INDEX

Numbers

802.11p standard, 179–180, 184
8devices USB2CAN converter, 244
1609.x standard, 179–180, 184

A

ACM (airbag control module), 61
ACN (automated crash notification) systems, 64
Advanced Linux Sound Architecture (ALSA) framework, 26
Advanced User Debugger (AUD), 133–134
airbag control module (ACM), 61
ALSA (Advanced Linux Sound Architecture) framework, 26
amplified relay attacks, PKES systems, 220
amplitude-shift keying (ASK) modulation, 210–211
analyze.exe tool, 100
anonymous certificates, 189
application-specific integrated circuits (ASICs), 95
apps (IVI system), 163
arbitration IDs
  defined, 18
  finding, 79–80
  grouping streamed data, 70–71
Arduino shields, 242
Armengaud, Eric, 30
`asc2log` tool (`can-utils` package), 41
ASICs (application-specific integrated circuits), 95
ASIL (Automotive Safety Integrity Level) system, 11, 13
ASK (amplitude-shift keying) modulation, 210–211
assembly code
  converting C code to, 196–198
  converting to shellcode, 199
asynchronous channel, MOST bus protocol, 25
AUD (Advanced User Debugger), 133–134
Audio Video Bridging (AVB) standard, 31
autoignition (detonation), 235
automated crash notification (ACN) systems, 64
Automotive Ethernet bus protocol, 30–31
automotive racing, 233
Automotive Safety Integrity Level (ASIL) system, 11, 13
auxiliary jacks (IVI systems), 158
AVB (Audio Video Bridging) standard, 31
AVR systems, resetting, 143
AVRDUDESS GUI, 137, 139–140, 251

B

backdoor attacks, 95
BCM (broadcast manager) service, 45, 46
`bcmserver` tool (`can-utils` package), 41
BerliOS, 35
best master clock algorithm, 31
BGE (Bus Guardian Enable), 30
.bin files, 160

The Car Hacker's Handbook: A Guide for the Penetration Tester
© 2016 Craig Smith

Binary Editor, 238
`binwalk` tool, 160
bird's eye view (Level 0) threats, 3, 6–7
bitmasks, 71–72
Bluetooth connection, 9, 164, 166–167, 212
Bluez daemon, 10
BMEP (brake mean effective pressure), 235
Boone, Jared, 213
Boothe, Peter, 228
bootloaders, brute-forcing, 138–148
brake mean effective pressure (BMEP), 235
bricking, 89
broadcast manager (BCM) service, 45, 46
brute-forcing
  diagnostic modes, 58–60
  key code, 217
  keypad entry, 228–230
  secure bootloaders, 138–148
BURN2 programmer, 236
Bus Guardian, 30
Bus Guardian Enable (BGE), 30
Bus Pirate cable, 131
bus protocols, 15–16. *See also names of specific protocols*
  Automotive Ethernet, 30–31
  Controller Area Network, 16–20
  FlexRay, 27–30
  ISO 9141-2, 23
  Keyword Protocol 2000, 22–23
  Local Interconnect Network, 24
  Media Oriented Systems Transport, 24–27
  OBD-III, 33–34
  SAE J1850, 20–22

C

C code, 194–202
c0f (CAN of Fingers) tool, 205–207, 250
.cab files, 160
California Air Resources Board (CARB), 33
CAMP (Crash Avoidance Metrics Partnership), 186–187
CAMs (cooperative awareness messages), 181–183
CAN (Controller Area Network) bus protocol. *See also* reverse engineering CAN bus
  CANopen protocol, 20
  differential signaling, 16–17
  extended packets, 19
  finding connections, 17–18
  GMLAN, 20
  ISO 15765-2, 19–20
  OBD-II connector, 17
  standard packets, 18–19
  vulnerabilities, 10
CAN bus Y-splitter, 245
CAN devices
  Arduino shields, 242
  CAN bus Y-splitter, 245
  CAN232 dongle, 244
  CANBus Triple board, 245
  CANtact, 242–243
  CANUSB dongle, 244
  ChipKit board, 243
  ChipWhisperer, 246
  CrossChasm C5 data logger, 245
  ELM327 chipset, 243–244
  ELM-USB connector, 244
  EVTV due board, 244–245
  Freematics OBD-II Telematics Kit, 242
  GoodThopter board, 244
  HackRF SDR, 245
  Raspberry Pi, 243
  Red Pitaya board, 246
  serial, 39–40
  setting up `can-utils` to connect to, 36
  USB2CAN converter, 244
  USRP SDR, 246
  ValueCAN, 252
  VSCOM adapter, 244

The Car Hacker's Handbook: A Guide for the Penetration Tester
© 2016 Craig Smith

CAN high (CANH) wires, 16–17
CAN low (CANL) wires, 16–17
CAN network. *See also* CAN bus protocol; reverse engineering CAN bus
  locating, 67–68
  sending data with, 55
  virtual, 40–41
CAN of Fingers (c0f) tool, 205–207, 250
`can0` device, 38
CAN232 dongle, 244
Canberry controller, 243
CANBus Control Panel, 82–83
CANBus Triple board, 245
`canbusload` tool (`can-utils` package), 41
`can-calc-bit-timing` command (`can-utils` package), 41
`can_dev` module, 37–38
CANdiy-shield, 242
`candump` utility (`can-utils` package), 41, 70
`canfdtest` tool (`can-utils` package), 42
`cangen` command (`can-utils` package), 42
`cangw` tool (`can-utils` package), 42
CANH (CAN high) wires, 16–17
CANiBUS server, 248
`can-isotp.ko` module (`can-utils` package), 43–44
CANL (CAN low) wires, 16–17
`canlogserver` utility (`can-utils` package), 42
CANopen protocol, 20
`canplayer` command (`can-utils` package), 42
`cansend` tool (`can-utils` package), 42
`cansniffer` tool (`can-utils` package), 42, 71–72
CANtact, 242–243
CANUSB dongle, 244
`can-utils` package, 20
  `asc2log` tool, 41
  `bcmserver` tool, 41
  `canbusload` tool, 41
  `can-calc-bit-timing` command, 41
  `candump` utility, 41
  `canfdtest` tool, 42
  `cangen` command, 42
  `cangw` tool, 42
  `can-isotp.ko` module, 43–44
  `canlogserver` utility, 42
  `canplayer` command, 42
  `cansend` tool, 42
  cansniffer, 42
  configuring built-in chipsets, 37–38
  configuring serial CAN devices, 39–40
  finding door-unlock control, 77–78
  installing, 36–37
  installing additional kernel modules, 42–43
  `isotpdump` tool, 42
  `isotprecv` utility, 42
  `isotpsend` command, 42
  `isotpserver` tool, 42
  isotpsniffer, 42
  `isotptun` utility, 42
  `log2asc` tool, 42
  `log2long` command, 42
  recording and playing back packets, 73
  setting up virtual CAN network, 40–41
  `slcan_attach` tool, 42
  `slcand` daemon, 42
  `slcanpty` tool, 42
CARB (California Air Resources Board), 33
Carberry controller, 243
CaringCaribou (CC), 58–60, 249
CAs (certificate authorities), 188
CC (CaringCaribou), 58–60, 249
CDR (crash data retrieval) tools, 62
cellular networks
  V2V communication and, 178
  vulnerabilities, 7–8
certificate authorities (CAs), 188

The Car Hacker's Handbook: A Guide for the Penetration Tester
© 2016 Craig Smith

certificate provisioning, 189–190
certificate revocation list (CRL), 190, 191–192
Character Sheet, Open Garages, 255–259
chip tuning. *See also* reverse engineering CAN bus
  EPROM programmers, 236–237
  ROM emulators, 237–238
ChipKit board, 243
chipping process, 236
chipsets
  configuring, 37–38
  identifying, 128–130
ChipWhisperer, 134–135, 246
  ChipWhisperer ADC, 143–144
  installing, 135–137
  Main Window settings for clock-glitch attack, 151
  prepping Victim Board, 137–138
  scripting with Python, 147–148
  setting up for serial communication, 140–141
Chrysler
  SAE J1850 protocol, 20
  VPW protocol, 22
ChuangZhou CAN-Bus shield, 242
circuit boards
  chips, 128–130
  model numbers, 128
CKP (crankshaft position), 121–122, 124
clock glitching, 148–154
COB-ID (communication object identifier), 20
code analysis, 106–107
  interactive disassemblers, 110–112
  plain disassemblers, 107–110
codes, DTC, 52–53
coding SocketCAN applications
  connecting to CAN socket, 44–45
  procfs interface, 45–46
  setting up CAN frame, 45
common vulnerability scoring system (CVSS), 13
communication object identifier (COB-ID), 20
connectors (IVI system), 166–170
control blocks, MOST bus protocol, 25–26
control channel, MOST bus protocol, 25
Controller Area Network bus protocol. *See* CAN (Controller Area Network) bus protocol
cooperative awareness messages (CAMs), 181–183
crankshaft position (CKP), 121–122, 124
Crash Avoidance Metrics Partnership (CAMP), 186–187
crash data retrieval (CDR) tools, 62
CRC32 hash, 162
`crc32` tool, 162
creative packet analysis, 76–80
CRL (certificate revocation list), 190, 191–192
CrossChasm C5 data logger, 245
`ctrl_tx` utility, 26
CVSS (common vulnerability scoring system), 13
cycles, FlexRay, 28–29

D

.dat files, 160
data length code (DLC), 19
data visualization tools, 100
DB9-to-OBDII connector, 32–33
debugging hardware
  Advanced User Debugger, 133–134
  JTAG protocol, 130–132
  Nexus, 133–134
  Serial Wire Debug, 132–133
decentralized environmental notification messages (DENMs), 183–184

The Car Hacker's Handbook: A Guide for the Penetration Tester
© 2016 Craig Smith

dedicated short-range communication protocol. *See* DSRC (dedicated short-range communication) protocol
definitions (*def*) file, 239
DENMs (decentralized environmental notification messages), 183–184
Department of Defense (DoD) threat rating system, 13
detonation (autoignition), 235
device under test (DUT), 137–138
DFRobot CAN-Bus shield, 242
diagnostic link connector (DLC), 17, 51, 119. *See also* diagnostics and logging
diagnostic trouble codes. *See* DTCs
diagnostics and logging, 51–65
  automated crash notification systems, 64
  diagnostic trouble codes, 33, 52–54
  event data recorder, 61–63
  malicious intent, 64–65
  Unified Diagnostic Services, 54–61
dictionary attacks, 218
differential signaling, 16
DIP (dual in-line package) chips, 236
disassemblers
  Dis51, 106
  Dis66k, 106
  interactive, 110–112
  plain, 107–110
disassembling IVI unit, 168
DLC (data length code), 19
DLC (diagnostic link connector), 17, 51, 119. *See also* diagnostics and logging
.*dll* files, 160
DoD (Department of Defense) threat rating system, 13
door-unlock control
  finding with `can-utils` package, 77–78
  finding with Kayak, 76–77
DREAD rating system, 11–13
DSRC (dedicated short-range communication) protocol, 179–180
  defined, 178
  features and uses, 180–181
  roadside systems, 181–184
  tracking vehicles with, 186
  WAVE standard, 184–186
DST-40 algorithm, 225–226
DST-80 algorithm, 226
DTCs (diagnostic trouble codes)
  codes, 52–53
  erasing, 54
  faults, 52
  OBD-III standard and, 33
  scan tools, 54
dual in-line package (DIP) chips, 236
dumping transponder memory, 218
DUT (device under test), 137–138
DVD checks (IVI system), 164–165
dynamic segment (FlexRay cycles), 28, 30

E

ECU (engine/electronic control unit). *See also* ECU hacking; ECU test benches; embedded systems
  block diagrams, 118–119
  finding, 116–117
  pinouts, 118
  TPMS connection, 8–9
  tuning, 235–239
ECU hacking, 91–92
  backdoor attacks, 95
  code analysis, 106–112
  exploits, 95–96
  front door attacks, 92–95
  reversing firmware, 96–105

The Car Hacker's Handbook: A Guide for the Penetration Tester
© 2016 Craig Smith

ECU test benches, 115–126
  hall effect sensors, 121–122
  simulating sensor signals, 120–121
  simulating vehicle speed, 123–126
ECU tuning, 235–236
  chip tuning, 236–238
  flash tuning, 238–239
EDR (event data recorder), 61–62
  reading data from, 62
  restraint control module, 63
  SAE J1698 standard, 63
  sensing and diagnostic module, 63
Ege, Barış, 222
electronic control unit. *See* ECU (engine/electronic control unit)
electronic controllers, 91. *See also* ECU hacking
ELLSI (Ethernet low-level socket interface), 158
ELM327 chipset, 54, 243–244
ELM-USB connector, 244
EM Micro Megamos algorithm, 221–223
EM4237 algorithm, 223
embedded systems, 127. *See also* wireless systems
  circuit boards, 128–130
  debugging hardware, 130–134
  fault injection, 148–156
  power-analysis attacks, 138–148
  side-channel analysis, 134–138
emissions, performance tuning and, 234–235
EMS PCMCIA card, 37
end-of-data (EOD), VPW protocol, 22
engine control unit. *See* ECU (engine/electronic control unit)
EOD (end-of-data), VPW protocol, 22
epidemic distribution model, 191
EPROM programmers, 236–237
Ethernet, 30–31, 158
Ethernet low-level socket interface (ELLSI), 158
ETSI (European Telecommunications Standards Institute)
  cooperative awareness messages, 181–183
  decentralized environmental notification messages, 183–184
Ettus Research, 210
European DSRC system, 180–181
European Telecommunications Standards Institute. *See* ETSI
Evenchick, Eric, 242
event data recorder. *See* EDR (event data recorder)
events
  event data recorder, 61–63
  triggering with TPMS, 214–215
EVTV due board, 244–245
EVTV.me, 248
.exe files, 160
exploits, 95–96
  responsible exploitation, 208
  writing in C code, 194–202
extended packets, CAN bus protocol, 19

F

fault injection
  clock glitching, 148–154
  defined, 148
  invasive, 156
  power glitching, 156
  setting trigger line, 154–155
faults, 52
field-programmable gate array (FPGA) board, 149, 225
`file` command, 160
fire-and-forget structure (CAN packets), 55
firmware, reversing, 96–105
flash tuning (flashing), 238–239

The Car Hacker's Handbook: A Guide for the Penetration Tester
© 2016 Craig Smith

FlexRay bus protocol, 27–30
cycles, 28–29
hardware, 27
network topology, 27
packet layout, 29–30
sniffing, 30
time division multiple access scheme, 27–28
Ford Motor Company
MAF transfer graph, 98
OpenXC, 84–88
PWM protocol, 21
restraint control module, 63
forged packets, sending with TPMS, 215
forward-prediction attacks, 218
FPGA (field-programmable gate array) board, 149, 225
frame ID, FleyRay packet, 30
Freematics OBD-II Telematics Kit, 242
freeze frame data, 52
frequency-shift keying (FSK) modulation, 211
front door attacks
J2534-1 standard, 92–93
KWP2000, 94
seed-key algorithms, 94–95
FSK (frequency-shift keying) modulation, 211
Future Technology Devices International, Ltd (FTDI), 39
fuzzing, 64, 88

G

Garcia, Flavio D., 222, 225
General Motors
GMLAN bus, 20
pinout, 31–32
SAE J1850 protocol, 20
sensing and diagnostic module, 63
VPW protocol, 22
Generalized Vehicle Reverse Engineering Tool (GVRET), 245
glitching
clock, 148–154
defined, 148
invasive, 156
power, 156
setting trigger line, 154–155
GMLAN bus, 20
GNU binutils disassembler, 106
GNU Radio Companion (GRC), 210, 216
Go language, 248
Goodspeed, Travis, 244
GoodThopter board, 244
Gqrx SDR, 216
GRC (GNU Radio Companion), 210, 216
Great Scott Gadgets, 210, 245
GVRET (Generalized Vehicle Reverse Engineering Tool), 245

H

HackRF One, 210
HackRF SDR, 245
Hall effect sensors, 121–122
hard (permanent) DTCs, 54
hard faults, 52
hardware
Arduino shields, 242
attacking IVI system via, 166–170
CAN bus Y-splitter, 245
CAN232 dongle, 244
CANBus Triple board, 245
CANtact, 242–243
CANUSB dongle, 244
ChipKit board, 243
ChipWhisperer, 246
CrossChasm C5 data logger, 245
debugging, 130–134
ELM327 chipset, 243–244
ELM-USB connector, 244
EVTV due board, 244–245
FlexRay bus protocol, 27
Freematics OBD-II Telematics Kit, 242
GoodThopter board, 244

The Car Hacker's Handbook: A Guide for the Penetration Tester
© 2016 Craig Smith

hardware, *continued*
  HackRF SDR, 245
  MegaSquirt, 239–240
  Raspberry Pi, 243
  Red Pitaya board, 246
  USB2CAN converter, 244
  USRP SDR, 246
  VSCOM adapter, 244
hashing, 162–163
header bits (VPW protocol), 22
header CRC (FlexRay packet), 29, 30
hex editors, 100
high-speed CAN (HS-CAN) lines, 18, 32, 38
high-speed synchronous interface (HSI), 10, 13–14
Hitag 1 algorithm, 224
Hitag 2 algorithm, 224–225
Hitag AES algorithm, 225
Horauer, Martin, 30
hotwiring, 230
HS-CAN (high-speed CAN) lines, 18, 32, 38
HSI (high-speed synchronous interface), 10, 13–14
hybrid approach, V2V communication, 178

I

ICSim (instrument cluster simulator)
  changing difficulty of, 84
  reading CAN traffic on, 83
  setting up, 81–83
IDA Pro disassembler, 106, 110
identifier extension (IDE), 19
idle segment (FlexRay cycles), 28
IEEE 802.1AS standard, 31
IFR (in-frame response) data, VPW protocol, 22
Immo Emulators, 228
immobilizer systems, 220–221
  defined, 8
  DST-40, 225–226
  DST-80, 226
  EM Micro Megamos, 221–223
  EM4237, 223
  Hitag 1, 224
  Hitag 2, 224–225
  Hitag AES, 225
  Keeloq, 226–227
  Open Source Immobilizer Protocol Stack, 227
  physical attacks on, 228–230
infotainment console, 5–6, 9. *See also* IVI system
in-frame response (IFR) data, VPW protocol, 22
instrument cluster simulator. *See* ICSim
intelligent transportation system, 177
interactive disassemblers, 110–112
interactive probing method, for determining vehicle make, 203–204
internal network controls (IVI systems), 158
Intrepid Control Systems, 252
invasive fault injection, 156
in-vehicle infotainment system. *See* IVI (in-vehicle infotainment) system
IPv4 passive fingerprinting, 205
IPv6 protocol, 185
ISO 15765-2 (ISO-TP) protocol, 19–20, 55
ISO 26262 ASIL rating system, 11, 13
ISO 9141-2 (K-Line) bus protocol, 23
ISO-TP (ISO 15765-2) protocol, 19–20, 55
`isotpdump` tool (`can-utils` package), 42
`isotprecv` utility (`can-utils` package), 42
`isotpsend` command (`can-utils` package), 42
`isotpserver` tool (`can-utils` package), 42
`isotpsniffer` (`can-utils` package), 42
`isotptun` utility (`can-utils` package), 42

The Car Hacker's Handbook: A Guide for the Penetration Tester
© 2016 Craig Smith

IVI (in-vehicle infotainment) system, 157–158
  acquiring OEM system for testing, 174–175
  attack surfaces, 158
  attacking hardware, 166–170
  attacking through update system, 158–165
  test benches, 170–174

J

J2534-1 standard, 92
  shims, 93
  sniffers and, 93
  tools, 93
jamming signal, key fobs, 216–217
JSON format, 86
JTAG protocol
  debugging with, 131–132
  defined, 130
  JTAGulator, 131
JTAGulator, 131

K

Kamkar, Samy, 217
Kayak, 248
  finding arbitration IDs, 79–80
  finding door-unlock control, 76–77
  recording and playing back packets, 73–75
  socketcand and, 46–49
Keeloq algorithm, 226–227
kernel device manager (udev), 11
key fobs, 215–216
  amplified relay attack, 220
  brute-forcing key code, 217
  dictionary attacks, 218
  dumping transponder memory, 218
  forward-prediction attacks, 218
  jamming signal, 216–217
  passive keyless entry and start systems, 219–220
  pulling response codes, 217
  reversing CAN bus, 218–219
  transponder duplication machines, 219
  vulnerabilities, 8
keyslot-only state (FlexRay cycles), 29
Keyword Protocol 2000 (KWP2000) bus protocol, 22–23, 94
Kidder, Collin, 248
K-Line (ISO 9141-2) bus protocol, 23
Komodo CAN bus sniffer, 251–252
Kvaser Driver, 11
KWP2000 (Keyword Protocol 2000) bus protocol, 22–23, 94

L

LA (linkage authority), 192
LAWICEL AB, 244
LAWICEL protocol, 242, 244
Level 0 (bird's eye view) threats, 3, 6–7
Level 1 (receivers) threats, 4, 7–10
Level 2 (receiver breakdown) threats, 5–6, 10–11
LF (low-frequency) RFID chip, 219
library procedures, 97
LIN (Local Interconnect Network) bus protocol, 24
linkage authority (LA), 192
Linux. *See also* SocketCAN
  Automotive Grade Linux system, 173–174
  ELM327 chipset and, 243–244
  FlexRay network and, 30
  GENIVI system and, 170–173
  hashing tools, 162
  ICSim, 81–84
  infotainment systems, 5–6
  installing ChipWhisperer software, 135–137
  most4linux project, 26–27
  Raspberry Pi, 243
  tools, 162, 247
LNA (low-noise amplifier), 213
Local Interconnect Network (LIN) bus protocol, 24
location obscurer proxy (LOP), 190
`log2asc` tool (`can-utils` package), 42

The Car Hacker's Handbook: A Guide for the Penetration Tester
© 2016 Craig Smith

log2long command (can-utils package), 42
long-term certificate (LTC), 188
LOP (location obscurer proxy), 190
low-frequency (LF) RFID chip, 219
low-noise amplifier (LNA), 213
low-number-of-coldstarters state (FlexRay cycles), 29
low-speed CAN (LS-CAN) lines, 18, 32, 38
LTC (long-term certificate), 188

M

MA (misbehavior authority), 192
macroticks, 28
MAF (mass air flow) sensor, 97
malfunction indicator lamp (MIL), 51, 52
malicious intent, 64–65
Manchester encoding, 214
mass air flow (MAF) sensor, 97
MCU (microcontroller unit), 101, 120
MD5 hash, 162
md5sum tool, 162
Media Oriented Systems Transport bus protocol. *See* MOST (Media Oriented Systems Transport) bus protocol
Megamos cryptographic system, 221–222
MegaSquirt hardware, 239–240
Meier, Jan-Niklas, 41, 46
memory chips, 95
Metasploit, 193–194, 200–202
microcontroller unit (MCU), 101, 120
mid-speed CAN (MS-CAN) lines, 18
MIL (malfunction indicator lamp), 51, 52
MIL-STD-882E rating system, 11, 13
misbehavior authority (MA), 192
misbehavior reports, V2V communication, 192
Moates ROM adapter board, 237
model numbers, circuit boards, 128
modes, diagnostic code, 57–60, 253–254
MOST (Media Oriented Systems Transport) bus protocol, 24–25
  control blocks, 25–26
  hacking, 26–27
  network layers, 25
most4linux driver, 26–27
most_aplay utility, 26
MS-CAN (mid-speed CAN) lines, 18
MultiTarget Victim Board
  ChipWhisperer, 135
  set for glitching, 149
Murphy, Austin, 246

N

NAD (node address for diagnostics), 24
National Highway Traffic Safety Administration (NHTSA), 62
NavTeq infotainment unit, 159
NeoIV devices, 252
network layers, MOST bus protocol, 25
network sniffers. *See* sniffers
NewAE Technologies, 245
Nexus interface, 133–134
NHTSA (National Highway Traffic Safety Administration), 62
Nissan
  MAF VQ graph, 98
  plain dissassembly of 1990 300ZX Twin Turbo ROM, 107–110
NLFSR (nonlinear feedback shift register), 226
node address for diagnostics (NAD), 24
nonlinear feedback shift register (NLFSR), 226

The Car Hacker's Handbook: A Guide for the Penetration Tester
© 2016 Craig Smith

no-operation instructions (NOPs), 164
NULL values, removing from code, 199–200

O

O2OO data logger, 249
OBD2 ScanTool, 246
OBD-II connector, 17, 51, 119. *See also* diagnostics and logging
OBD-III bus protocol, 33–34
OBDTester.com, 244
Octane CAN bus sniffer, 250
OEM (original equipment manufacturer)
  front door attacks, 92
  testing IVI system, 174–175
OLS300 emulator, 238
on-off keying (OOK), 211
Open Garages, 81, 205, 241, 248, 255–259
Open Source development site, 35
Open Source Immobilizer Protocol Stack, 227
Open Systems Interconnection (OSI) model, 25
OpenXC, 84–85
  hacking, 87–88
  translating CAN bus messages, 85–86
  writing to CAN bus, 86
optical glitches, 132
original equipment manufacturer. *See* OEM (original equipment manufacturer)
OSI (Open Systems Interconnection) model, 25
Ostrich2 emulator, 237

P

parameter IDs (PIDs), 57–60, 254
passband, RFID receiver, 216
passive CAN bus fingerprinting, 204–207
passive keyless entry and start (PKES) systems, 219–220
passwords
  monitoring power usage when entering, 145–147
  setting custom password, 141–143
payload length, FlexRay packet, 30
payloads, 193–194, 200–202. *See also* weaponizing CAN findings
PC (pseudonym certificate), 189
PCA (Pseudonym Certificate Authority), 190
PCM (powertrain control module), 33, 51
PEAK-System PCAN-USB adapter, 38
performance tuning, 233–234
  ECU tuning, 235–239
  stand-alone engine management, 239–240
  trade-offs, 234–235
permanent (hard) DTCs, 54
`PF_CAN` protocol family, 36
PICAN CAN-Bus board, 243
PIDs (parameter IDs), 57–60, 254
PKES (passive keyless entry and start) systems, 219–220
PKI (public key infrastructure) systems, 188
  anonymous certificates, 189
  certificate provisioning, 189–190
  certificate revocation list, 191–192
  misbehavior reports, 192
  vehicle certificates, 188–189
plain disassemblers, 107–110
plastic optical fiber (POF), 24–25
plug-ins (IVI system), 163
PoC (proof-of-concept) broadcast manager server, 41
POF (plastic optical fiber), 24–25
potentiometers, 120
power glitching, 156
power-analysis attacks, 138–148, 227

The Car Hacker's Handbook: A Guide for the Penetration Tester
© 2016 Craig Smith

powertrain control module (PCM), 33, 51
PRF (pseudorandom function), 220
PRNG (pseudorandom number generator), 218, 220
procfs interface, 45–46
proof-of-concept (PoC) broadcast manager server, 41
pseudonym certificate (PC), 189
Pseudonym Certificate Authority (PCA), 190
pseudorandom function (PRF), 220
pseudorandom number generator (PRNG), 218, 220
public key infrastructure systems. See PKI (public key infrastructure) systems
pulse width modulation (PWM) protocol, 21
PyOBD module, 246–247
Python
  CaringCaribou, 58–60, 249
  scripting ChipWhisperer with, 147–148

Q

QoS (quality of service), 31
quadlets, 26

R

RA (Registration Authority), 189
radare2 disassembler, 163
radio-frequency identification (RFID), 215
randomize option, ICSim, 84
ransomware, 7
Raspberry Pi, 243
rating systems, threat
  CVSS system, 13
  DREAD system, 11–13
RCM (restraint control module), 63
`ReadDataByID` command, 61
receiver breakdown (Level 2) threats, 5–6, 10–11
receivers (Level 1) threats, 4, 7–10
Red Pitaya board, 246
Registration Authority (RA), 189
relay attacks, PKES systems, 219–220
remote transmission request (RTR), 19
Renesas automotive chipset, 133
response codes, pulling, 217
restraint control module (RCM), 63
reverse engineering CAN bus
  `candump` tool, 70
  creative packet analysis, 76–80
  fuzzing, 88
  grouping streamed data, 70–73
  instrument cluster simulator, 81–84
  key fobs, 218–219
  locating CAN network, 67–68
  with OpenXC, 84–88
  recording and playing back packets, 73–75
  troubleshooting, 89
  Wireshark, 69
reversing firmware
  comparing bytes, 101–103
  identifying tables, 97–101
  library procedures, 97
  microcontroller unit, 101
  ROM data, 103–105
  self-diagnostic system, 96–97
  WinOLS, 103–105
RFID (radio-frequency identification), 215
RoadRunner emulator, 238
roadside DSRC systems
  cooperative awareness messages, 181–183
  decentralized environmental notification messages, 183–184
ROM data, 103–105
ROM emulators, 237–238
RomRaider, 238–239, 251
RTR (remote transmission request), 19

The Car Hacker's Handbook: A Guide for the Penetration Tester
© 2016 Craig Smith

S

SAE J1850 bus protocol, 20–21
  event data recorder, 63
  pulse width modulation, 21
  variable pulse width, 22
SavvyCAN, 248–249
SCMS (Security Credentials Management System), 188
Scope Tab settings, ChipWhisperer ADC, 143–144
SDK (software development kit), 164
SDM (sensing and diagnostic module), 63
SDR (software-defined radio), 210
  Gqrx, 216
  HackRF, 245
  signal modulation, 210–211
  tracking vehicles with, 186
security through obscurity, 220
Security Credentials Management System (SCMS), 188
`SecurityAccess` command, 61
seed-key algorithms, 94–95
SeeedStudio SLD01105P CAN-Bus shield, 242
self-diagnostic system, 96–97
sensing and diagnostic module (SDM), 63
sensor signals, simulating, 120–121
SensorID, TPMS packet, 213–214
serial CAN devices, 39–40
Serial Wire Debug (SWD), 132–133
SHA-1 hash, 162
`sha1sum` tool, 162
shellcode, 194
shims, J2534-1 standard, 93
signal generators, 126
signal modulation, SDR, 210
  amplitude-shift keying, 210–211
  frequency-shift keying, 211
simulating
  sensor signals, 120–121
  vehicle speed, 123–126
`slcan_attach` tool (`can-utils` package), 42
`slcand` daemon (`can-utils` package), 39–40, 42
`slcanpty` tool (`can-utils` package), 42
sniffers
  cansniffer, 42
  FlexRay bus protocol, 30
  fuzzing and, 88
  isotpsniffer, 42
  J2534-1 standard and, 93
  Komodo CAN bus, 251–252
  Octane CAN bus, 250
  WAVE packets and, 179
SocketCAN, 35–36, 247
  `can-utils`, 36–44
  coding applications, 44–46
  Kayak, 46–49
  socketcand daemon, 46
socketcand daemon, 46
soft faults, 52
software. *See also names of specific software*
  AVRDUDESS GUI, 251
  CAN of Fingers, 205–207, 250
  CANiBUS server, 248
  CaringCaribou, 58–60, 249
  Kayak, 248
  Komodo CAN bus sniffer, 251–252
  Linux tools, 247
  O2OO data logger, 249
  Octane CAN bus sniffer, 250
  PyOBD module, 246–247
  RomRaider, 251
  SavvyCAN, 248–249
  UDSim ECU simulator, 250
  Vehicle Spy, 252
  Wireshark, 246
software development kit (SDK), 164
software-defined radio. *See* SDR (software-defined radio)
SparkFun SFE CAN-Bus shield, 242
splash screen, modifying, 161
spoofing packets, 30
SRR (substitute remote request), 19
stand-alone engine management, 239–240
standard packets, 18–19

The Car Hacker's Handbook: A Guide for the Penetration Tester
© 2016 Craig Smith

static segment (FlexRay cycles), 28, 30
status bits (FlexRay packet), 29
Steininger, Andreas, 30
ST-Link, 132–133
STM32F4 chips, 132
STM32F407Vx chips, 129
Subaru, 238, 251
substitute remote request (SRR), 19
SWD (Serial Wire Debug), 132–133
symbol window segment (FlexRay cycles), 28, 29
SYNC field, LIN protocol, 24
synchronous channel, MOST bus protocol, 25
`sync_rx` utility, 27
`sync_tx` utility, 27
system updates, attacking IVI system via, 158–165

T

tables, identifying, 97–101
tachometers, 77–79
Tactrix OpenPort 2.0, 238
TCM (transmission control module), 91. *See also* ECU hacking
TCU (transmission control unit), 91. *See also* ECU hacking
TDMA (time division multiple access) scheme, 27–28
test benches (IVI system)
  Automotive Grade Linux, 173–174
  GENIVI Meta-IVI, 170–173
threat modeling, 1
  attack surfaces, 2
  handling results, 13–14
  Level 0 (bird's eye view), 3, 6–7
  Level 1 (receivers), 4, 7–10
  Level 2 (receiver breakdown), 5–6, 10–11
  rating systems, 11–13
  threat identification, 6–11
time division multiple access (TDMA) scheme, 27–28
time-dependent signaling, 22
TinySafeBoot, 139
tire pressure monitor sensor. *See* TPMS (tire pressure monitor sensor)
tools. *See also names of specific tools*
  hardware, 241–246
  software, 246–252
TPMS (tire pressure monitor sensor) , 211–212
  activating signal, 214
  eavesdropping on, 212–213
  exploiting connection, 8–9
  packets, 213–214
  sending forged packets, 215
  tracking vehicles with, 214
  triggering event with, 214–215
tracking vehicles
  with DSRC protocol, 186
  with TPMS, 214
trade-offs, performance tuning, 234–235
transmission control module (TCM), 91. *See also* ECU hacking
transmission control unit (TCU), 91. *See also* ECU hacking
transponder duplication machines, 219
transponders, 33–34, 218. *See also* key fobs
TREAD (Transportation Recall Enhancement, Accountability, and Documentation) Act of 2000, 212
trigger wheel, 122
troubleshooting, 89. *See also* diagnostics and logging
TunerStudio tool, 240
tuning, 233–234
  ECU tuning, 235–239
  stand-alone engine management, 239–240
  trade-offs, 234–235

The Car Hacker's Handbook: A Guide for the Penetration Tester
© 2016 Craig Smith

U

UART protocol, 23
udev (kernel device manager), 11
UDS (Unified Diagnostic Services), 54–55
  error responses, 55–57
  keeping vehicle in diagnostic state, 60–61
  modes and PIDS, 57–60
  sending data, 55–57
UDSim ECU simulator, 250
ultra-high-frequency (UHF) signal, 219
Unified Diagnostic Services. *See* UDS (Unified Diagnostic Services)
Universal Software Radio Peripheral (USRP), 210
`Unknown symbol` messages, 44
unshielded twisted-pair (UTP) cables, 25
update system, attacking IVI system via, 158–165
USB port connection, 9
USB2CAN converter, 244
USRP (Universal Software Radio Peripheral), 210
USRP SDR, 246
UTP (unshielded twisted-pair) cables, 25

V

V2I (vehicle-to-infrastructure) communication, 177
V2V (vehicle-to-vehicle) communication, 177–179
  acronyms, 179
  DRSC protocol, 179–186
  PKI systems, 188–192
  security, 186–187
ValueCAN devices, 252
variable pulse width (VPW) protocol, 22
`vcan` module, 40–41
VDS (Vehicle Descriptor Section), 203
vehicle certificates, 188–189
Vehicle Descriptor Section (VDS), 203
vehicle identification number. *See* VIN
vehicle interface (VI), 85
vehicle make, determining, 202
  interactive probing method, 203–204
  passive CAN bus fingerprinting, 204–207
Vehicle Safety Consortium (VSC3), 186–187
vehicle speed, simulating, 123–126
Vehicle Spy, 252
vehicle-to-infrastructure (V2I) communication, 177
vehicle-to-vehicle communication. *See* V2V (vehicle-to-vehicle) communication
Verdult, Roel, 222, 225
VI (vehicle interface), 85
Victim Board, 137–138
VIN (vehicle identification number)
  decoding, 203–204
  OBD-III standard and, 33
  querying, 203
virtual CAN network, 40–41
VoIP (voice over IP), 31
Volkswagen Group Research, 36
VPW (variable pulse width) protocol, 22
VQ tables, 98
VSC3 (Vehicle Safety Consortium), 186–187
VSCOM adapter, 244

W

WAVE (wireless access for vehicle environments) standard, 184–186
WAVE management entity (WME), 185
WAVE service announcement (WSA) packet, 185
WAVE short-message protocol (WSMP), 179, 185

The Car Hacker's Handbook: A Guide for the Penetration Tester
© 2016 Craig Smith

weaponizing CAN findings, 193–194
  determining vehicle make, 202–207
  responsible exploitation, 208
  writing exploit in C code, 194–202
Wi-Fi connection, 8. *See also* wireless systems
Willem programmer, 236
WinOLS, 103–105
wireless access for vehicle environments (WAVE) standard, 184–186
wireless inputs (IVI systems), 158
wireless systems, 209
  immobilizer systems, 220–230
  key fobs, 215–220
  SDR and, 210–211
  TPMS and, 211–215
Wireshark, 69, 179, 246
wiring (IVI system), 166–170
WME (WAVE management entity), 185
WMI (World Manufacturer Identifier) code, 203
wpa_supplicant threats, 10
WSA (WAVE service announcement) packet, 185
WSMP (WAVE short-message protocol), 179, 185

Z

.zip files, 160

The Car Hacker's Handbook: A Guide for the Penetration Tester
© 2016 Craig Smith