/data

The /data partition is where all the user's personal data resides. Providing a separate partition for this provides several important advantages:

- /data is decoupled from the underlying Android OS version: System upgrade and recovery can thus wipe and rewrite the entire /system partition, without affecting the user's data in any way. Conversely, the device can quickly be reset and all personal data wiped by formatting /data, which is exactly what happens during a "factory reset".
- /data may be encrypted, if the user requires it: Encryption, however efficient, adds a degree of latency, since reading and writing involves decryption and encryption, respectively. Because, by design, /system contains no sensitive information, there is no need to encrypt it, and therefore this latency is avoided.
- /data may also be made non-executable (i.e. mounted with the `noexec` option, or enforced with SELinux). As of KitKat, this isn't a default option. Doing so, however, would not only would make it more true to its name, but would greatly mitigating an attack vector for malware, since the latter would have no writable partition that it can drop executables to. This would not affect legitimate Dalvik apps, because DEX runs in a virtual machine, but would likely impact rooting (for example, by requiring a remount, the same as it does with /system).

The /data partition is mounted with `nosuid`, which makes rooting the device a bit more of a cumbersome operation - assuming that root access is somehow obtained, the `su` binary (which makes for an efficient, persistent backdoor) must be placed in /system, which is read-only. In practice, this is only a minor obstacle, since it's a simple enough operation to remount /system in read-write mode. Nonetheless, this is an example of defense-in-depth, and could actually prove effective when /system is cryptographically hashed, as with KitKat's dm-verity (q.v. Chapter 21).

Table fs-data shows the contents of the /data partition. Note vendors and carriers may place additional files or directories.

Table fs-data: Directories under the /data partition

Directory	Notes
anr	Used by `dumpstate` to record stack traces of non-responsive Android Apps. Stack traces are recorded into `traces.txt`, as per the `dalvik.vm.stack-trace-file` property.
app	User-installed applications. Downloaded .apk files can be found here.
app-asec	Application asec containers (described later in this chapter).
app-lib	JNI libraries of applications (both system and user-installed) can be found here.
app-private	Provided for application private storage; In practice largely unused, since asec provides better security.
backup	Used by the backup service
bugreports	Used exclusively by `bugreport` for generated reports, which include a text file and screenshot (png), both named `bugreport-`*`yyyy-mm-dd-hh-mm-ss`*.
dalvik-cache	The optimized `classes.dex` of system and user applications. Each app's dex is preceded by the path to its apk, with "@" replacing the path separator (e.g. `system@framework@bu.jar@classes.dex`).
data	Data directories for installed applications, in reverse DNS format. Discussed next
dontpanic	Formerly used to store Android panic console and threads. Unused.
drm	Used by Android's Digital Rights Management
local	A readable/writable temporary directory for uid shell (usable in ADB sessions)
lost+found	Automatically generated directory for fsck operations on /data. Empty (unless the filesystem crashed, in which case it may contain unlinked inodes)
media	Used by the `sdcard` service for mounted media
mediadrm	Used by the Media DRM service
misc	"Miscellaneous" data and configuration directories for components. q.v. Table 2-dm.

Table fs-data (cont): Directories under the /data partition

Directory	Notes
nfc	Stores NFC parameters
property	Contains persistent properties (i.e. saved across device reboots). Each property is saved in its own file, with the property name serving as the file name
resource-cache	Resources cached by the AssetManager (described in Chapter 5).
security	commonly empty
ssh	For devices which provide the Secure Shell service. (Usually empty)
system	A multitude of system configuration files, shown in table f-datasys
tombstones	Application crash reports generated by debuggerd. Due to limited filesystem space, full core dumps are not feasible. The debuggerd provides basic autopsy services in absence of a core dump. Some vendors allocate a separate partition to this directory.
user	JB and later: provides "multi-user" capabilties, by symlinking user numbers (0,1..) to directories with installed applications and data for those users. In a single user system, 0 links to /data/data.

/data/data

The somewhat redundantly-named /data/data is the directory where all applications - both system and user-installed - store their information. Each application gets its own subdirectory, in reverse DNS format, which is `chmod 751` (`rwxr-x--x`), under the uid/gid of the owning application. The /data/data directory itself is `chmod 771 system system`, and therein lies a tenet of Android's security model: /data/data is executable (i.e. `cd`-able*) to all applications, but unreadable (so applications can't enumerate "neighbor" directories). The burden of securing specific application files, however, rests on each and every application, as the per-app directories are freely executable, though are unreadable by anyone other than the owner.

The /data/data per-app subdirectory is the only location in the entire filesystem which is writable by apps. Coupled with the fact that the stock applications for location, texting and calls can be found on every Android device, this makes several locations in it key for performing forensics. Subdirectories of particular interest are shown in table 2-appdata:

Table 2-appdata: But a few of the app directories of interest in /data/data

App subdirectory	Used by	Contains
com.android.providers.contacts	Phone Contacts	Virtually every tidbit of information which might be of remote interest on the device, in databases/contacts2.db: a SQLite3 master contact database, including tables like contacts (All contacts stored on the device) and calls (Log of last calls). files/thumbnail_photo_*xxxxx*.png are individual thumbnails of contacts.
com.android.providers.calendar	Calendar	Calendar: databases/calendar.db (in the events table).
com.android.providers.telephony	Messaging	Multimedia(MMS)/text(SMS) message database: database/mmssms.db
com.google.android.apps.maps	Google Maps	Destinations looked up: gmm_myplaces.db, gmm_storage.db and log_events.db. cache/http contains map tiles.
com.google.android.gm	GMail	databases/mailstore.*email*.db: a SQLite3 database containing all the user's mail which has been downloaded to the device, for each registered *email* address (in the messages table). Viewed attachments are stored in cache/*email*.
com.android.chrome	Chrome browser	State of Chrome browser (which replaces the old Android built-in com.android.browser). Files of interest include the cache/ directory (browser cache), and the app_chrome/Default/ directory, which contains many important SQLite3 databases, such as History and Archived History (browsing history in urls table), Login Data (saved credentials, in logins table) and Cookies.

* - The meaning of `+x` on a directory is slightly different than on a file: `+x` means you can `cd` into the directory. Note that this does **not** necessarily imply you can read the contents, which requires `+r`.

Note the list is far from comprehensive. Nonetheless, if you're interested if you're finding specific application files, it's fairly straightforward to look for the app in /data/data by the reverse DNS notation (which matches the APK name). From there, it's a simple matter of grabbing the files (on a rooted device), then using `sqlite3` on the various databases and `file` to identify and view others. This is shown in the following experiment:

Experiment: Device forensics through /data/data

On a rooted device, you can easily examine application data directories with SQLite3. The Android emulator image contains a `sqlite3` binary in /system/xbin, as do most rooting packages (for reasons which should now be fairly obvious).

Taking as an example Chrome, start the browser and navigate to any site of your choice. To look at the history database you will need to kill the process, since it holds a lock on the database. From there, a simple SQL query reveals all.

Output 2-ch: Examining Chrome's history with `sqlite3`

```
root@htc_m8wl:/ # cd /data/data/com.android.chrome
# Using ".schema" shows the table definition:

root@htc_m8wl:/data/data/com.android.chrome #  sqlite3 app_chrome/Default/History
sqlite> .schema urls
CREATE TABLE urls(id INTEGER PRIMARY KEY,url LONGVARCHAR,title LONGVARCHAR,
visit_count INTEGER DEFAULT 0 NOT NULL,typed_count INTEGER DEFAULT 0 NOT NULL,
last_visit_time INTEGER NOT NULL,hidden INTEGER DEFAULT 0 NOT NULL,
favicon_id INTEGER DEFAULT 0 NOT NULL);
CREATE INDEX urls_url_index ON urls (url);
sqlite> select * from urls where url like "%android%";
id|url                          |title                  | | |last_visit_time  | |
52|http://newandroidbook.com/         |Android Internals      |2|2|13054934895637919|0|0
53|http://newandroidbook.com/TOC.html|Android Internals::TOC |1|0|13054934883061164|0|0
```

Demonstrating the same on the contacts2.db in /data/data/com.android.providers.contacts/databases:

Output 2-calls: Examining the call log

```
sqlite> .schema calls
CREATE TABLE calls (_id INTEGER PRIMARY KEY AUTOINCREMENT,number TEXT,
presentation INTEGER NOT NULL DEFAULT 1,date INTEGER,duration INTEGER,
...
# E.g. find all toll free calls
sqlite> select _id, number, date, duration from calls where number like "%800%";
id|number     |date         |duration
2 |18001750930|1396019679278|0
16|18007562000|1402005179460|0
```

Another useful forensic trick - which merely requires the device to be unlocked, and not necessarily rooted - is to connect the device via adb to a host, and issue an `adb backup` request for the packages of interest. This calls on the the `BackupManagerService`, which - by virtue of running as system - can access /data/data with no restriction, and not only read all the files of any app, but also conveniently transport them to the host. (The backup process and the `BackupManagerService` are both described in detail in Chapter 5, respectively).

When initiating a backup, the `BackupManagerService` will prompt the user for confirmation (hence the need for an unlocked device). If the operation is approved, a backup archive is created on the host with an .ab (Android Backup) extension. Backups can be easily extracted on the host using the `ab` tool from the book's companion website, introduced in an experiment in Chapter 5.

/data/misc

The /data/misc directory contains miscellaneous data and configuration directories for Android's subsystems. Contrary to its name, the contents include some of the most important files in the system. More detail can be found in Table 2-dm:

Table 2-dm: Directories in /data/misc

Directory	Contents
adb	Trusted ADB host public-keys (as of JB)
bluetooth	BlueZ (< 4.2 bluetooth subsystem) configuration files
bluedroid	Bluetooth subsystem (>4.2) configuration files
dhcp	Contains PID file of dhcp ctdent daemon, and any active lease
keychain	Android built-in certificate pins and blacklists
keystore	Per-user keystore data
sensors	Sensor debug data
sms	Contains the sms codes database
systemkeys	ASEC container keys (AppsOnSD.sks)
vpn	VPN state configuration files
wifi	Wi-fi subsystem configuration files (e.g. wpa_supptdcant.conf), and sockets

/data/system

Another important subdirectory of /data is /data/system, as it contains files critical to maintaining the state of device. As cab be expected, access is restricted to system:system, so if your device is not rooted, you can't see any of the files shown in table f-datasys:

Table f-datasys:: The contents of /data/system

Directory	Notes
appops.xml	Used by the AppOps service, which controls application permissions.
batterystats.bin	Used by the BatteryStats service, which keeps statistics on power usage by applications.
called_pre_boots.dat	Used by the ActivityManager to hold pre boot broadcast receivers
device_policies.xml	Configuration file used by the DevicePolicyManagerService.
dropbox/	Directory used by the DropBox.
entropy.dat	System entropy store, used by EntropyMixer for random number generation.
gesture.key	Lockscreen pattern or PIN, as discussed in Chapter 21.
framework_atlas.config	Used by the AssetAtlasService, which packs bitmaps into a single file.
ifw/	Intent FireWall rulebase (q.v. Chapter 21).
locksettings.db*	Lock screen settings: Contains device lock policy (q.v. Chapter 21).
netpolicy.xml	Configuration file used by the NetworkPolicyManagerService.
netstats/	Directory used to hold NetworkStatsService statistics - by device, uid, or xt. Previous versions of Android simply dropped the files netstats.bin, netstats_xt.bin and netstats_uid in /data/system.
packages.list	Lists all installed packages (APKs) in the system - both system and user.
packages.xml	Used by the PackageManager, and contains metadata on all installed packages in the system. Discussed in Chapter 6.
password.key	Lockscreen password hash, as discussed in Chapter 21.
procstats/	Directory used to store files for the ProcessStats service
registered_services/	Directory used by android.content.pm.RegisteredServicesCache
usagestats/	Used to store files for the UsageStats service. In particular, usage-history.xml
users/	Android's "Multi-User" support. userlist.xml holds the uids, with *uid*.xml files and a *uid* directory for each user. Described in more detail in Chapter 21.