INDEX

A

Address Sanitizer (ASan), 55
Address Space Layout Randomization (ASLR), 8, 53–54, 87
advertisingIdentifier, 235
advertisingTrackingEnabled, 235
AES algorithm, 226–227
AFNetworking, 122–124
alloc, 19
.app directory, 78–79
Apple System Log (ASL), 161–164
application anatomy, 27–38
  Bundle directory, 33–34
  Data directory, 34–37
  device directories, 32–33
  Documents directory, 34–35
  Shared directory, 37
applicationDidEnterBackground, 20, 167, 179–180, 183
application extensions, 140–144
  extensionPointIdentifier, 144
  extension points, 140
  NSExtensionActivationRule, 142
  NSExtensionContext, 143
  NSExtensionItem, 143
  shouldAllowExtensionPoint-Identifier, 143
  third-party keyboards, 144
Application Support directory, 35
applicationWillResignActive, 180
applicationWillTerminate, 20, 167, 183
app review, 3–4, 10–12
  evading, 11–12
App Store, 3–4,
  review process, 10–12
    bypassing, 11–12
ARC (Automatic Reference Counting), 19
ASan (Address Sanitizer), 55
ASIHTTPRequest, 122, 124–125
ASL (Apple System Log), 161–164
ASLR (Address Space Layout Randomization), 8, 53–54, 87
authentication
  biometrics, 231–232
  fingerprint authentication, safety of, 232
  HTTP basic authentication, 110–111, 119–121
  Local Authentication API, 231–232
  TouchID, 231–232
    LAContext, 231–232
autocorrection, 175–177
Automatic Reference Counting (ARC), 19
autoreleasepool, 19

B

backtrace (bt) command, 65–66
BEAST attack, 117
biometrics, 231–232
black-box testing, 77
blacklisting, 200
blocks, Objective-C
  declaring, 18
  exposing to JavaScript, 150–151
Bluetooth Low Energy (BTLE), 244
Bluetooth PAN, 125
Bonjour, 125
Boot ROM, 4
breakpoints, 62
  actions, 70–72, 164
  conditions, 72
  enabling/disabling, 64
  setting, 62–64
brute-forcing, PINs, 214
bt (backtrace) command, 65–66
BTLE (Bluetooth Low Energy), 244
buffer overflows, 12, 193–196
  example, 194–195
  preventing, 195–196
Bundle directory, 33–34
bundle ID, 33, 138

iOS Application Security: The Definitive Guide for Hackers and Developers
© 2016 David Thiel

- bundle seed ID, 218–219
- BurpSuite, 43–47

C

- CA (certificate authority), 114–115
- CA certificate, 44
  - certificate management, 47
  - certificate pinning, 114–117, 124
    - defeating, 96–97
- cache management, 170–171
  - removing cached data, 171–174
- *Caches* directory, 35–36
- caching, 36
- `CALayer`, 182–183
- canonical name (CN), 128–129
- `canPerformAction`, 168–169
- categories, Objective-C, 22–23
- `CBPeripheralManager`, 246
- CCCrypt, 186, 226
- `CCHmac`, 229
- CCRespring, 79
- certificate authority (CA), 114–115
- `certificateHandler`, 126
- `CFArrayRef`, 112–113
- `CFBundleURLSchemes`, 133
- `CFDataRef`, 113
- `CFPreferences`, 36, 178
- `CFStream`, 48, 107, 128–129
- `chflags` command, 42
- clang, 51–53
- `class-dump` command, 90, 92–93
- `CLBeaconRegion`, 244, 246
- `CLLocationManager`, 240, 244
- CN (canonical name), 128–129
- Cocoa, 14
- Cocoa Touch, 14
- code segment, 193–195
- `codesign` command, 82
- CommonCrypto, 151, 230, 230
  - CCCrypt, 226
- `CompleteUnlessOpen`, 222–223
- `CompleteUntilFirstUserAuthentication`, 220
- `cookieAcceptPolicy`, 237
- cookies, 36
  - acceptance policy, 237–238
  - theft of, 114
- *Cookies* directory, 36
- copy/paste, disabling, 168–169
- Cordova, 150, 154–157
- *Cordova.plist*, 156
- `CoreBluetooth`, 246
- Core Data, 204, 223
- CRIME attack, 118
- cross-site scripting (XSS), 199–200
  - input sanitization, 200–201
  - output encoding, 201–202
- `cryptid`, 81, 86–90
- `cryptoff`, 86–90
- `cryptsize`, 86–90
- cURL, 78
  - certificates, 93–94
- Cycript, 90, 93–94
- Cydia, 31, 77
- Cydia Substrate, 78, 97–100

D

- DAC (discretionary access control), 4–5
- *Data* directory, 34–37
- data leakage, 161–188
  - Apple System Log, 161–164
  - autocorrection, 175–177
  - breakpoint actions, 164
  - cache management, 170–174
  - *dynamic-text.dat*, 177
  - ephemeral sessions, 173
  - HTTP caches, 169–174
    - local storage, 174
  - iCloud, 161
    - avoidance of, 188
  - keylogger, 177
  - `NSLog`, 161–164
    - disabling, 163
  - `NSURLSession`, 173
  - pasteboards, 164–169
    - `canPerformAction`, 168
    - disabling copy/paste, 168–169
    - *pasteboardDB* file, 165–167
    - `pasteboardWithUniqueName`, 165–167
    - protecting data, 167–169
    - `UISearchBar`, 165
    - wiping, 167
  - snapshots, 178–184
    - `applicationDidEnterBackground`, 179–180
    - preventing suspension, 183–184
    - screen obfuscation, 179–183

iOS Application Security: The Definitive Guide for Hackers and Developers
© 2016 David Thiel

state preservation, 184–187
restorationIdentifier, 184–185
user preferences, 178
Data Protection API, 7–8, 219–225
Class Key, 220
CompleteUnlessOpen, 222–223
CompleteUntilFirstUserAuthentication, 220
DataProtectionClass, 223
Data Protection entitlement, 223–224
delegate methods, 224
detecting, 225
FileProtectionComplete, 220–221
isProtectedDataAvailable, 225
protection levels, 220–223
DataProtectionClass entitlement, 157
data segment, 193–195
dataTaskWithRequest, 18
data theft, 161
dd command, 88
debugging, 61–75
breakpoints, 62
actions, 70–72
conditions, 72
enabling/disabling, 64
setting, 62–64
Debug Navigator, 65
debugserver, 81–84
connecting to, 83
installing, 81–82
fault injection, 72–73
frames and variables, 64–68
lldb, 62–75
backtrace (bt) command, 65–66
expr command, 69
frame select command, 66–67
frame variable command, 66
image list, 87
print object command, 67–68
object inspection, 68
tracing data, 74
variables and properties, 69–70
Debug Navigator, 65
debugserver, 81–84
connecting to, 83
installing, 81–82
decodeRestorableStateWithCoder, 184
decrypting binaries, 80–90
.default_created.plist, 32
Default-Portrait.png, 179
defaults command, 42
delegation, 20
DES algorithm, 226
deserialization, 21
developer team ID, 138
device directories, 32–33
Device Key, 7
device.plist, 32
didFinishNavigation, 159–160
did messages, 20
didReceiveCertificate, 126
disassembly, with Hopper, 94–96
discretionary access control (DAC), 4–5
Documents directory, 34–35
Do Not Track, 236–237
dpkg command, 96, 99–101
DTrace, 55, 61
dumpdecrypted command, 80
_dyld_get_image_name, 10
dylibs, 10
dynamic analysis, 55
dynamic patching, 11–12

E

emulator, *see* Simulator
encodewithcoder, 21–22
encrypted segment, 84–90
encryption, 211–230
AES, CCB mode, 226–227
bad algorithms, 226
CommonCrypto, 225, 230
CCCrypt, 226
Curve25519, 222
Data Protection API, 5, 7–8, 219–225
Class Key, 220
CompleteUnlessOpen, 222–223
CompleteUntilFirstUserAuthentication, 220
DataProtectionClass, 223
Data Protection entitlement, 223–224
delegate methods, 224
detecting, 225
FileProtectionComplete, 220–221

- encryption, Data Protection API, *continued*
    - `FileProtectionCompleteUnlessOpen`, 222
    - `isProtectedDataAvailable`, 225
    - protection levels, 220–223
  - DES algorithm, 226
  - Device Key, 7
  - disk encryption, 5–7
  - Elliptic Curve Diffie-Hellman algorithm, 222
  - entropy, 227
  - File Key, 7
  - full disk encryption, 5–7
  - hashing, 228–230
  - HMAC (Hash Message Authentication Code), 229–230
  - initialization vector (IV), 226–227
  - Keychain, 6–7, 113, 186, 211–219
    - API, 7
    - backups, 212
    - iCloud synchronization, 219
    - item classes, 214
    - key hierarchy, 6–7
    - `kSecAttrAccessGroup`, 218–219
    - protection attributes, 212–214
    - `SecItemAdd`, 219
    - shared Keychains, 218–219
    - usage, 214–217
    - wrappers, 217–218
  - key derivation, 227–228
  - key quality, 227–228
  - Lockbox, 217
  - OpenSSL, 228–229
  - RNCryptor, 230
  - `SecRandomCopyBytes`, 227
  - TLS (Transport Layer Security), 127–129
- entitlements, 218, 223
- *entitlements.plist*, 81–82
- entropy, 227
- Erica Utilities, 31, 78
- */etc/hosts*, 49
- `EXC_BAD_ACCESS`, 191
- eXecute Never (XN), 8–9
- `expr` command, 69
- `extensionPointIdentifier`, 144
- `extractIdentityAndTrust`, 112–113

F

- fault injection, 72–73
- File Juicer, 169, 174
- File Key, 7
- `FileProtectionComplete`, 220–221
- filesystem monitoring, 58–59
- Finder, 42
- fingerprint authentication, safety of, 232
- forensic attackers, 161
- format string attacks, 190–193
  - `NSString`, 192–193
  - preventing, 191–193
- Foundation classes, 14
- frames and variables, 68
- `frame select` command, 66–67
- `frame variable` command, 66
- Full Disk Encryption, 5–7
- fuzzing, 55

G

- garbage collection, 18
- gdb, 62
- geolocation, 238
  - accuracy, 239
  - `CLLocationManager`, 240
  - risks, 238–239
- `get-task-allow`, 82
- Google Toolbox for Mac, 202
- GPS, 238

H

- `handleOpenURL`, 136
- hashing, 228–230
- Hash Message Authentication Code (HMAC), 229
- `hasOnlySecureContent`, 159–160
- HealthKit, 240–241
- heap, 8, 53–54, 193
- hidden files, 41–42
- HMAC (Hash Message Authentication Code), 229
- Homebrew, 46, 88, 94, 99
- hooking
  - with Cydia Substrate, 97–100
  - with Introspy, 100–103
- Hopper, 94–96

iOS Application Security: The Definitive Guide for Hackers and Developers
© 2016 David Thiel

HTML entities, 201
- encoding, *see* output encoding

HTTP basic authentication, 110–111, 119–121
HTTP local storage, 174
HTTP redirects, 113–114

I

iBeacons, 244–247
- CBPeripheralManager, 246
- CLBeaconRegion, 244–246
- CLLocationManager, 244
- startMonitoringForRegion, 244

iBoot, 4
iCloud, 35, 111, 161, 212, 219
- avoidance of, 187

IDA Pro, 94
identifierForVendor, 234
iExplorer, 28–29
iGoat, 178
image list, 87
implementation, declaring, 16–17
Info.plist, 33
init, 19
initialization vector (IV), 226–227
initWithCoder, 21–22
initWithContentsOfURL, 206
injection attacks, 199–207
- cross-site scripting (XSS), 199–202
  - input sanitization, 200–201
  - output encoding, 200–202
- displaying untrusted data, 202
- predicate injection, 204–205
- SQL injection, 203–204
  - parameterized SQL, 203–204
  - SQLite, 203–204
- XML injection, 207
  - XML external entities, 205–206
  - XPath, 207

input sanitization, 200–201
installipa command, 80
InstaStock, 12
Instruments, 55–57
integer overflow, 196–198
- example, 197–198
- preventing, 198

interface, declaring, 15–16
interprocess communication, *see* IPC (interprocess communication)
Introspy, 100–103
iOS-targeted web apps, 147–160
IPA Installer Console, 78
.ipa packages, 80
IPC (interprocess communication), 131–145
- application extensions, 131, 140
  - extensionPointIdentifier, 144
  - extension points, 140
  - isContentValid, 143
  - NSExtensionActivationRule, 142
  - NSExtensionContext, 143
  - NSExtensionItem, 143
  - shouldAllowExtensionPoint -Identifier, 143
  - third-party keyboards, 143–144
- canOpenURL, 138
- handleOpenURL, 136
- isContentValid, 143
- openURL, 132–137
- sourceApplication, 136
- UIActivity, 139–140
- UIPasteboard, 144
- universal links, 137–138
- URL schemes, 132–133
  - CFBundleURLSchemes, 133
  - defining, 132–133
  - hijacking, 136–137
  - validating URLs and senders, 134

iproxy command, 84
isContentValid, 143
IV (initialization vector), 226–227
ivars, 15–17, 91

J

jailbreak detection, 9–10
- futility of, 9

jailbreaking, 4, 9–10, 77
JavaScript, 11
- executing in Cordova, 154–157
- executing in UIWebView, 149–150
- stringByEvaluatingJavaScriptFrom-String, 149–150

JavaScript–Cocoa bridging, 150–157
JavaScriptCore, 150–154
- blocks, 150–151
- JSContext, 152–154
- JSExport, 151–152

Jekyll, 12

© 2016 David Thiel

just-in-time (JIT) compiler, 8–9, 149
JRSwizzle, 25
JSContext, 152–154
JSExport, 151–152

K

kCFStreamSSLLevel, 129
Keychain API, 6–7, 113, 186, 211
  backups, 212
  iCloud synchronization, 219
  kSecAttrAccessGroup, 218–219
  protection attributes, 212–214
  SecItemAdd, 214–215, 219
  SecItemCopyMatching, 216
  SecItemDelete, 216
  SecItemUpdate, 215
  shared Keychains, 218–219
  usage, 214–217
  wrappers, 217–218
key derivation, 227–228
keylogging, 175–177
killall command, 79, 101
kSecAttrAccessGroup, 218–219
kSecAttrAccessible, 220
kSecAttrSynchronizable, 219

L

LAContext, 231–232
ldid command, 97
LDID (link identity editor), 97
legacy issues, from C, 189–198
  buffer overflows, 193–196
    example, 194–195
    preventing, 195–196
  format string attacks, 190–193
    NSString, 192–193
    preventing, 191–193
  integer overflow, 196–198
    example, 197–198
    preventing, 198
libc, 8
Library directory, 35–37
  Application Support directory, 35
  Caches directory, 35–36, 187
    Snapshots directory, 36
  Cookies directory, 36
  Preferences directory, 36
  Saved Application State directory, 37
LIKE operator, 205
link identity editor (LDID), 97
lipo command, 78, 85
lldb, 62–81, 83–84, 191
  backtrace (bt) command, 65–66
  breakpoints, 62
    actions, 70–72, 164
    conditions, 72
    enabling/disabling, 64
    setting, 62–64
  expr command, 69
  frame select command, 66–67
  frame variable command, 66
  image list, 87
  print object command, 67–68
llvm, 90
Local Authentication API, 231–232
Logos, 98
loopback interface, 46–47
Lua, 12

M

M7 processor, 242
MAC (mandatory access control), 4–5
MAC address, 234
Mach-O binary format, 77, 85
MachOView, 88
MacPorts, 94
malloc, 197–198
mandatory access control (MAC), 4–5
MATCHES operator, 205
MCEncryptionNone, 126
MCEncryptionOptional, 126
MCEncryptionRequired, 126
MCSession, 126
message passing, 13–15
method swizzling, 23–25
Mobile Safari, 44
MobileTerminal, 78
multipeer connectivity, 125–127
  certificateHandler, 126
  didReceiveCertificate, 126
  encryption, 125–127

N

netcat command, 78
networking, 107–129
  AFNetworking, 122–124
    certificate pinning, 123–124

iOS Application Security: The Definitive Guide for Hackers and Developers
© 2016 David Thiel

ASIHTTPRequest, 122, 124–125
backgroundSessionConfiguration, 117
CFStream, 48, 107, 128–129
ephemeralSessionConfiguration, 117
multipeer connectivity, 125–127
certificateHandler, 126
didReceiveCertificate, 126
encryption, 125–127
NSInputStream, 49
NSOutputStream, 49
NSStream, 48, 107, 127–128
NSURLSession, 122
URL loading system, 107–122
HTTP basic authentication, 110–111, 119–121
HTTP redirects, 113–114
NSURLConnection, 48, 108
NSURLConnectionDelegate, 109
NSURLCredential, 120
NSURLCredentialStorage, 110–111
NSURLRequest, 108
NSURLResponse, 108
NSURLSession, 48, 117
NSURLSessionConfiguration, 120–121
NSURLSessionTaskDelegate, 119
sharedCredentialStorage, 120–122
stored URL credentials, 121–122
Notification Center, 224–225
NSCoder, 185–187
NSCoding, 21–22
NSData, 113
NSExtensionContext, 143
NSExtensionItem, 143
NSFileManager, 221–223
NSHTTPCookieStorage, 237
NSHTTPRequest, 122
NSInputStream, 49
NSLog, 95, 161–164, 192
disabling, 163
NSNotificationCenter, 224–225
NSOperation, 122
NSOutputStream, 49
NSPredicate, 204–205
NSStream, 48, 107, 127–128
NSString, 192–193, 195, 202
NSURAuthenticationMethodClientCertificate, 112
NSURL, 188
NSURLCache, 74–75, 150
NSURLConnection, 48, 108, 117
NSURLConnectionDelegate, 109, 114
NSURLCredential, 113, 120
NSURLCredentialStorage, 110–111, 121
NSURLIsExcludedFromBackupKey, 35, 187–188
NSURLProtectionSpace, 109–111, 122
NSURLProtocol, 155
NSURLRequest, 108, 148–149
NSURLResponse, 108
NSURLSession, 48, , 117–122
NSURLSessionConfiguration, 117–119
NSURLSessionDataTask, 18
NSURLSessionTaskDelegate, 119
NSUserDefaults, 36, 37, 178
NSUUID, 234
NSXMLParser, 205–206

O

Objective-C, 13–25
blocks
declaring, 18
exposing to JavaScript, 150–151
categories, 22–23
code structure, 15–17
delightfulness of, 13
garbage collection, 18
implementation, declaring, 16–17
ivars, 15–16
message passing, 14–15
private methods, lack thereof, 16
property synthesis, 17
reference counting, 18–19
odcctools, 78, 84
OpenSSH, 78
OpenSSL, 94, 228–229
openssl command, 138
openURL, 132–137
otool, 53, 78, 84–86,
inspecting binaries, 90–92
output encoding, 200–202

P

p12 file, 113
parameterized SQL, 203–204
pasteboardDB file, 165–167

pasteboards, 164–169
  canPerformAction, 168
  disabling copy/paste, 168–169
  pasteboardDB file, 165–167
  pasteboardWithUniqueName, 165–167
  UISearchBar, 165
pasteboardWithUniqueName, 165–167
PhoneGap, 11, 150
physical attackers, 161
PIE (position-independent executable), 53–54
  removing, 87
plist files, 29–31
  converting, 30–31
  XML, 29–30
plutil command, 30–31
popen, 10
position-independent executable (PIE), 53–54
  removing, 87
predicate injection, 204–205
  LIKE operator, 205
  MATCHES operator, 205
  wildcards, 204–205
predicates, 205
predicateWithFormat, 204–205
Preferences directory, 36
printf command, 87, 190–192
print object command, 67–68
privacy concerns, 233–248
  advertisingTrackingEnabled, 235
  bluetooth low energy (BTLE), 244
  cookies, 237–238
  Do Not Track, 236–237
  geolocation, 238–240
    accuracy, 239
    CLLocationManager, 240
    locationManager, 244
    risks, 238–239
  GPS, 238
  HealthKit, 240–241
  iBeacons, 244–247
    CBPeripheralManager, 246
    CLBeaconRegion, 244–246
    CLLocationManager, 244
    startMonitoringForRegion, 244
  M7 processor, 242
  MAC address, 234
  microphone, 233
  privacy policies, 247–248
  proximity tracking, 244–247
  requesting permission, 243
  unique device identifier (UDID), 233–235
    advertisingIdentifier, 235
    identifierForVendor, 234
    NSUUID, 234
    uniqueIdentifier, 234
private methods, 16
property synthesis, 17
protocols, 20–22
  declaring, 21–22
proximity tracking, 244–247
proxy setup, 43–50

Q

Quick Look, 35, 68
QuickType, 177

R

reference counting model, 18–19
  retain and release, 18–19
references, strong and weak, 19
release, 18–19
remote device wipe, 5, 6
removeAllCachedResponses, 75
respringing, 79, 101
restorationIdentifier, 184–185
retain, 18–19
return-to-libc attack, 8
RNCryptor, 186, 230
rootViewController, 183
rsync command, 78

S

safe string APIs, 195
Sandbox, 4–5
Saved Application State directory, 37
Seatbelt, 4–5
SecCertificateRef, 112–113
SecIdentityRef, 112–113
SecItemAdd, 186, 212, 215, 219
SecItemCopyMatching, 216
SecItemDelete, 216
SecItemUpdate, 215
SecRandomCopyBytes, 227
SecTrustRef, 112–113

iOS Application Security: The Definitive Guide for Hackers and Developers
© 2016 David Thiel

Secure Boot, 4
SecureNSCoder, 186–187
securityd, 7
serialization, 21
setAllowsAnyHTTPSCertificate, 108
setJavaScriptCanOpenWindowsAutomatically, 159
setJavaScriptEnabled, 159–160
setResourceValue, 188
setSecureTextEntry, 175–177
setShouldResolveExternalEntities, 206
Shared directory, 37
sharedCredentialStorage, 120–122
sharedHTTPCookieStorage, 237
shared Keychains, 218–219
shouldAllowExtensionPointIdentifier, 143
should messages, 20
shouldSaveApplicationState, 20
shouldStartLoadWithRequest, 148
sideloading, 77–80
signed integer, 196
signedness, 51, 196
Simulator, 43–46
  camera, 43
  case-sensitivity, 43
  installing certificates, 44
  Keychain, 43
  PBKDF2, 43
  proxying, 44–46
  trust store, 44
SpringBoard, 79
SQL injection, 201, 203–204
  parameterized SQL, 203–204
  SQLite, 203–204
SSH, 28, 82
SSL, *see* TLS (Transport Layer Security)
SSL Conservatory, 115–117
SSL Killswitch, 96–97
stack, 8, 53–54, 190, 193
startMonitoringForRegion, 244
state preservation, 184–187
  leaks, 184–185
  restorationIdentifier, 184–185
  secure, 185–187
static analysis, 54
std::string, 195
strcat, 195
strcpy, 194, 195
stringByEvaluatingJavaScriptFromString, 149–150
strlcat, 195–196
strlcpy, 195–196
strong references, 19
stunnel, 46
subclassing, 23
synthesize, 17
syslog, 162, 190

T

task_for_pid-allow, 82
tcpdump command, 78
tcpprox, 49–50
TCP proxying, 49–50
test devices, 42
text segment, 85–86
Theos, 97–98
thin binaries, 85
ThisDeviceOnly, 212
TLS (Transport Layer Security), 108–119, 127–129
  BEAST attack, 118
  bypassing validation, 44–47, 119
  certificate pinning, 114–117, 123–124
  CRIME attack, 118
  mutual authentication, 112–113
  setAllowsAnyHTTPSCertificate, 108
  validation, category bypasses, 22
tmp directory, 37, 80, 187
Today screen, 131
TOFU (trust on first use), 127
TouchID, 231–232
  LAContext, 231–232
Transport Layer Security, *see* TLS (Transport Layer Security)
Tribbles, 51
trust on first use (TOFU), 127
tweaks, Cydia Substrate, 97

U

UDID (unique device identifier), 233–235
  advertisingIdentifier, 235
  identifierForVendor, 234

© 2016 David Thiel

- UDID (unique device identifier), *continued*
  - NSUUID, 234
  - uniqueIdentifier, 234
- UIActivity, 139–140
- UIAlertView, 183
- UI Layers, 182–183
- UIPasteBoard, 144, 164–169
- UIPasteboardNameFind, 165
- UIPasteboardNameGeneral, 165
- UIRequiredDeviceCapabilities, 34
- UIResponderStandardEditActions, 169
- UISearchBar, 165, 175
- UITextField, 175
- UITextView, 175
- UIView, 182–183
- UIWebView, 200, 201
- UIWindow, 182–183
- unique device identifier (UDID), 233–235
  - advertisingIdentifier, 235
  - identifierForVendor, 234
  - NSUUID, 234
  - uniqueIdentifier, 234
- uniqueIdentifier, 234
- universal links, 137–138
- unsigned integer, 196
- URL loading system, 107
  - credential persistence types, 111
  - HTTP basic authentication, 110–111
  - HTTP redirects, 113–114
  - NSURLConnection, 108
  - NSURLConnectionDelegate, 109
  - NSURLCredential, 120
  - NSURLCredentialStorage, 110–111
  - NSURLRequest, 108
  - NSURLResponse, 108
  - NSURLSession, 117–122
  - NSURLSessionConfiguration, 117–119
  - NSURLSessionTaskDelegate, 119
  - sharedCredentialStorage, 120–122
  - stored URL credentials, 121–122
- URL schemes, 132–133
  - CFBundleURLSchemes, 133
  - defining, 132–133
  - hijacking, 136–137
  - validating URLs and senders, 134
- USB, TCP proxying, 84
- usbmuxd command, 84
- user preferences, 178
- UUID, 27
- uuidgen, 244

V

- Valgrind, 55
- vbindiff command, 78, 88
- vfork, 10
- *.vimrc* file, 30
- vmaddr, 88

W

- wardriving, 238
- warning policies, 51
- watchdog, 58–59
- watchmedo command, 58–59
- weak_classdump, 93
- weak references, 19
- web apps, 147–160
- WebViews, 9, 147–160
  - Cordova, 154–157
    - risks, 156
    - XmlHttpRequest, 155
  - JavaScript, 149
    - executing in Cordova, 154–157
    - executing in UIWebView, 149–150
    - stringByEvaluatingJavaScriptFromString, 149–150
  - JavaScript–Cocoa bridging, 150–157
  - JavaScriptCore, 149–154
    - blocks, 150–151
    - JSContext, 152–154
    - JSExport, 151–152
  - just-in-time (JIT) compiler, 149
  - Nitro, 148, 149
  - NSURLRequest, 148–149
  - UIWebView, 147–150
  - WebKit, 11, 147–148
  - WKWebView, *see* WKWebView
- whitelisting, 149, 152, 200–201
- will messages, 20
- willSendRequestForAuthenticationChallenge, 112
- Wireshark, 46
- WKPreferences, 160
- WKWebView, 148 158–160
  - addUserScript, 159
  - benefits of, 159–160

iOS Application Security: The Definitive Guide for Hackers and Developers
© 2016 David Thiel

- didFinishNavigation, 159–160
- hasOnlySecureContent, 159–160
- setJavaScriptCanOpenWindows-Automatically, 159
- setJavaScriptEnabled, 159–160
- WKPreferences, 160
- WKUserScript, 159
- WKWebViewConfiguration, 160

X

xcodebuild, 190
Xcode setup, 50–53, 55
- warnings, 51–53

Xcon, 10
XML injection, 207
- NSXMLParser, 205–206
- XML external entities, 205–206
- XPath, 207

XN (eXecute Never), 8–9
XPath, 207
XSS (cross-site scripting), 199–200
- input sanitization, 200–201
- output encoding, 201–202

xxd command, 88

© 2016 David Thiel