and replay using `canplayer`.

Chapter 4: Diagnostics and Logging

Diagnostics

- DTCs (Diagnostic Trouble Codes): format and reading/erasing.
- UDS over ISO-TP: modes, PIDs, security access.
- Keeping diagnostic sessions alive; brute-forcing modes safely.

Event Data

- Recorder logging: ethical, legal, and technical considerations.

Exercise: Query PIDs on a test bench; practice UDS session control on a bench ECU (never on-road).

Chapter 5: Reverse Engineering the CAN Bus

Methods

- Record → Replay to confirm causality.
- Binary search on logs to find minimal triggering frame(s).
- Bit/byte diffing when physical actions occur.

Exercise: Identify door lock/unlock frames on a bench network. Map arbitration IDs to functions and log them.

Chapter 6: ECU Hacking

Targets & Techniques

- Bootloaders, firmware extraction, flashing procedures.
- Debug ports (UART, JTAG, SWD), tear-down and safe probing.

- Security access levels, seed-key, signature checks.

Only work on bench ECUs or authorized targets. Firmware tampering can brick devices or create unsafe conditions.

Exercise: Read firmware via bench setup, identify update formats, and verify signatures where applicable.

Chapter 7: Building and Using ECU Test Benches

Design

- Harnesses, breakout boxes, fusing, power supplies with current limiting.
- Network simulation: multiple ECUs, gateway behavior, termination.

Exercise: Assemble a basic CAN bench with an ECU, proper termination, and a CAN interface. Validate with `candump`.

Chapter 8: Attacking ECUs and Other Embedded Systems

Approach

- Static/dynamic analysis of firmware; common memory protections.
- Input parsing issues (buffers, state machines, timing).
- Persistence and rollback protections.

Exercise: Fuzz a non-safety ECU service on a bench via ISO-TP; monitor for crashes or resets.

Chapter 9: In-Vehicle Infotainment (IVI) Systems

Surfaces

- Media parsers, Bluetooth/Wi-Fi stacks, app platforms, update channels.
- Bridges from IVI to vehicle networks; hardening gateways.

Exercise: Analyze an IVI update package format and verify signature enforcement; test in a VM when possible.

Chapter 10: Vehicle-to-Vehicle (V2V) Communication

Concepts

- V2V/V2X stacks (DSRC/C-V2X), timing, identity, and privacy.
- Threats: spoofing, replay, jamming, and misbehavior detection.

Exercise: Simulate message flows in a lab with SDR or software stacks; explore authentication models conceptually.

Chapter 11: Weaponizing CAN Findings

From Research to Proof-of-Concept

- Design safe, minimal PoCs; avoid impacting safety systems.
- Document IDs, data formats, preconditions, and fail-safes.
- Coordinated disclosure and ethical considerations.

Exercise: Create a benign PoC (e.g., toggle a non-critical indicator) on a bench with a clear safety checklist.

Chapter 12: Attacking Wireless Systems with SDR

Targets

- Key fobs, TPMS, remote services, cellular modems.
- Legal spectrum use and strict adherence to regulations.

Exercise: Passive capture of TPMS transmissions in a Faraday-safe setup to study protocol framing (receive-only).

Chapter 13: Performance Tuning

Principles

- Calibration vs hacking; legality, emissions, and safety.
- Reading maps, tables, and applying safe, reversible changes.

Never apply performance changes to on-road vehicles without compliance and safety validation.

Exercise: Analyze a sample calibration in a simulator tool; document any change impacts.

Appendix A: Tools of the Trade

- Hardware: CAN interfaces (USB adapters), power supplies, multimeter, oscilloscope/logic analyzer, soldering/probing tools.
- Software: can-utils, SocketCAN, Wireshark (CAN dissectors), Kayak, firmware analysis suites, SDR stacks.

Appendix B: Diagnostic Modes and PIDs

- Generic OBD-II modes/PIDs and manufacturer-specific extensions.
- UDS services: session control, security access, read/write data by identifier.

Appendix C: Create Your Own Open Garage

- Community lab guidelines, safety culture, documentation standards, and collaboration.

Practical Lab Setup

Minimum Bench

- 12V current-limited supply with fuses, ECU with harness, proper CAN termination.
- Interface (USB-CAN), host laptop with SocketCAN, can-utils, and logging.

Safety Checklist

- Never connect to on-road vehicles when testing unknown frames.
- Use Faraday shielding for RF experiments; respect spectrum laws.
- Log every action; design reversible changes; back up firmware/configs.

This guide is for educational purposes. Always get permission, follow the law, and prioritize safety.