Secure an Android device

10-13 minutes

Android incorporates industry-leading security features and works with developers and device implementers to keep the Android platform and ecosystem safe. A robust security model is essential to enable a vigorous ecosystem of apps and devices built on and around the Android platform and supported by cloud services. As a result, through its entire development lifecycle, Android has been subject to a rigorous security program.

Android is designed to be open. Android apps use advanced hardware and software, as well as local and served data, exposed through the platform to bring innovation and value to consumers. To realize that value, the platform offers an app environment that protects the confidentiality, integrity, and availability of users, data, apps, the device, and the network.

Securing an open platform requires a strong security architecture and rigorous security programs. Android was designed with multilayered security that's flexible enough to support an open platform while still protecting all users of the platform. For information about reporting security issues and the update process, see Security Updates and Resources.

Android is designed for developers. Security controls were designed to reduce the burden on developers. Security-savvy developers can easily work with and rely on flexible security controls. Developers less familiar with security are protected by safe defaults.

In addition to providing a stable platform to build upon, Android gives additional support to developers in a number of ways. The Android security team looks for potential vulnerabilities in apps and suggests ways to fix those issues. For devices with Google Play, Play Services delivers security updates for critical software libraries, such as OpenSSL, which is used to secure app communications. Android security released a tool for testing SSL (nogotofail) that helps developers find potential security issues on whichever platform they are developing.

Android also leverages underlying hardware support for security. For example, ARM TrustZone technology is used to provide secure storage for cryptographic keys as well as attestations of boot integrity. DICE is used to measure firmware that is loaded prior to booting Android. This enables remote verification that the firmware isn't affected by known, critical vulnerabilities which could be exploited to harm both developers and users.

More information for Android app developers can be found on developer.android.com.

Android is designed for users. Users are provided visibility into the permissions requested by each app and control over those permissions. This design includes the expectation that attackers would attempt to perform common attacks, such as social engineering attacks to convince device users to install malware, and attacks on third-party apps on Android. Android was designed to both reduce the probability of these attacks and greatly limit the impact of the attack in the event that it was successful. Android security continues to progress after the device is in the user's hands. Android works with partners and the public to provide patches for any Android device that is continuing to receive security updates.

More information for end users can be found in the Nexus help center, Pixel help center, or your device manufacturer’s help center.

This page outlines the goals of the Android security program, describes the fundamentals of the Android security architecture, and answers the most pertinent questions for system architects and security analysts. It focuses on the security features of Android's core platform and doesn't discuss security issues that are unique to specific apps, such as those related to the browser or SMS app.

Background

Android provides an open source platform and app environment for mobile devices.

The sections and pages below describe the security features of the Android platform. Figure 1 illustrates the security components and considerations of the various levels of the Android software stack. Each component assumes that the components below are properly secured. With the exception of a small amount of Android OS code running as root, all code above the Linux kernel is restricted by the Application Sandbox.

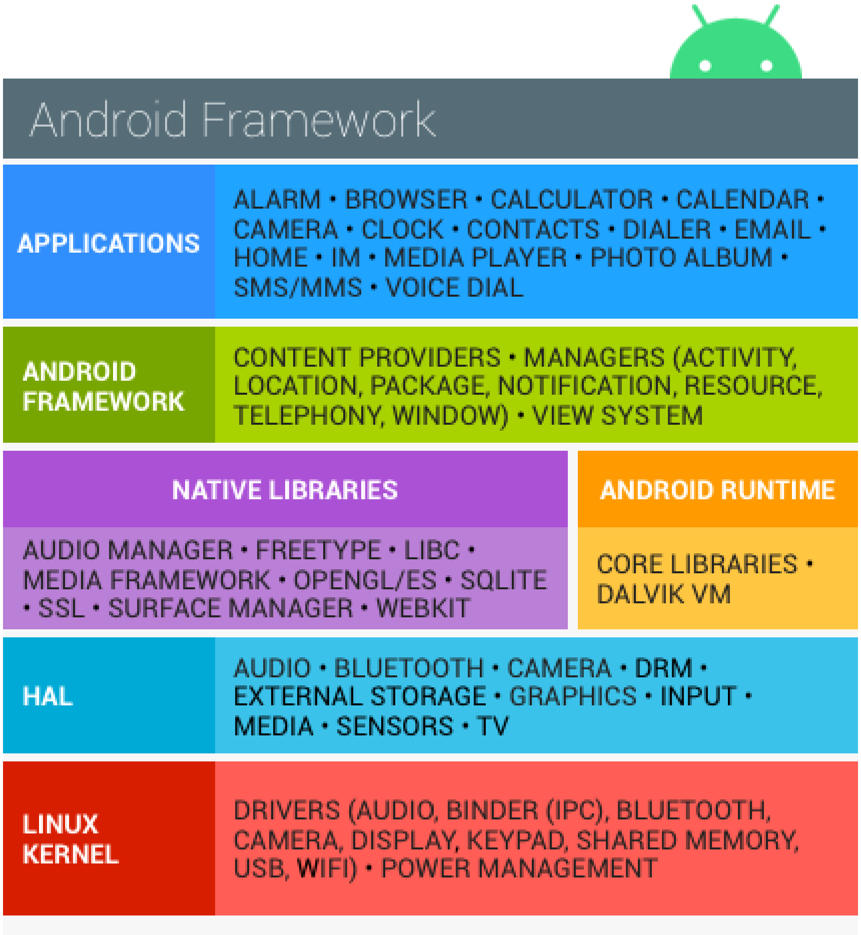
Android Framework
APPLICATIONS
ALARM • BROWSER • CALCULATOR • CALENDAR • CAMERA • CLOCK • CONTACTS • DIALER • EMAIL • HOME • IM • MEDIA PLAYER • PHOTO ALBUM • SMS/MMS • VOICE DIAL
ANDROID FRAMEWORK
CONTENT PROVIDERS • MANAGERS (ACTIVITY, LOCATION, PACKAGE, NOTIFICATION, RESOURCE, TELEPHONY, WINDOW) • VIEW SYSTEM
NATIVE LIBRARIES
AUDIO MANAGER • FREETYPE • LIBC • MEDIA FRAMEWORK • OPENGL/ES • SQLITE • SSL • SURFACE MANAGER • WEBKIT
ANDROID RUNTIME
CORE LIBRARIES • DALVIK VM
HAL
AUDIO • BLUETOOTH • CAMERA • DRM • EXTERNAL STORAGE • GRAPHICS • INPUT • MEDIA • SENSORS • TV
LINUX KERNEL
DRIVERS (AUDIO, BINDER (IPC), BLUETOOTH, CAMERA, DISPLAY, KEYPAD, SHARED MEMORY, USB, WIFI) • POWER MANAGEMENT

Figure 1. Android software stack

The main Android platform building blocks are:

- **Device hardware:** Android runs on a wide range of hardware configurations including mobile phones, tablets, watches, automobiles, smart TVs, OTT gaming boxes, and set-top-boxes. Android is processor-agnostic, but it takes advantage of some hardware-specific security capabilities such as ARM eXecute-Never.
- **Android operating system:** The core operating system is built on top of the Linux kernel. All device resources, like camera functions, GPS data, Bluetooth functions, telephony functions, and network connections are accessed through the operating system.
- **Android Application Runtime:** Android apps are most often written in the Java programming language and run in the Android runtime (ART). However, many apps, including core Android services and apps, are native apps or include native libraries. Both ART and native apps run within the same security environment, contained within the Application Sandbox. Apps get a dedicated part of the file system in which they can write private data, including databases and raw files.

Android apps extend the core Android operating system. There are two primary sources for apps:

- **Preinstalled apps:** Android includes a set of preinstalled apps including phone, email, calendar, web browser, and contacts. These function as user apps and they provide key device capabilities that can be accessed by other apps. Preinstalled apps may be part of the open source Android platform, or they may be developed by a device manufacturer for a specific device.
- **User-installed apps:** Android provides an open development environment that supports any third-party app. Google Play offers users hundreds of thousands of apps.

Google security services

Google provides a set of cloud-based services that are available to compatible Android devices with Google Mobile Services. While these services aren't part of the Android Open Source Project (AOSP), they are included on many Android devices. For more information on some of these services, see Android Security's 2018 Year in Review.

The primary Google security services are:

- **Google Play:** Google Play is a collection of services that allow users to discover, install, and purchase apps from their Android device or the web. Google Play makes it easy for developers to reach Android users and potential customers. Google Play also provides community review, app license verification, app security scanning, and other security services.
- **Android updates:** The Android update service delivers new capabilities and security updates to selected Android devices, including updates through the web or over the air (OTA).
- **App services:** Frameworks that allow Android apps to use cloud capabilities such as (backing up) app data and settings and cloud-to-device messaging (C2DM) for push messaging.
- **Verify Apps:** Warn or automatically block the installation of harmful apps, and continually scan apps on the device, warning about or removing harmful apps.
- **SafetyNet:** A privacy preserving intrusion detection system to assist Google tracking, mitigate known security threats, and identify new security threats.
- **SafetyNet Attestation:** Third-party API to determine whether the device is CTS compatible. Attestation can also identify the Android app communicating with the app server.
- **Android Device Manager:** A web app and Android app to locate lost or stolen device.

Security program overview

The key components of the Android Security Program include:

- **Design review:** The Android security process begins early in the development lifecycle with the creation of a rich and configurable security model and design. Each major feature of the platform is reviewed by engineering and security resources, with appropriate security controls integrated into the architecture of the system.
- **Penetration testing and code review:** During the development of the platform, Android-created and open source components are subject to vigorous security reviews. These reviews are performed by the Android Security Team, Google's Information Security Engineering team, and independent security consultants. The goal of these reviews is to identify weaknesses and possible vulnerabilities well before major releases, and to simulate the types of analysis that are performed by external security experts upon release.
- **Open source and community review:** AOSP enables broad security review by any interested party. Android also uses open source technologies that have undergone significant external security review, such as the Linux kernel. Google Play provides a forum for users and companies to provide information about specific apps directly to users.
- **Incident response:** Even with these precautions, security issues may occur after shipping, which is why the Android project has created a comprehensive security response process. Full-time Android security team members monitor the Android-specific and the general security community for discussion of potential vulnerabilities and review security bugs filed on the Android bug database. Upon the discovery of legitimate issues, the Android team has a response process that enables the rapid mitigation of vulnerabilities to ensure that potential risk

to all Android users is minimized. These cloud-supported responses can include updating the Android platform (AOSP updates), removing apps from Google Play, and removing apps from devices in the field.
- **Monthly security updates:** The Android security team provides monthly updates to Google Android devices and all our device manufacturing partners.

Platform security architecture

Android seeks to be the most secure and usable operating system for mobile platforms by repurposing traditional operating system security controls to:

- Protect app and user data
- Protect system resources (including the network)
- Provide app isolation from the system, other apps, and from the user

To achieve these objectives, Android provides these key security features:

- Robust security at the OS level through the Linux kernel
- Mandatory app sandbox for all apps
- Secure interprocess communication
- App signing
- App-defined and user-granted permissions