看雪学院-浅入浅出Android安全 笔记

原创

Cytosine 于 2017-02-09 23:00:26 发布 333 收藏
分类专栏： 笔记_浅入浅出Android安全 Android Security Security Android 逆向 文章标签： android 安全


笔记_浅入浅出Android安全 同时被 3 个专栏收录
1 篇文章 0 订阅
订阅专栏


Android Security
1 篇文章 0 订阅
订阅专栏


Security
1 篇文章 0 订阅
订阅专栏

Note 看雪学院-浅入浅出Android安全

原作地址：http://www.kanxue.com/?article-read-547.htm

翻译作者：飞龙

Android 由四个层组成：**Linux** 内核，本地用户空间，应用程序框架和应用程序层。 有时本地用户空间和应用程序框架层被合并到一个层中，称为 **Android** 中间件层。

Linux内核

负责进程，内存，通信，文件系统管理等。

本地用户空间层

这个层的第一个组件是硬件抽象层（ HAL） ，它与 Linux内核和本地用户空间层之间实际上是模糊的。

在 Linux 中，硬件驱动程序嵌入到内核中或作为模块动态加载。

虽然 Android 是建立在 Linux 内核之上，它利用了一种非常不同的方法来支持新的硬件。相反，对于每种类型的硬件，Android 定义了一个 API，它由上层使用并用于与这种类型的硬件交互。

硬件供应商必须提供一个软件模块，负责实现在 Android 中为这种特定类型的硬件定义的API。因此，此解决方案不再允许 Android 将所有可能的驱动程序嵌入内核，并禁用动态模块加载内核机制。提供此功能的组件在 Android 中称为硬件抽象层。

此外，这样的架构解决方案允许硬件供应商选择许可证，在其下分发它们的驱动程序。

内核通过启动一个名为 init 的用户空间进程来完成其启动。 此过程负责启动 Android 中的所有其他进程和服务，以及在操作系统中执行一些操作。 例如，如果关键服务在 Android 中停止应答，init 进程可以重新启动它。 该进程根据 init.rc 配置文件执行操作。 工具箱包括基本的二进制文件，在 Android 中提供 shell 工具的功能。

Android 还依赖于一些关键的守护进程。 它在系统启动时启动，并在系统工作时保持它们运行。 例如，rild （ 无线接口层守护进程，负责基带处理器和其他系统之间的通信） ，servicemanager （ 一个守护进程，它包含在 Android 中运行的所有 Binder 服务的索引） ， adbd （ Android Debug Bridge 守护进程，作为主机和目标设备之间的连接管理器）等。

本地用户空间中最后一个组件是本地库。 有两种类型的本地库：来自外部项目的本地库，以及在 Android 自身中开发的本地库。 这些库被动态加载并为 Android 进程提供各种功能。

应用程序框架

Dalvik 是 Android 的基于寄存器的虚拟机。它允许操作系统执行使用 Java语言编写的 Android 应用程序。在构建过程中，Java 类被编译成由 Dalvik VM 解释的 .dex 文件。

为了加速进程初始化过程，Android 利用了一个名为 Zygote 的特定组件。这是一个将所有核心库链接起来的特殊“预热"过程。当新应用程序即将运行时，Android 会从 Zygote 分配一个新进程，并根据已启动的应用程序的规范设置该进程的参数。该解决方案允许操作系统不将链接库复制到新进程中，从而加快应用程序启动操作。

系统服务是 Android 的最重要的部分之一。 Android 提供了许多系统服务，它们提供了基本的移动操作系统功能，供 Android 应用开发人员在其应用中使用。 例如， PackageManagerService 负责管理（ 安装，更新，删除等） 操作系统中的 Android 包。 使用 JNI 接口系统服务可以与本地用户空间层的守护进程，工具箱二进制文件和本地库进行交互。