INDEX

Numbers

2FA (two-factor authentication), 207

A

A_AUTH messages, 281
abort function, 359
AbstractAccountAuthenticator class, 194, 202, 203
accept() method, 152
ACCESS_ALL_EXTERNAL_STORAGE permission, 26
access control lists (ACLs), 7
ACCESS_KEYGUARD_SECURE_STORAGE permission, 274
ACCESS_SURFACE_FLINGER permission, 27
access vectors (AVs), 326
  rules, 329–330
    allow rule, 329
    auditallow rule, 330
    dontaudit rule, 330
    neverallow rule, 330
AccountAuthenticatorCache class, 194, 195
AccountAuthenticator.xml file, 195
account management
  AccountManager class, 193
  AccountManagerService
    listing and authenticating accounts, 196
    managing accounts, 196–197
    overview, 193
    requesting authentication token access, 198
    using account credentials, 197
  accounts database
    password security, 201
    table access, 200–201
    table schema, 199–200
  authenticator modules
    adding, 203–205
    cache, 194–195
    overview, 194
  Google accounts
    ClientLogin protocol, 209–210
    Google Account Manager, 206–209
    Google Login Service, 206–209
    Google Play Services, 211–213
    OAuth 2.0, 210–212
    overview, 206
  multi-user support
    overview, 201–202
    per-user account databases, 202
    shared accounts, 202–203
  overview, 192–193
AccountManager class, 94, 112, 192, 193
ACCOUNT_MANAGER permission, 194
AccountManagerService
  listing and authenticating accounts, 196
  managing accounts, 196–197
  overview, 193
  requesting authentication token access, 198
  using account credentials, 197
accounts database
  password security, 201
  table access, 200–201
  table schema, 199–200
accounts.db file, 99
ACLs (access control lists), 7
ACTION_ADD_DEVICE_ADMIN intent action, 219
ACTION_DEVICE_ADMIN_ENABLED intent action, 224
ACTION_GET_RESTRICTION_ENTRIES intent action, 94
ACTION_NDEF_DISCOVERED intent action, 291

ACTION_PACKAGE_ADDED broadcast, 72
ACTION_PASSWORD_FAILED broadcast, 221
ACTION_PASSWORD_SUCCEEDED broadcast, 221
ACTION_SET_NEW_PASSWORD action, 221
ACTION_START_ENCRYPTION intent action, 223
ACTION_TAG_DISCOVERED intent action, 291
ACTION_TECH_DISCOVERED intent action, 291
<active-password> tag, 221
activities
  app architecture, 11
  permissions, 44–45
    enforcement, 36
ActivityManagerService, 49
ADB (Android Debug Bridge), 17, 64, 277–283
  authentication keys, 282
  daemon overview, 277–279
  implementation, 281–282
  need for secure, 279–280
  root access
    on engineering builds, 365–367
    on production builds, 368–369
  securing, 280
  verifying host key fingerprint, 282–283
ADB_AUTH_RSAPUBLICKEY argument, 281
ADB_AUTH_SIGNATURE argument, 281
ADB_AUTH_TOKEN argument, 281
adbd (ADB) daemon, 277–278. *See also* ADB
adb install command, 61, 76–78
adb push command, 76
adb reboot bootloader command, 350
adb reboot recovery command, 354
adb restore command, 284
adb shell command, 110
adb sideload OTA-package-file command, 357
ADB_TRACE environment variable, 279
ADB_VENDOR_KEYS environment variable, 282
addAccountExplicitly() method, 196, 203, 204
addAccountFromCredentials() method, 202
addAccount() method, 196, 203
addNetwork() method, 248
addPermission() method, 43
addProvider() method, 118, 143
ADT (Android Development Tools) Eclipse plugin, 278
Advanced Security SD (ASSD) cards, 298
AEAD (Authenticated Encryption with Associated Data), 125
AES algorithm, 120, 125, 132, 138, 139, 141, 175, 286–287
AIA (Authority Information Access), 162
AID (Application Identifier), 304–305, 312–314
<aid-group> tag, 312, 315
AIDL (Android Interface Definition Language), 6
airplane mode, 92
--algo parameter, 77
algorithm parameter, 120
aliases() method, 135
allowBackup attribute, 284, 287
allow rule, 329
always-on VPN configuration, 235–236
Android Beam feature, 295
Android Debug Bridge. *See* ADB
Android Development Tools (ADT) Eclipse plugin, 278
Android Interface Definition Language (AIDL), 6
AndroidKeyStoreProvider, 188–189
Android Mainlining Project, 2
AndroidManifest.xml file
  overview, 11
  permission requests in, 14, 15, 23
  protected broadcasts in, 38
  verifying packages and, 67
Android Master Key, 67
Android Open Source Project (AOSP), 39, 344
AndroidOpenSSL provider, 140–142, 156
Android Runtime (ART), 69
ANDROID_SDK_HOME environment variable, 282
Android Secure External Caches (ASEC) containers, 76, 81

ANNSI intermediate CA incident, 167
AOSP (Android Open Source Project), 39, 344
APDU_RECIEVED broadcast, 302
APDUs (Application Protocol Data Units), 302–305, 311–315
  specifying routing
    for SE applets, 313–315
    for HCE services, 312–314
APKs
  code signing and, 16
  Dalvik VM and, 3
appdomain domain, 345–346
app_domain() macro, 345
app ID, 12
application-based VPNs, 236–239
  declaring, 237–238
  establishing connection, 238
  notifying user about connection, 238–239
  preparing, 238
Application Identifier (AID), 304–305, 312–314
Application Protocol Data Units. *See* APDUs
application/vnd.android.package-archive MIME type, 52
apply_patch_check function, 359, 361
applypatch command, 362
apply_patch function, 359
apps
  activities, 11
  broadcast receivers, 12
  components of, 11
  content providers, 12
  multi-user support, 92
  sandboxing, 12–14
  services, 11–12
  system, 10
  user-installed, 11, 63
  user management
    application sharing, 101–104
    data directories, 100–101
    overview, 99
ARC4 algorithm, 138, 139, 141, 229
architecture
  apps
    activities, 11
    broadcast receivers, 12
    content providers, 12
    components, 11
    services, 11–12
    system, 10
    user-installed, 11
  Binder
    accessing objects, 8–9
    capability-based security, 7
    death notification, 9
    implementation, 5–6
    object identity, 7
    overview, 5
    reference counting, 9
    security, 6–7
    tokens, 7–8
  Dalvik VM, 3–4
  framework libraries, 10
  inter-process communication, 4–5
  Java runtime libraries, 4
  JCA
    dynamic provider registration, 118–119
    overview, 116–117
    provider implementation, 117–118
    static provider registration, 118
  Linux kernel, 2
  native userspace layer, 2–3
  system services, 4
ART (Android Runtime), 69
ASEC (Android Secure External Caches) containers, 76, 81
asec mount command, 81
asec path command, 81
asec unmount command, 81
ASSD (Advanced Security SD) cards, 298
assert function, 359
assigning permissions, 26–28
asymmetric encryption, 122, 123
AT commands, 297
attribute statement, 325
auditallow rule, 330, 340
AUTHENTICATE_ACCOUNTS permission, 196, 204
Authenticated Encryption with Associated Data (AEAD), 125

authenticator modules
account management and, 192
adding, 203–205
cache, 194–195
overview, 194
Authority Information Access (AIA), 162
authtokens table, 199, 201
AVs. *See* access vectors

B

backdoors, 83
backup framework, 283–288
cloud backup, 283
controlling scope, 287–288
encryption, 286–287
file format, 284–286
local backup, 283–284
BackupManagerService, 284, 285–286
BCB (bootloader control block), 356
BIND_DEVICE_ADMIN permission, 45, 224
Binder
accessing objects, 8–9
capability-based security, 7
death notification, 9
implementation, 5–6
keystore service and, 174
object identity, 7
overview, 5
permissions and, 34
reference counting, 9
security, 6–7
tokens, 7–8
BIND_NFC_SERVICE permission, 311, 317
--bind parameter, 106
BKS (Bouncy Castle KeyStore), 134, 139
blacklisting certificates
Android, 164–166
handling CA key compromises, 163
handling EE key compromises, 163–164
block cipher modes, 124–125
BLOWFISH algorithm, 138, 139
Bluetooth, 92
BOOT_COMPLETED broadcast, 37
bootloader control block (BCB), 356
bootloader program, 252–253, 350–354
fastboot mode, 352–354
commands, 353–354
partition layout, 352–353
protocol, 353
unlocking, 18, 350–352
Bouncy Castle KeyStore (BKS), 134, 139
Bouncy Castle provider, 137–140
broadcasts
permissions, 45–46
permissions enforcement, 37
receivers, 12
user management and, 95–96
BROADCAST_STICKY permission, 37
brute-force attacks, 276–277

C

CA (certificate authority)
defined, 147
handling compromises, 163
installing certificates, 183–184
private, 148
camera, disabling, 223
CAMERA permission, 25
CameraService, 223
capability-based security, 7
CAP_CHOWN capability, 65
CAP_DAC_OVERRIDE capability, 65
CAP_KILL capability, 329
CAP_NET_ADMIN capability, 31
CAP_NET_RAW capability, 31
CAP_SETGID capability, 365–366
CAP_SETUID capability, 365–366
CAP_SYS_ADMIN capability, 330
CAP_SYS_PTRACE capability, 329
card emulation (CE) mode, 290, 295
category attribute, 312
CBC (Cipher-block chaining), 124, 259
CDD (Compatibility Definition Document), 105
CDP (CRL Distribution Point), 162
CE (card emulation) mode, 290, 295
certificate authority. *See* CA
CertificateException, 162, 170
CertificateFactory class, 135–136, 138
certificate revocation list (CRL), 150

certificates
Android, 164–166
blacklisting
handling CA key compromises, 163
handling EE key compromises, 163–164
certificate pinning, 168–170
deleting, 185
direct trust and private CAs, 148
EAP credentials, 172–173
installing CA, 183–184
management of
Android 4.x system trust store, 157–158
system trust store APIs, 161–162
system trust stores, 156–157
using system trust store, 158–161
PKI, 148–150
public key, 146–147
revocation of, 150–151
CertPathBuilder class, 136, 138
CertPath class, 135–136, 153
CertPathTrustManagerParameters class, 153
CertPathValidator class, 136, 138
CERT.SF file, 58
CertStore class, 138
CFB (Cipher feedback), 124
Challenge-Handshake Authentication Protocol (CHAP), 229
changeEncryptionPassword() method, 275
CHANGE_WIFI_STATE permission, 248
Channel class, 308–309
CHAP (Challenge-Handshake Authentication Protocol), 229
chcon utility, 333
checkPassword() method, 275
checkPattern() method, 272
checkPermission() method, 35, 42
checkServerTrusted() method, 162, 169
choosePrivateKeyAlias() method, 182, 186
Cipher-block chaining (CBC), 124, 259
Cipher class
block cipher modes, 124–125
obtaining instance of, 125–126
overview, 123
supported algorithms, 138, 141
using, 126–127
Cipher feedback (CFB), 124
cipher mode, 259–260
cipher suites, 146
ciphertext, 123, 124
classes.dex file, 52
class keyword, 326
class statement, 326–327
clearPassword() method, 197, 200
client credentials, passing and querying, 32
ClientLogin protocol, 209–210
clone() method, 106
CLONE_NEWNS flag, 106, 108, 110
CN (common name), 146
codePath attribute, 71
code signing, 16
COM (Common Object Model), 5
common keyword, 326
common name (CN), 146
Common Object Model (COM), 5
Common Object Request Broker Architecture (CORBA), 5
Comodo attack, 167
compareSignatures() method, 74
Compatibility Definition Document (CDD), 105
CONFIG_ANDROID_PARANOID_NETWORK permission, 31
CONFIG_DM_VERITY kernel configuration item, 255
config_multiuserMaximumUsers system resource, 88
confirmCredentials() method, 197
Conscrypt provider, 141
ContainerEncryptionParams class, 78, 79
content providers
app architecture, 12
defined, 6
permissions
dynamic, 47–49
enforcement, 36
static, 46–47
CONTROL_KEYGUARD permission, 269
-c option, 373

copyResource() method, 79
CORBA (Common Object Request Broker Architecture), 5
Counter (CTR) mode, 124
created attribute, 98
createInstallIntent() method, 181
createSecureContainer() method, 82
credentials
  access control to keystore, 186–187
  Android implementation
    access restrictions, 176
    framework integration, 180
    key blobs, 176
    keymaster module, 176–177
    keystore service, 174–176
    Nexus 4 hardware-backed implementation, 178–180
  AndroidKeyStoreProvider, 188–189
  EAP credentials
    authentication keys and certificates, 172–173
    overview, 172
    system credential store, 173–174
  KeyChain API
    deleting keys and user certificates, 185
    installing CA certificate, 183–184
    KeyChain class, 181–182
    overview, 181, 185–186
    supported algorithms, 185
    using private key, 182
  overview, 187
  passing and querying, 32
CRL (certificate revocation list), 150
CRL Distribution Point (CDP), 162
CRYPT_ENCRYPTION_IN_PROGRESS flag, 265
cryptfs checkpw command, 267
cryptfs enablecrypto inplace command, 265
cryptfs module, 262–263
CryptKeeper class, 266
crypto footer, 260, 265
Cryptographically Secure Pseudo Random Number Generator (CSPRNG), 120, 121
cryptographic service provider (CSP), 115
cryptography
  Android providers
    AndroidOpenSSL, 140–142
    Bouncy Castle, 137–140
    Crypto, 137
    OpenSSL and, 142
    overview, 137
    Spongy Castle, 143–144
  custom providers, 142–143
  hashes, 18
  JCA architecture
    dynamic provider registration, 118–119
    overview, 116–117
    provider implementation, 117–118
    static provider registration, 118
  JCA engine classes
    algorithm names, 120
    CertificateFactory class, 135–136
    CertPathBuilder class, 136
    CertPath class, 135–136
    CertPathValidator class, 136
    Cipher class, 123–127
    KeyAgreement class, 132–133
    KeyFactory class, 129–130
    KeyGenerator class, 131–132
    Key interface, 128
    KeyPairGenerator class, 131
    KeyPair interface, 129
    KeySpec interface, 129
    KeyStore class, 133–135
    Mac class, 127
    MessageDigest class, 120–121
    obtaining engine class instance, 119
    overview, 119
    PBEKey interface, 128–129
    PrivateKey interface, 129
    PublicKey interface, 129
    SecretKeyFactory class, 130–131
    SecretKey interface, 128–129
    SecureRandom class, 120–121
    Signature class, 122–123
Crypto provider, 137
CSP (cryptographic service provider), 115

CSPRNG (Cryptographically Secure Pseudo Random Number Generator), 120, 121
ctl_default_prop type, 336
CTR (Counter) mode, 124
CyanogenMod Android distribution, 310, 374

D

DAC (discretionary access control), 17, 319–320, 364
daemons
  native daemon-level enforcement, 31–33
  security model and, 12
daemonsu binary, 372
Dalvik Executable (DEX), 3, 63
Dalvik VM, 3–4
dangerous protection level, 25
data_file_type attribute, 325
death notification, 9
debuggable flag, 14
debugging, USB, 277–283
  authentication keys, 282
  daemon overview, 277–279
  implementation, 281–282
  need for secure, 279–280
  securing, 280
  verifying host key fingerprint, 282–283
DECRYPT_MODE, 126, 127
decryptStorage() method, 267
delayed provider selection, 116
delete_all() method, 177
deleteEntry() method, 135
delete function, 359
delete_keypair() method, 177
delete_recursive function, 359
derivation mode, 112
DES algorithm, 138, 139, 140
description attribute, 312
development permissions, 39–40
DEVICE_ADMIN_ENABLED broadcast, 46
DeviceAdminInfo class, 216
Device Administration API, 216–228
  account integration, 226–228
    Google Apps, 227–228
    Microsoft Exchange ActiveSync, 226–227
  device administrator, 223–227
    implementing, 224
    managed devices, 226
    setting device owner, 224–225
  policy enforcement, 221–223
  policy persistence, 220–221
  privilege management, 218–219
device administrators, 216, 223–227
  implementing, 224
  managed devices, 226
  setting device owner, 224–225
DeviceAdminReceiver class, 224
Device-mapper framework, 254
device_policies.xml file, 99, 220, 221
DevicePolicyManager class, 217, 220, 226, 274
DevicePolicyManagerService, 217–219
device security, 251–288
  backup framework, 283–288
    cloud backup, 283
    controlling scope, 287–288
    encryption, 286–287
    file format, 284–286
    local backup, 283–284
  disk encryption, 258–268
    booting encrypted devices, 265–267
    changing password, 262–263
    cipher mode, 259–260
    enabling, 263–265
    key derivation, 260–261
    password, 261–262
  OS boot-up and installation control, 252–254
    bootloader program, 252–253
    recovery OS, 253–254
  screen security, 268–277
    brute-force attack protection, 276–277
    keyguard unlock methods, 269–277
    lockscreen implementation, 268–269
  secure USB debugging, 277–283
    authentication keys, 282
    daemon overview, 277–279
    implementation, 281–282
    need for, 279–280

device security, secure USB debugging *(continued)*
securing, 280
verifying host key fingerprint, 282–283
verified boot feature, 254–258
enabling, 256–258
implementation, 255–256
overview, 254–255
device storage encryption, 223
`dex2oat` command, 69
DEX (Dalvik Executable), 3, 63
`dexopt` command, 65
DH (Diffie-Hellman), 132, 139
`digest()` method, 122
DigiNotar attack, 167
Digital Signature Algorithm (DSA), 137, 139, 141, 177
`<disable-camera>` tag, 217
`<disable-keyguard-features>` tag, 217
`disableReaderMode()` method, 294
`DISALLOW_CONFIG_BLUETOOTH` restriction, 92
`DISALLOW_CONFIG_CREDENTIALS` restriction, 92
`DISALLOW_CONFIG_WIFI` restriction, 92
`DISALLOW_INSTALL_APPS` restriction, 92, 93
`DISALLOW_INSTALL_UNKNOWN_SOURCES` restriction, 92
`DISALLOW_MODIFY_ACCOUNTS` restriction, 92, 196
`DISALLOW_REMOVE_USER` restriction, 92
`DISALLOW_SHARE_LOCATION` restriction, 92, 93
`DISALLOW_UNINSTALL_APPS` restriction, 92, 93
`DISALLOW_USB_FILE_TRANSFER` restriction, 92
discretionary access control (DAC), 17, 319–320, 364
disk encryption, 258–268
booting encrypted devices, 265–267
decrypting and mounting */data*, 267
obtaining password, 267
starting all system services, 267
cipher mode, 259–260
enabling, 263–265
controlling encryption using system properties, 263–264
triggering encryption process, 265
unmounting */data*, 264
updating crypto footer, 265
key derivation, 260–261
limitations of, 267
password for, 261–262
changing, 262–263
distinguished name (DN), 146
dm-crypt device-mapper target, 254, 259, 265
dm-verity device-mapper block integrity checking target, 254–258
enabling, 256–258
implementation, 255–256
overview, 254–255
DN (distinguished name), 146
`doFinal()` method, 125
`domain` attribute, 325
`domain_auto_trans()` macro, 328
`domain_trans()` macro, 328
`dontaudit` rule, 330
`doPhase()` method, 133
`DownloadManager` service, 66
DSA (Digital Signature Algorithm), 137, 139, 141, 177

E

EAP (Extensible Authentication Protocol), 242–250
adding networks with `WifiManager`, 248–250
Android Wi-Fi architecture, 244–245
authentication keys and certificates, 172–173
authentication methods, 243–244
EAP-PWD, 244
EAP-TLS, 244
EAP-TTLS, 244
PEAP, 243
credentials management, 245–248
overview, 172
system credential store, 173–174

EAP-PWD (EAP Using Only a Password), 244, 247
EAP-TLS (EAP-Transport Layer Security), 172, 244, 246, 248, 249
EAP-TTLS (EAP-Tunneled Transport Layer Security), 244, 247
EAS (Exchange ActiveSync) account integration, 226–227
ECB (Electronic Code Book), 124
ECDSA (Elliptic Curve DSA), 60, 177
EC (Elliptic Curve), 131
edify functions, 359
editProperties() method, 197
EE (end entity)
  defined, 149
  handling compromises, 163–164
effective user ID (EUID), 6
Electronic Code Book (ECB), 124
Electronic Frontier Foundation, 167
Elliptic Curve DSA (ECDSA), 60, 177
Elliptic Curve (EC), 131
embedded secure elements (eSEs), 298–302
  broadcasts, 301–302
  granting access to, 299–300
  NfcExecutionEnvironment class, 300–301
emulated external storage, 104
EMULATED_STORAGE_SOURCE environment variable, 107, 110
EMULATED_STORAGE_TARGET environment variable, 108, 110
enableForegroundDispatch() method, 292
enableReaderMode() method, 294
encrypted salt-sector initialization vector (ESSIV), 259, 260
<encrypted-storage> tag, 217
encryption. *See also* disk encryption
  backup, 286–287
  device storage, 223
ENCRYPT_MODE, 126
end entity. *See* EE
enforcement, permissions
  framework-level
    activity permission enforcement, 36
    broadcast permission enforcement, 37
    content provider permission enforcement, 36
    dynamic enforcement, 34–36
    protected broadcasts, 37
    service permission enforcement, 36
    sticky broadcasts, 37
  kernel-level, 30–31
  native daemon-level, 31–33
enforcePermission() method, 35, 42
enterprise security, 215–250
  Device Administration API, 216–228
    account integration, 226–228
    device administrator, 223–227
    policy enforcement, 221–223
    policy persistence, 220–221
    privilege management, 218–219
  EAP framework, 242–250
    adding networks with WifiManager API, 248–250
    Android Wi-Fi architecture, 244–245
    authentication methods, 243–244
    credentials management, 245–248
  VPNs, 227–250
    application-based, 236–239
    L2TP, 229–230
    legacy, 231–236
    multi-user support, 239–242
    PPTP, 229
    SSL-based, 230–231
    Xauth, 230
EntropyMixer service, 121
eSEs. *See* embedded secure elements
ESSIV (encrypted salt-sector initialization vector), 259, 260
establish() method, 238
EUID (effective user ID), 6
EV (Extended Validation) certificates, 148
Exchange ActiveSync (EAS) account integration, 226–227
<expire-password> tag, 217
ext4 filesystem, 80, 324

Extended Validation (EV) certificates, 148
Extensible Authentication Protocol. *See* EAP
Extensible Authentication Protocol-Transport Layer Security (EAP-TLS), 172, 244, 246, 248, 249
Extensible Authentication Protocol-Tunneled Transport Layer Security (EAP-TTLS), 244, 247
Extensible Authentication Protocol Using Only a Password (EAP-PWD), 244, 247
external storage
  Android implementation, 106–111
  Linux mount features, 105–106
  overview, 104–105
  permissions, 111–112
EXTERNAL_STORAGE environment variable, 110
EXTRA_CERTIFICATE key, 183

F

Face Unlock method, 271
factory reset, 18
failedAttempts attribute, 98
fastboot boot command, 353, 363
fastboot command-line utility, 353
fastboot devices command, 353
fastboot flashall command, 353
fastboot flash command, 353, 363
fastboot flash:raw command, 353
fastboot mode, 252–253, 352–354
  commands, 353–354
  partition layout, 352–353
  protocol, 353
fastboot oem lock command, 351
fastboot oem unlock command, 351
fastboot update command, 353
fastboot update *ZIP-filename* command, 353
FAT filesystem, 80
FDE (full-disk encryption), 258–259
File Control Information (FCI), 305
file_getprop function, 359
Filesystem in Userspace (FUSE), 105
file_type attribute, 325
FLAG_ADMIN flag, 98
FLAG_GRANT_PERSISTABLE_URI_PERMISSION flag, 48
FLAG_GRANT_READ_URI_PERMISSION flag, 48
FLAG_GRANT_WRITE_URI_PERMISSION flag, 48
FLAG_GUEST flag, 98
FLAG_INITIALIZED flag, 98
FLAG_PRIMARY flag, 98
FLAG_RESTRICTED flag, 98
flags attribute, 71, 98, 220
FLAG_UPDATED_SYSTEM_APP flag, 76
Flame, 53
<force-lock> tag, 216
fork() system call, 28
format function, 359
framework
  credential storage implementation, 180
  libraries making up, 10
  permissions enforcement at framework-level
    activity permission enforcement, 36
    broadcast permission enforcement, 37
    content provider permission enforcement, 36
    dynamic enforcement, 34–36
    protected broadcasts, 37
    service permission enforcement, 36
    sticky broadcasts, 37
ft attribute, 71
FullBackupAgent class, 284
full-disk encryption (FDE), 258–259
FUSE (Filesystem in Userspace), 105

G

GCM (Galois/Counter Mode), 125
GCM (Google Client Messaging), 166
generateCertificate() method, 136
generateCertPath() method, 136
GENERATE_KEYPAIR command, 178
generate_keypair() method, 177
generateKeyPair() method, 131
generatePublic() method, 130

generateSecret() method, 133
Generic Routing Encapsulation (GRE), 229
getAccountCredentialsForCloning() method, 202
GET_ACCOUNTS permission, 196
getAlgorithm() method, 128
getApplicationRestrictions() method, 94
getAuthToken() method, 197, 200
getCallingPid() method, 36
getCallingUid() method, 36
getCertificateChain() method, 183
getCertificate() method, 186
getDeviceOwner() method, 225
getDeviceOwnerName() method, 225
getEmbeddedExecutionEnvironment() method, 301
getEncoded() method, 128, 130
getEncryptionStatus() method, 223
getenforce utility, 333
getEntry() method, 189
getExternalFilesDir() method, 111
getExternalStorageDirectory() method, 110
getFormat() method, 128
getInstance() method, 119
get_keypair_public() method, 177
getKeySpec() method, 130
get() method, 293
getModulus() method, 129
getPassword() method, 196, 200, 201
getPrivateExponent() method, 129
getPrivateKey() method, 183, 186
getprop function, 359
getReaders() method, 309
getsebool utility, 333
getSelectionModeForCategory() method, 312
getUserData() method, 196
getvar command, 353
GID, associating permissions with, 27
GlobalPlatform Card Specification, 303, 304–305
global proxy settings, 222
GLOBAL_SEARCH permission, 47
GLS (Google Login Service), 206–209
Google Account Manager, 206–209
Google accounts
  ClientLogin protocol, 209–210
  Google Account Manager, 206–209
  Google Login Service, 206–209
  Google Play Services, 211–213
  OAuth 2.0, 210–211
  overview, 206
Google Apps account integration, 227–228
Google Client Messaging (GCM), 166
Google experience devices, 191
Google Login Service (GLS), 206–209
Google Play, 25
Google Services Framework (GSF), 206
Google Wallet, 299–300, 302
GPS (Google Play Services), 211–213
GrantedPermission class, 35
grantPermission() method, 48
grants table, 199
GRE (Generic Routing Encapsulation), 229
GSF (Google Services Framework), 206
guest user, 94–95

H

HAL (Hardware Abstraction Layer), 177, 244–245
hardware security module (HSM), 135
HCE. *See* host-based card emulation
HMAC algorithm, 82
HostApduService, 310, 315, 316
<host-apdu-service> tag, 312
host-based card emulation (HCE; software card emulation), 311–318
  Android 4.4 architecture, 310–311
  APDU routing, 311–315
    specifying for HCE services, 312–314
    specifying for SE applets, 313–315
  application security, 317–318
  writing services, 315–317
hostname verification, 154
HostnameVerifier class, 154

HSM (hardware security module), 135
HttpClient class, 159
HTTPS (Hypertext Transfer Protocol Secure), 151
HttpsURLConnection class, 151, 154, 156, 159

I

IAccountAuthenticator interface, 194, 203
icon attribute, 98
id attribute, 98
id utility, 333
IKE (Internet Key Exchange), 230
IKeyguardService, 269
IMPORT_KEYPAIR command, 178
import_keypair() method, 177
inherits keyword, 326
init_daemon_domain() macro, 328
initialization vector (IV), 124, 175
initSign() method, 123
insertProviderAt() method, 118, 143
INSTALL_ACTION intent, 63
InstallAppProgress activity, 67
installCaCert() method, 226
installd daemon, 342–344
installer attribute, 71
installExistingPackageAsUser() method, 102
INSTALL_FAILED_INVALID_APK error, 79
INSTALL_FAILED_SHARED_USER_INCOMPATIBLE error, 40
INSTALL_FAILED_UID_CHANGED error, 40
INSTALL_FAILED_USER_RESTRICTED error, 93
INSTALL_FAILED_VERIFICATION_FAILURE error, 85
INSTALL_FORWARD_LOCK flag, 82
INSTALL_NON_MARKET_APPS setting, 66
INSTALL_PACKAGES permission, 67
installPackageWithVerification AndEncryption() method, 64, 77
INSTALL_PARSE_FAILED_INCONSISTENT_CERTIFICATES error, 74
INTERACT_ACROSS_USERS_FULL permission, 96
INTERACT_ACROSS_USERS permission, 45, 96
INTERNAL_SYSTEM_WINDOW permission, 268
INTERNAL_TARGET_DESELECTED broadcast, 302
Internet Key Exchange (IKE), 230
INTERNET permission, 23
Internet Protocol Security (IPSec) protocol, 229–230
Internet Security Association and Key Management Protocol (ISAKMP), 230
invalidateAuthToken() method, 197
IPC (inter-process communication)
  architecture and, 4–5
  security model, 15–16
IPSec Extended Authentication (Xauth), 230
IPSec (Internet Protocol Security) protocol, 229–230
isAdminActive() method, 224
ISAKMP (Internet Security Association and Key Management Protocol), 230
isBoundKeyAlgorithm() method, 185
isDefaultServiceForCategory() method, 312
isDeviceOwnerApp() method, 225
isDeviceOwner() method, 225
isKeyAlgorithmSupported() method, 185
Issuer Security Domain (ISD) component (Card Manager), 303
isSystemServer selector, 337
it attribute, 71
ITelephony interface, 275–276
IV (initialization vector), 124, 175
--iv parameter, 77

J

jarsigner command, 57, 58
Java Card runtime environment (JCRE), 302–303
Java Cryptography Architecture (JCA)
  algorithm names, 120
  architecture
    dynamic provider registration, 118–119
    overview, 116–117

provider implementation, 117–118
static provider registration, 118
CertificateFactory class, 135–136
CertPath class, 135–136
CertPathValidator class, 136
Cipher class
block cipher modes, 124–125
obtaining instance of, 125–126
overview, 123
using, 126–127
engine classes, 119
obtaining instance of, 119
KeyAgreement class, 132–133
KeyFactory class, 129–130
KeyGenerator class, 131–132
Key interface, 128
KeyPairGenerator class, 131
KeyPair interface, 129
KeySpec interface, 129
KeyStore class, 133–135
Mac class, 127
MessageDigest class, 121–122
PBEKey interface, 128–129
PrivateKey interface, 129
PublicKey interface, 129
SecretKeyFactory class, 130–131
SecretKey interface, 128–129
SecureRandom class, 120–121
Signature class, 122–123
Java runtime libraries, 4
Java Secure Socket Extension (JSSE)
Android implementation, 155–156
certificate blacklisting
Android, 164–166
handling CA key compromises, 163
handling EE key compromises, 163–164
certificate management and validation
Android 4.x system trust store, 157–158
system trust store APIs, 161–162
system trust stores, 156–157
using system trust store, 158–161
hostname verification, 154
overview, 151–152
peer authentication, 152–154
providers for, 137
secure sockets, 152
Java Virtual Machine (JVM), 3
JCA. *See* Java Cryptography Architecture
JCRE (Java Card runtime environment), 302–303
JSSE. *See* Java Secure Socket Extension
--just_exit option, 356
JVM (Java Virtual Machine), 3

K

KDF (key-derivation function), 133
KEK (key-encryption key), 179, 258, 260, 261
kernel-level permissions enforcement, 30–31
KeyAgreement class, 132–133, 139, 141
key blobs, 175, 176
KeyChain API, 226
deleting keys and user certificates, 185
installing CA certificate, 183–184
KeyChain class, 181–182
overview, 181, 185–186
supported algorithms, 185
using private key, 182
KeyChainBroadcastReceiver, 185, 187
KeyChain class, 181–182
KeyChainService class, 185
key derivation, 260–261
key-derivation function (KDF), 133
key-encryption key (KEK), 179, 258, 260, 261
KeyFactory class, 129–130, 137, 139, 141
KeyGenerator class, 131–132, 139
keyguard customizations, disabling, 223
KeyguardHostView class, 269
KeyguardPINView class, 269
KeyguardService, 269
keyguard unlock methods, 269–277
Face Unlock, 271
Password unlock, 270, 273–275
Pattern unlock, 270, 272–273
PIN unlock, 270–271, 273–276
PUK unlock, 271, 275–276
Slide unlock, 270

Key interface, 128
KeyManager class, 153
KeyManagerFactory class, 152
keymaster module, 176–177
KeyPairGenerator class, 131, 139, 141, 188, 189
KeyPairGeneratorSpec class, 189
KeyPair interface, 129
--key parameter, 77
KeySpec interface, 129
KeyStore class, 133–135, 139, 152, 158, 181, 188
-keystore option, 58
keystore service, 174–176
key stretching, 129

L

L2TP (Layer 2 Tunneling Protocol), 229–230
labels. *See* security contexts
lastAttemptMs attribute, 98
lastLoggedIn attribute, 98
least recently used (LRU), 96
legacy VPNs, 231–236
  accessing credentials, 234
  always-on, 235–236
  implementation, 231–233
  profile and credential storage, 233–234
<limit-password> tag, 216
link to death, 9
Linux kernel, 2. *See also* SELinux
  advanced routing, 239–240
  Device-mapper framework, 254
  Logical Volume Manager, 254
Linux Security Modules (LSM) framework, 320
load_policy utility, 333
--locale option, 356
location, multi-user support, 92
lock down functionality, 22, 222
LOCKDOWN_VPN file, 235
LockdownVpnTracker class, 235
lockNow() method, 222
LockPatternUtils class, 269, 275
lockscreen, multi-user support, 90. *See also* keyguard unlock methods
LockScreenUtils class, 272
LockSettingsService, 274
Logical Volume Manager (LVM), 254
login attempt notifications, 221
-l option, 111
low memory killer, 2
LRU (least recently used), 96
ls command, 323, 333
LSM (Linux Security Modules) framework, 320
LVM (Logical Volume Manager), 254

M

--macalgo parameter, 78
Mac class, 127, 139, 141
--mackey parameter, 78
MAC (mandatory access control), 1, 17, 319–320, 321–322, 331
MAC (Message Authentication Code), 127, 176
MANAGE_ACCOUNTS permission, 196, 197
MANAGE_CA_CERTIFICATES permission, 161, 226
MANAGE_DEVICE_ADMINS permission, 219
MANAGE_USERS permission, 95
mandatory access control, 1, 17, 319–320, 321–322, 331
MANIFEST.MF file, 53, 58
marking packets, 240, 242
MASTER_CLEAR_NOTIFICATION broadcast, 302
master key, 175
MediaContainerService, 68, 79, 82
@MEDIA macro, 339
Message Authentication Code (MAC), 127, 176
MessageDigest class, 121–122, 137, 139, 141
META-INF directory, 52
microSD-based secure elements, 298
microSD cards, 80
Microsoft Exchange ActiveSync (EAS) account integration, 226–227
Microsoft Point-to-Point Encryption (MPPE) protocol, 229
middleware MAC (MMAC), 338–339
MITM attack, 166
mkuserdata command, 69, 100
MLS (multi-level security), 321–323

mMacAlgorithm field, 78
mMacKey field, 78
MMAC (middleware MAC), 338–339
mMacTag field, 78
Modecfg (mode-configuration), 230
MODIFY_AUDIO_SETTINGS permission, 27
mountEmulatedStorage() function, 110
mount function, 359
mount() method, 106, 108
mountSecureContainer() method, 82
MountService, 265, 267, 275
MPPE (Microsoft Point-to-Point Encryption) protocol, 229
MS_BIND flag, 106
MSG_COMMAND_APDU broadcast, 310
MS_SHARED flag, 106
MS_SLAVE flag, 108
mtpd daemon, 231–232
multi-level security (MLS), 321–323
multi-user support
  account management
    overview, 201–202
    per-user account databases, 202
    shared accounts, 202–203
  app management
    application sharing, 101–104
    data directories, 100–101
    overview, 99
  broadcasts and, 95–96
  command-line tools, 95
  external storage
    Android implementation, 106–111
    Linux mount features, 105–106
    overview, 104–105
    permissions, 111–112
  features of, 112
  metadata
    user list file, 96–97
    user metadata files, 97–98
    user system directory, 99
  overview, 87–89
  security model, 16–17
  user types
    guest user, 94–95
    primary user, 90–91
    restricted profiles, 92–93
    secondary users, 91–92
  VPNs, 239–242
    implementation, 240–241
    Linux advanced routing, 239–240

N

name attribute, 71
nativeLibraryPath attribute, 71
native userspace layer, 2–3
NDEF_DISCOVERED intent, 292
NDEF (NFC Data Exchange Format), 291–294
near-field communication. *See* NFC
NET_ADMIN permission, 26
netd daemon, 233, 235, 342
netfilter kernel framework, 239–240
NetworkManagementService, 240
network security
  certificate pinning, 168–170
  certificate revocation, 150–151
  Convergence and, 167–168
  direct trust and private CAs, 148
  issues with current PKI system, 166–167
  JSSE
    Android 4.x system trust store, 157–158
    Android implementation, 155–156
    certificate blacklisting, 162–165
    hostname verification, 154
    overview, 151–152
    peer authentication, 152–154
    secure sockets, 152
    system trust store APIs, 161–162
    system trust stores overview, 156–157
    using system trust store, 158–161
  PKI, 148–150
  public key certificates, 146–147
neverallow rule, 330
nextBytes() method, 121
nextSerialNumber attribute, 97
Nexus devices, 104
  credential storage, 178–180
  stock recovery, 354–355

NfcActivity class, 292–293
NfcAdapter class, 292, 294, 295
NfcAdapterExtras class, 301
NFC Data Exchange Format (NDEF), 291–294
NfceeAccessControl class, 299
NFCEE_ADMIN permission, 299
NfcExecutionEnvironment class, 300–301, 303
NFC (near-field communication), 92, 289–318. *See also* secure elements
  Android support for, 290–295
    card emulation mode, 295
    peer-to-peer mode, 294–295
    reader/writer mode, 290–294
  host-based card emulation, 311–318
    Android 4.4 architecture, 310–311
    APDU routing, 311–315
    application security, 317–318
    writing services, 315–317
  overview, 289–290
NfcService, 290–291, 299, 310–311
normal protection level, 24–25
nosetuid flag, 369
NoSuchAlgorithmException, 119
NoSuchProviderException, 119

O

OAuth 2.0, 210–212
OBB (opaque binary blob) files, 65
objects, Binder
  accessing, 8–9
  identity of, 7
OCSP (Online Certificate Status Protocol), 151
OFB (Output feedback), 124
OffHostApduService class, 314
<offhost-apdu-service> element, 314
OFF_HOST_APDU_SERVICE intent, 314
onDeactivated() method, 310, 316
onDisabled() method, 224
onEnabled() method, 224
one-time password (OTP), 207, 296
Online Certificate Status Protocol (OCSP), 151
onPasswordExpiring() method, 223
onTagDiscovered() method, 294
opaque binary blob (OBB) files, 65
OpenID Connect, 209
openLogicalChannel() method, 309
OpenMobile API, 308–309
openSession() method, 309
OpenSSL
  Android keystore engine, 180
  converting to PKCS#8 format, 60
  cryptography providers and, 142
  enc command, 77
openssl enc command, 77
OpenVPN application, 230–231, 238–239
Optimized DEX files, 4, 63
OS boot-up and installation control, 252–254
  bootloader program, 252–253
  recovery OS, 253–254
OTA (over-the-air), 17
  flashing packages, 370–375
  sideloading packages, 357
  signature verification, 357–358
  SIM card updates, 307
  update packages, 253, 258, 355–356, 358–359
OTP (one-time password), 207, 296
Output feedback (OFB), 124
over-the-air. *See* OTA

P

P2P (point-to-point) connections, 172
package_extract_dir function, 359
package_extract_file function, 359
PACKAGE_INSTALLED broadcast, 37
PackageInstallerActivity, 66, 67
package management
  Android Application Package Format, 51–86
  APK install process
    active components, 63–67
    Android 4.1 forward locking implementation, 82
    encrypted apps and Google Play, 82–83
    forward locking, 79–80
    installing encrypted APKs, 76–79
    installing local package, 66–76

location of application packages and data, 62–63
updating package, 72–76
code signing
in Android, 59–61
in Java, 54–59
package verification
Android support for, 84–85
Google Play implementation, 85–86
PackageManagerService, 35, 68, 77, 84, 85, 95, 100, 102, 194, 339
PACKAGE_NEEDS_VERIFICATION action, 84
PACKAGE_REMOVED broadcast, 187
PACKAGE_REPLACED broadcast, 75
package-restrictions.xml file, 99, 101
packages. *See* APKs
packages.xml file, 63
PACKAGE_VERIFICATION_AGENT permission, 84, 85
PACKAGE_VERIFIED broadcast, 85
PACKAGE_VERIFIER_ENABLE setting, 84
padding, 123
PAP (Password Authentication Protocol), 229
partial attribute, 98
password expiration timeout, 223
<password-owner> tag, 221
PASSWORD_QUALITY_ALPHANUMERIC constant, 220, 221
PASSWORD_QUALITY constant, 274
PASSWORD_QUALITY_NUMERIC constant, 274
Password unlock method, 270, 273–275
Pattern unlock method, 270, 272–273
PBEKey interface, 128–129
PBKDF2 algorithm, 260–262, 262, 286
PEAP (Protected Extensible Authentication Protocol), 243, 246
peekAuthToken() method, 196
peer authentication, 152–154
peer-to-peer (P2P) mode, 290, 294–295
pending intents, 49–50
PERMISSION_DENIED response, 34
PERMISSION_GRANTED response, 34
permissions
activity, 44–45
assigning, 26–28
broadcast, 45–46
content provider
dynamic, 47–49
static, 46–47
custom, 42–43
enforcement of
framework-level, 33–37
kernel-level, 30–31
native daemon-level, 31–33
external storage, 111–112
management of, 21–23
overview, 21–22
pending intents, 49–50
PID assignment and, 28–30
private components, 43–44
protection levels
dangerous, 25
defined, 24
normal, 24–25
signature, 26
signatureOrSystem, 26
public components, 43–44
requesting, 22
security model, 14–15
service, 44–45
shared user ID, 40–42
system
development permissions, 39–40
overview, 37–39
signature permissions, 39
viewing list of known, 22
when granted, 104
permissive statement, 326
PFX (Personal Information Exchange Syntax), 134
phishing applications, 83
PID (process ID)
assignment of, 28–30
Binder and, 6
pinHash attribute, 98
PIN unlock key (PUK) unlock method, 271, 275–276
PIN unlock method, 270–271, 273–275, 275–276
PittPatt (Pittsburgh Pattern Recognition) company, 271
PKCS#12 files, 172
PKCS (Public Key Cryptography Standard), 125

PKI (Public Key Infrastructure)
certificate revocation, 150–151
direct trust and private CAs, 148
overview, 148–150
public key certificates, 146–147
PKIX (X.509-based PKI), 138, 152
PKPE (Public Key Pinning Extension for HTTP), 168
platform keys
security model, 16
system apps and, 39
@PLATFORM macro, 339
platform signing key, 10
`pm` command, 61
`pm create-user` command, 95
`pm get-max-users` command, 95
`pm install` command, 76, 78
`pm list users` command, 95
point-to-point (P2P) connections, 172
`<policies>` tag, 220
`powerctl_prop` type, 336
PPP (Point-to-Point Protocol), 229
PPTP (Point-to-Point Tunneling Protocol), 229
`prepare()` method, 238
primary user, 90–91
private CAs, 148
`PrivateKeyEntry`, 133
`PrivateKey` interface, 129
private keys, using, 182
`processCommandApdu()` method, 317
`processCommand()` method, 310
process ID. *See* PID
process isolation, 5
protected broadcasts, 37
Protected Extensible Authentication Protocol (PEAP), 243, 246
protection levels
dangerous, 25
defined, 24
normal, 24–25
signature, 10, 26
signatureOrSystem, 26, 63
`Provider` class, 118
providers
AndroidKeyStoreProvider, 188–189
cryptography
AndroidOpenSSL, 140–142
Bouncy Castle, 137–140
Crypto, 137
custom, 142–143
OpenSSL and, 142
overview, 137
Spongy Castle, 143–144
`ps` command, 323, 333
public components, 43–44
public key certificates, 146–147
Public Key Cryptography Standard (PKCS), 125
Public Key Infrastructure. *See* PKI
`PublicKey` interface, 129
`Public-Key-Pin` header, 168
Public Key Pinning Extension for HTTP (PKPE), 168
`Public-Key-Pins-Report-Only` header, 169
PUK (PIN unlock key) unlock method, 271, 275–276

Q

QSEE (Qualcomm's Secure Execution Environment), 178

R

racoon daemon, 231–232, 234
radio-frequency identification (RFID) technology, 290
Random Number Generator (RNG), 120
RA (registration authority), 149
RC4 algorithm, 138, 139, 141, 229
`READ_CONTACTS` permission, 47
reader/writer (R/W) mode, 290–294
reader mode, 293–294
reading tags, 293
registering for tag dispatch, 291–292
tag technologies, 292–293
`READ_EXTERNAL_STORAGE` permission, 111
`READ_LOGS` permission, 39
read-only partition, 10
`READ_SMS` permission, 25
read-write partition, 11
recovery binary, 356–357
recovery OS, 253–254, 354–364
custom recoveries, 363–364
stock recovery, 354–363
applying updates, 359–360
controlling, 356–357
copying and patching files, 361

finishing updates, 361–362
OTA signature verification, 357–358
setting file ownership, permissions, and security labels, 361
sideloading OTA packages, 357
starting system update process, 358–359
updating recovery, 362–363
RecoverySystem class, 357
reference counting, 9
reference implementation (RI), 138
registration authority (RA), 149
relabelto permission, 343
@RELEASE macro, 339
remote procedure calls (RPC), 5
removeAccount() method, 197, 201
removeActiveAdmin() method, 224
removeProvider() method, 118
requesting permissions, 22
Requests for Comments (RFCs), 125
requireDeviceUnlock attribute, 312
resetPassword() method, 221, 222
<reset-password> tag, 216
resource attribute, 194
resourcePath attribute, 71
Resources class, 52
restorecon command, 333, 335
restrictedAccountType attribute, 94, 203
restricted profiles
access to online accounts, 94
applying restrictions, 93–94
user restrictions, 92
revokePermission() method, 48
revokeUriPermission() method, 48
RFCs (Requests for Comments), 125
RFID (radio-frequency identification) technology, 290
rild (radio interface) daemon, 275
RI (reference implementation), 138
RNG (Random Number Generator), 120
ro.crypto.fs_crypto_blkdev property, 264, 267
ro.crypto.state property, 263
ro.debuggable property, 369
root access, 364–376
engineering builds, 365–368
starting ADB as root, 365–367
using su command, 367–368
production builds, 368–376
changing boot or system image, 369
flashing OTA packages, 370–375
via exploits, 375–376
root user, 65
ro.secure property, 369
RPC (remote procedure calls), 5
RSA algorithm, 55, 120, 139, 141, 255, 257, 357
runcon utility, 333
run_program function, 359
runtime libraries, 4
R/W mode. *See* reader/writer mode

S

S2C (SignalIn/SignalOut connection), 299
salt attribute, 98
same origin policy, 16
sandboxing, app, 12–14
SA (Security Association), 230
saveLockPassword() method, 275
saveLockPattern() method, 272
scanPackageLI() method, 68, 75
SCM_CREDENTIALS control message, 32
screen security, 268–277
brute-force attack protection, 276–277
keyguard unlock methods, 269–277
Face Unlock, 271
Password unlock, 270, 273–275
Pattern unlock, 270, 272–273
PIN unlock, 270–271, 273–275, 275–276
PUK unlock, 271, 275–276
Slide unlock, 270
lockscreen implementation, 268–269
scrypt key derivation function, 261
SD card, 104
seclabel command, 333, 334

secondary users, 91–92
SecretKeyEntry, 133
SecretKeyFactory class, 130–131, 140
SecretKey interface, 128–129
secure elements (SEs), 179, 295–309. *See also* host-based card emulation
  embedded, 298–301
    broadcasts, 301–302
    granting access to, 299–300
    NfcExecutionEnvironment class, 300–301
  execution environment, 302–305
    querying, 304–305
  microSD-based SEs, 298
  UICCs, 297–298, 305–309
    accessing, 307–308
    application implementation and installation, 307
    applications, 306–307
    SIM cards and, 305–306
    using OpenMobile API, 308–309
SecureRandom class, 120–121, 137, 142
Secure Socket Layer. *See* SSL
Security Association (SA), 230
security contexts (labels), 322–323
  assignment and persistence, 324
  labeling
    application processes, 336–338
    files, 334–335
    system properties, 335–336
Security-Enhanced Linux. *See* SELinux
SecurityException, 36
security model
  application sandboxing, 12–14
  code signing, 16
  IPC, 15–16
  multi-user support, 16–17
  overview, 12
  permissions, 14–15
  platform keys, 16
  SELinux, 17
  system updates, 17–18
  verified boot, 18–20
security.properties file, 118
sedispol command, 341
SEEK for Android project, 297, 308
seinfo command, 341
seinfo tag, 30, 68, 338–339
self keyword, 329
SELinux (Security-Enhanced Linux), 319–347
  access vector rules, 329–330
    allow rule, 329
    auditallow rule, 330
    dontaudit rule, 330
    neverallow rule, 330
  Android 4.4 policy, 340–347
    app domains, 345–347
    enforcing domains, 342–344
    overview, 341–342
    unconfined domains, 344–345
  architecture of, 320–321
  defined, 17
  domain transition rules, 328
  implementation, 330–340
    device policy files, 339–340
    kernel changes, 331–332
    policy event logging, 340
    userspace changes, 332–339
  mandatory access control, 319–323
  modes, 322
  security contexts (labels), 322–323
    assignment and persistence, 324
  security model, 17
  security policy, 324–328
    object class and permission statements, 326–327
    type and attribute statements, 325
    user and role statements, 325
  type transition rules, 327–328
sendBroadcast() method, 37, 45
--send_intent option, 356
sendResponseApdu() method, 316
serialNumber attribute, 98
Server Name Indication (SNI), 156
Service Provider Interface (SPI), 117
services
  app architecture, 11–12
  permissions enforcement, 36
SEs. *See* secure elements
sesearch command, 341, 342
SEService class, 308–309

setActiveAdmin() method, 219
setAuthToken() method, 196
setCameraDisabled() method, 223
setcon command, 333
setDefaultSSLSocketFactory() method, 154
setDeviceOwner() method, 225
setenforce command, 333
<set-global-proxy> tag, 216
setGrant() method, 187
set-group-ID (SGID), 12
setKeyguardDisabledFeatures() method, 223
setMaximumFailedPasswordsForWipe() method, 222
setMaximumTimeToLock() method, 222
set_metadata function, 359
set_metadata_recursive function, 359, 361
setNdefPushMessageCallback() method, 295
setNdefPushMessage() method, 295
setPasswordExpirationTimeout() method, 223
setPassword() method, 196, 200
setsebool command, 333
setSeed() method, 121
setSSLSocketFactory() method, 154
setStorageEncryption() method, 223
setUserData() method, 196
set-user-ID (SUID), 12
SGID (set-group-ID), 12
SHA-1 algorithm, 137, 139, 141, 358
SHA1PRNG algorithm, 137, 142
SHA1withDSA algorithm, 137
SHA-256 algorithm, 117, 120, 127, 139, 141, 259, 358
shared_accounts table, 200, 202
@SHARED macro, 339
shared user ID, 40–42
sharedUserId attribute, 71
sharpening, 69
show_progress function, 359
--show_text option, 356
-sigfile option, 57
SignalIn/SignalOut connection (S2C), 299
signapk tool, 58, 60
Signature class, 73, 122–123, 137, 140, 142
signature files, 54
signatureOrSystem protection level, 26, 63
signature permissions, 39
signature protection level, 10, 26
SIGN_DATA command, 178
sign_data() method, 177
SIMalliance Open Mobile API specification, 297
SIM cards. *See also* UICCs
  multi-user support, 91
  UICCs and, 305–306
  unlocking, 18
Simple NDEF Exchange Protocol (SNEP) protocol, 294
SIM Toolkit (STK) applications, 307
Single Wire Protocol (SWP), 298
Slide unlock method, 270
SmartCard API, 297–298
SMARTCARD permission, 309
SmartcardService, 308–309
SNEP (Simple NDEF Exchange Protocol) protocol, 294
SNI (Server Name Indication), 156
SoC (system on a chip), 178
software card emulation. *See* host-based card emulation
SO_PEERCRED socket option, 32
SPI (Service Provider Interface), 117
Spongy Castle provider, 143–144
spyware, 83
SQLite, 99
SSLContext class, 151
SSLEngine class, 151
SSL Observatory project, 167
SSL (Secure Socket Layer)
  certificate revocation, 150–151
  direct trust and private CAs, 148
  PKI, 148–150
  public key certificates, 146–147
  SSL-based VPNs, 230–231
SSLServerSocket class, 152
SSLSocket class, 152
SSLSocketFactory class, 154
--stages option, 356
startActivityForResult() method, 36, 44
startActivity() method, 36, 44
sticky broadcasts, 37
STK (SIM Toolkit) applications, 307

store() method, 135
StrictJarFile class, 67
su command, 367–368, 372–373
SUID (set-user-ID), 12
SuperSU application, 370–372
    initializing, 372–374
superuser, 64
supplyPinReportResult() method, 275
supplyPukReportResult() method, 276
surfaceflinger daemon, 345
SWP (Single Wire Protocol), 298
symlink function, 359
symmetric encryption, 123
system
    apps, 10
    credential store, 173–174
    permissions
        development permissions, 39–40
        overview, 37–39
        signature permissions, 39
    services, 4
system_data_file type, 325
system on a chip (SoC), 178
system partition, 10
system trust stores
    Android 4.x, 157–158
    APIs, 161–162
    overview, 156–157
    using, 158–161
system updates, 17–18, 349–364
    bootloader program, 350–354
        fastboot mode, 352–354
        unlocking, 350–352
    recovery OS, 354–364
        custom recoveries, 363–364
        stock recovery, 354–363

T

TACK (Trust Assertions for Certificate Keys), 168–169
TAG_DISCOVERED intent, 292
--tag parameter, 78
Team Win Recovery Project (TWRP), 363
TECH_DISCOVERED intent, 292
<tech-list> element, 292
tethering, 91
TE (type enforcement), 321–322, 341
Timestamping Authority (TSA), 57
TLS (Transport Layer Security), 145
TOFU (Trust on First Use), 72, 167
tokens, Binder, 7–8
towelroot exploit, 375
TPMs (Trusted Platform Modules), 179
transceive() method, 303
translateKey() method, 130
transmit() method, 308
Transport Layer Security (TLS), 145
trust anchors, 148
Trust Assertions for Certificate Keys (TACK), 168–169
TrustedCertificateEntry class, 133
TrustedCertificateStore class, 157, 187
Trusted Platform Modules (TPMs), 179
TrustManager class, 153
TrustManagerFactory class, 152, 159
Trust on First Use (TOFU), 72, 167
trustStore property, 156
TrustZone, 179
TSA (Timestamping Authority), 57
two-factor authentication (2FA), 207
TWRP (Team Win Recovery Project), 363
TYPE_ANY, 176
type enforcement (TE), 321–322, 341
TYPE_GENERIC, 176
TYPE_KEY_PAIR, 176
TYPE_MASTER_KEY, 176
type statement, 325
type_transition rule, 327–328

U

ueventd daemon, 334
UICCs (Universal Integrated Circuit Cards), 180, 296, 297–298, 305–309
    accessing, 307–308
    application implementation and installation, 307
    applications, 306–307
    SIM cards and, 305–306
    using OpenMobile API, 308–309

UIDs
associating permissions with, 27
Linux UIDs and, 88
multi-user support, 16
sharing, 14
ui_print function, 359
umount function, 359
unconfineddomain domain, 344–345
uninstallCaCert() method, 226
Universal Integrated Circuit Cards. *See* UICCs
Unknown Sources
multi-user support and, 91, 92
PackageInstaller and, 63, 66
unshare() method, 106
UnsupportedOperationException, 203
untrusted_app type, 325, 346
UNWRAP_MODE, 126
updateCredentials() method, 197
update() method, 122
--update_package option, 356
UPDATE_PINS broadcast, 170
updates. *See* system updates
USB
multi-user support, 92
secure debugging, 277–283
authentication keys, 282
daemon overview, 277–279
implementation, 281–282
need for, 279–280
securing, 280
verifying host key fingerprint, 282–283
UsbDebuggingActivity, 281
UsbDeviceManager class, 282
USE_CREDENTIALS permission, 197, 198
userdata partition, 11
decrypting and mounting, 267
unmounting for encryption, 264
userId attribute, 71
user-installed apps, 11
userlists.xml file, 97
user management
app management
application sharing, 101–104
data directories, 100–101
overview, 99
broadcasts and, 95–96
command-line tools, 95
external storage
Android implementation, 106–111
Linux mount features, 105–106
overview, 104–105
permissions, 111–112
metadata
user list file, 96–97
user metadata files, 97–98
user system directory, 99
multi-user support
features of, 112
overview, 87–89
user types
guest user, 94–95
primary user, 90–91
restricted profiles, 92–93
secondary users, 91–92
UserManager API, 88
UserManagerService, 95
USER_STARTED broadcast, 96
USER_STARTING broadcast, 96
user statement, 325
USER_STOPPED broadcast, 96
USER_STOPPING broadcast, 96
USES_ENCRYPTED_STORAGE constant, 217, 223
<uses-policies> tag, 218
USES_POLICY_DISABLE_CAMERA constant, 217, 223
USES_POLICY_DISABLE_KEYGUARD_FEATURES constant, 217, 223
USES_POLICY_EXPIRE_PASSWORD constant, 217, 223
USES_POLICY_FORCE_LOCK constant, 216, 222
USES_POLICY_LIMIT_PASSWORD constant, 216, 221
USES_POLICY_RESET_PASSWORD constant, 216, 218, 222
USES_POLICY_SETS_GLOBAL_PROXY constant, 216, 222
USES_POLICY_WATCH_LOGIN constant, 216, 221, 222
USES_POLICY_WIPE_DATA constant, 216, 222
ut attribute, 71

V

validate() method, 136
VerificationParams class, 78
verified boot feature, 18–20, 254–258
    enabling, 256–258
    implementation, 255–256
    overview, 254–255
VERIFY_DATA command, 179
verify_data() method, 177
verify flag, 255
verify() method, 123, 154
verifyPackage() method, 357
verifyPendingInstall() method, 85
verity metadata block, 257
version attribute, 71, 97
VFS (Virtual Filesystem), 105
virtual private networks. *See* VPNs
vold daemon, 263, 267, 342
vold_prop type, 336
VPNs (virtual private networks), 227–250
    application-based, 236–239
        declaring, 237–238
        establishing connection, 238
        notifying user about connection, 238–239
        preparing, 238
    configuration screen for, 91
    EAP credentials
        authentication keys and certificates, 172–173
        overview, 172
        system credential store, 173–174
    L2TP, 229–230
    legacy, 231–236
        accessing credentials, 234
        always-on, 235–236
        implementation, 231–233
        profile and credential storage, 233–234
    multi-user support, 239–242
        implementation, 240–241
        Linux advanced routing, 239–240
    PPTP, 229
    SSL-based, 230–231
    Xauth, 230
VpnService class, 236–238

W

wakelocks, 2
<watch-login> tag, 216
WebView control, 210
Wi-Fi
    EAP credentials
        authentication keys and certificates, 172–173
        overview, 172
        system credential store, 173–174
    EAP framework, 242–250
        adding networks with WifiManager API, 248–250
        Android Wi-Fi architecture, 244–245
        authentication methods, 243–244
        credentials management, 245–248
    multi-user support and, 91
    user restrictions, 92
WifiConfiguration class, 248
wifi_data_file type, 327
WifiEnterpriseConfig class, 249
WifiManager API, 248–250
WifiManager class, 245
Wi-Fi Protected Access II (WPA2), 242
Wi-Fi Protected Access (WPA), 242
WifiService, 245
WifiStateMachine class, 245
WiMAX, 91
--wipe_cache option, 356, 361
<wipe-data> tag, 216
wipeData() method, 222
--wipe_data option, 356
wiping user data, 222
-w option, 60
WPA2 (Wi-Fi Protected Access II), 242
wpa_socket type, 327
wpa_supplicant daemon, 244–246, 327
WPA (Wi-Fi Protected Access), 242
WRAP_MODE, 126
WRITE_CONTACTS permission, 47
WRITE_EXTERNAL_STORAGE permission, 23, 104, 111
write_raw_image function, 359
WRITE_SECURE_SETTINGS permission, 39, 299

X

X.509-based PKI (PKIX), 138, 152
X.509 certificates, 130, 135, 138, 141, 143, 146, 357
`X509ExtendedKeyManager` interface, 153
`X509KeyManager` interface, 153
`X509_NAME_hash_old()` function, 157
`X509TrustManagerExtensions` class, 169
`X509TrustManager` interface, 153
Xauth (IPSec Extended Authentication), 230
XTS (XEX-based tweaked-codebook mode with ciphertext stealing), 260

Z

ZIP format, 52, 353
`-Z` option, 323, 333
`ZygoteConnection` class, 336
zygote process, 28, 107, 336, 342, 345