

基于 SEAndroid 的隐私保护机制研究

温瀚翔 李玉军 侯孟书

(电子科技大学计算机科学与工程学院 成都 611731)

摘要 随着移动应用的迅猛发展, 安卓手机用户群体日益庞大, 而随之不断增加的用户数据也使安卓系统成为恶意攻击者的主要目标。通过对安卓 4.4 系统中加入的 SELinux 机制进行分析研究, 指出了其中对 root 权限进行细化限制的可能性, 并基于此机制提出了一种增强隐私安全的设计, 使得用户的隐私数据即使存在于已获得 root 权限的手机中, 也可以得到有效的保护。

关键词 安卓, SELinux, 强制访问控制, 隐私保护

中图分类号 TP309.2 **文献标识码** A

Research on Privacy Protection Based on SEAndroid

WEN Han-xiang LI Yu-jun HOU Meng-shu

(School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China)

Abstract With the rapid development of mobile application, the number of Android phone users has increased sharply, and the growing users' data have made Android system become the main target of malicious attackers. We analyzed and researched the SELinux added in Android 4.4 system to point out the possibility of refining restrictions on root permissions. Based on the mechanism, we put forward a design which can strengthen privacy in order to protect private data even if the data store in the mobile phone which has obtained root permissions.

Keywords Android, SELinux, Mandatory access control, Privacy protection

1 引言

随着手机操作系统迅速发展, 安卓操作系统由于其开源性好、兼容性强、可定制性强的特点成为手机制造商最常用的操作系统之一。同时, 随着安卓手机功能的日益强大, 其存储的用户信息种类及数据也与日俱增。但由于安卓系统的碎片化以及部分代码开源, 导致个人隐私数据泄露问题频繁出现。因此, 安卓的隐私安全保护机制已成为当前研究的热点。

基于静态代码检测识别技术, 研究人员首先试图通过静态权限增强控制来解决安卓隐私安全问题, 并提出了 PScout^[1]、Kirin^[2]、PEG^[3]、Woodpecker^[4]、Stowaway^[5]、APEFS^[6]、DroidMat^[7] 等解决方案。但这些解决方案对动态行为缺乏监控机制, 难以防止通过加载恶意动态库读取隐私数据的问题。同时, 研究人员也提出了 Apex^[8]、PrimAndroid^[9]、MockDroid^[10]、Quire^[11]、TTPE^[12]、CHEX^[13]、XManDroid^[14] 等基于动态权限分析的保护方案, 通过对关键组件和系统调用进行监控检测, 解决应用运行时的权限控制问题。但动态权限分析保护机制无法有效检测多个应用的相互调用, 对于动态修改运行时代码的程序仍存在无法预测的问题。为了解决以上问题, 研究人员基于 Linux 的 SELinux 机制, 提出了基于强制访问控制的套件体系 SEAndroid^[15]。SEAndroid 对于数据流的标记以及对文件的访问控制卓有成

效, 通过对关键系统调用点的权限检查以及强制访问控制有效地解决了上述问题。之后, YAASE 改进了中间层访问控制机制^[16], 进一步增强了 SEAndroid 的安全性。

SEAndroid 框架是一种有效的为操作系统权限完整性提供保护的解决方案, 但仅对应用自身的用户级数据以及系统调用增加了访问控制的检查, 缺乏敏感用户数据专属保护及检查阻止机制, 对于隐私安全的增强仍有提高的空间。针对这一问题, 本文对安卓 4.4 平台上的 SEAndroid 机制进行分析, 在保持其原有安全目标的前提下, 做出进一步的修改定制, 形成了基于强制访问控制的隐私保护模型, 使得仅具备特定权限的应用才能访问敏感数据, 达到了对上层访问的透明、对文件系统友好的效果。

2 SEAndroid 体系分析

SEAndroid 是 Google 在 Android 4.3 上推出的以 SELinux 为基础的安全机制。SELinux 是由美国国家安全和 RedHat、Tresys 针对 Linux 而设计的安全增强机制。SEAndroid 是将 SELinux 移植到 Android 平台后做了相应扩展的机制。在 Android 4.4 中 SEAndroid 部分功能已经启用, 但在推出时没有对权限控制进行足够完善的测试, 所以为了保证系统稳定性, 仅仅对系统调用以及内核相关部分进行了强制访问控制, 其余部分只在 Android 模拟器中生效。

本文受国家自然科学基金面上项目(61472067), 四川省科技支撑计划(2013GZ006)资助。

温瀚翔(1993—), 男, 主要研究领域为移动安全, E-mail: 610378985@qq.com; 李玉军(1975—), 男, 博士, 副教授, 硕士生导师, 主要研究领域为无线传感器网络与操作系统; 侯孟书(1971—), 男, 博士, 教授, 博士生导师, 主要研究领域为计算机网络。

SELinux 包括 Linux 内核安全模型框架、审计模块,以及带有安全标签的文件系统。在 SELinux 中,每一个进程和文件都有自身所对应的进程上下文 (SContext)或文件上下文 (Type),这些进程和文件作为权限控制的主体或者客体,遵守制定的安全策略。在安全策略的制定上,使用了 SELinux Policy 语言规范。安全策略会直接或间接地作用于 SContext 和 Type,经过权限判断后最终形成对文件或者进程的强制访问控制。在 SELinux 中,授权约定遵循了权限最小化的原则,即不声明则默认没有权限。SELinux 强制访问控制中最小的作用区域为一个域,域是一个或者多个上下文的有限集,域可以扩展合并,具有类集合属性。策略控制的最小粒度对每一个进程的每一次操作都会进行访问控制。往往一条访问策略会给多个域授权。

SEAndroid 支持了 Android 特有的 yaffs2 文件系统。对于 Android 系统采用的基于 C/S 的通信模式的 Binder 机制,SEAndroid 也增加了对 Binder 的访问审计控制。在 SEAndroid 中,后缀为 .te 的文件为存储安全策略的文件,这些文件在安全策略生成时被汇集到 policy.conf 文件中,随后进行权限规则检查,确定无冲突后,最终通过 m4 编译生成二进制的策略文件 sepolicy。如图 1 所示,具体的策略文件簇构成了 sepolicy,同时也作为 SEAndroid 运行时的依赖核心。

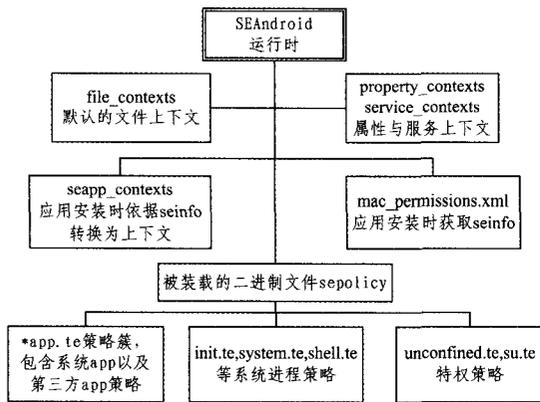


图 1 SEAndroid 运行时组织结构图

图 1 中 file_contexts 文件存放初始化 SEAndroid 时用到的文件上下文,当操作系统启动或者策略被重载时,将根据此文件中的正则规则给相应文件夹和子文件设置初始文件上下文,即给文件标记权限标签。在 SEAndroid 开启强制访问控制时所有不符合安全策略的行为都会被系统拒绝,并保存记录在内核日志中。此外,SEAndroid 还将根据 property_contexts 文件中的策略检查进程对于 Android 系统属性的设置权限。

除此之外,SEAndroid 为了使得中间层更加安全,添加了安装时中间层强制访问控制机制。这一机制主要依据图 1 中的 mac_permissions.xml 和 seapp_contexts 实现控制。控制流程如图 2 所示,在安装应用时,会根据应用中证书的公钥的 base16 值在自身的 mac_permissions.xml 文件中进行匹配,进而给应用分配相应的 seinfo 标签;在安装应用启动运行时,会根据 seinfo 的值以及当前用户、是否为系统服务等参数,在 seapp_contexts 文件中通过最多匹配查找,输出相应的进程域和文件域,之后由 sepolicy 策略库依据相应标签授予对应权限。

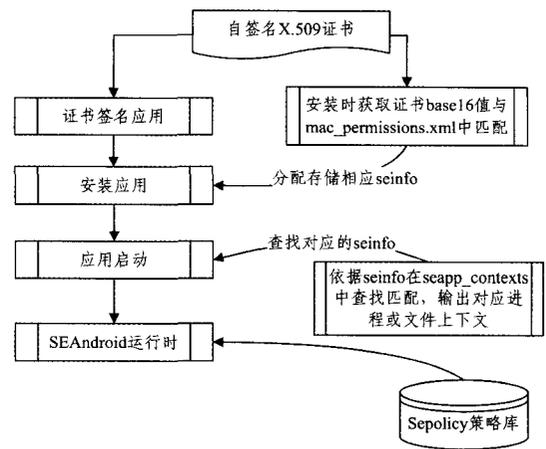


图 2 安装时 MMAC 设计图

SEAndroid 的出现大幅度增强了 Android 系统的安全性;其强制访问控制的细粒度化以及灵活的权限分配提高了系统的健壮性,减轻了恶意程序对系统的危害。但是由于策略设计只保证了 Android 系统的正常运行,对如 root 权限的设计仅保证了 root 用户的正常使用,并未制定出对用户数据保护的安全策略,所以用户在设备中存有的隐私数据信息仍然存在着被窃取或者篡改的风险。针对这一缺陷,利用 SEAndroid 设计出对隐私数据的保护系统很有必要。

3 隐私保护机制设计与实现

3.1 隐私保护机制设计

本系统是基于 SEAndroid 强制访问控制机制的,在文件系统中构造实现一个能供特定应用存取隐私文件的区域。其他用户包括 root 用户均不可访问此区域,从而达到对数据的静态保护,系统设计示意如图 3 所示。该系统设计主要分为两个层次:安全存储策略设计和安装时强制访问控制。安全存储策略设计主要集中于在进程运行时对安全区进行访问控制,而安装时强制访问控制则是在应用安装时依据证书对不同的应用授予各自的进程域与文件域,从而控制对安全存储区的访问。

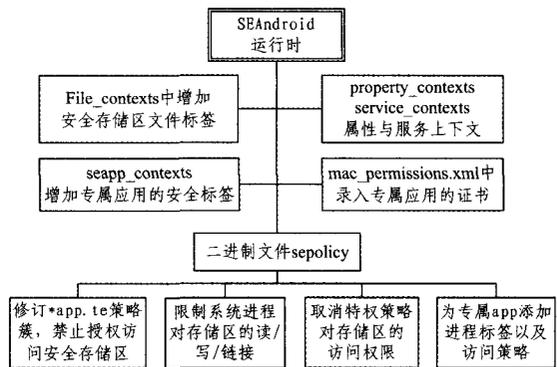


图 3 安全存储策略设计

图 3 中 file_contexts 文件新增分配安全存储区文件标签为 u;object_r;sec_data_file;s0,其中 u 代表用户,object_r 为文件角色,sec_data_file 是隐私文件标签,s0 代表默认的 MLS 级别。通过对应应用以及系统进程所属安全策略的继承关系的分析,最终确定在策略部分主要通过修改 app.te,unconfined.te,为专属应用定制的新安全策略 sec_app.te,保证只有专属

应用的进程域才能访问安全存储区,其他的应用、shell 进程、系统进程等均无权访问。

通过安装时中间层强制访问控制,将签名专属应用对应的证书的公钥信息,以及分配的在 seapp_contexts 中匹配用的标签存入 mac_permissions.xml 中,在 seapp_contexts 中为对应的标签分配专属的进程域以及文件域。进程域和文件域在运行时的访问权限由 sepolicy 策略库分配。

3.2 隐私保护系统实现

在 mac_permissions.xml 分配中间层标签“sec”:

```
<signer signature="@SEPRIVACY" >
  <package name="com.example.secData" >
    <seinfo value="sec" />
  </package>
</signer>
```

在 keys.conf 中指定使用的证书的位置,sec.x509.pem 是通过 openssl 生成的含有特定签名的证书。

```
[@SEPRIVACY]
```

```
ALL;sec.x509.pem
```

在 file_contexts 中增加安全文件上下文:

```
/data/secData(/. * )? u:object_r:sec_data_file:s0
```

在 seapp_contexts 中增加新标签 sec_app:

```
user=_app seinfo=sec domain=sec_app type=app_data_file
```

在 file.te 中关联安全文件上下文:

```
type sec_data_file,file_type,data_file_type;
```

增加 sec_app.te 策略文件,授予其正常第三方应用所具有的权限许可以及访问安全存储区的权限:

```
type sec_app,domain;
app_domain(sec_app)
sec_app_domain(sec_app)
platform_app_domain(sec_app)
net_domain(sec_app)
bluetooth_domain(sec_app)
unconfined_domain(sec_app)
allow sec_app sec_data_file;file_class_set create_file_perms;
allow sec_app sec_data_file;dir create_dir_perms;
```

在 unconfined.te 中限制对安全存储区的访问权限:

```
allow unconfineddomain {fs_type dev_type file_type -sec_data_file};
{dir blk_file lnk_file sock_file fifo_file} ~relabelto;
allow unconfineddomain {fs_type dev_type file_type -sec_data_file};
{chr_file file} ~{entrypoint relabelto};
neverallow{unconfineddomain-sec_app} sec_data_file;dir create_dir_perms;
neverallow{unconfineddomain-sec_app} sec_data_file;file_class_set create_file_perms;
```

由于此策略为 su、shell 以及一些系统进程的父域,因此也级联取消了它们对安全存储区的访问权限。同样取消 app.te 对安全存储区的访问权限也会级联地取消系统应用以及第三方应用的对应权限。

```
neverallow { appdomain -unconfineddomain-sec_app} sec_data_file;
dir create_dir_perms;
neverallow { appdomain -unconfineddomain-sec_app} sec_data_file;
file_class_set create_file_perms;
```

策略制定完之后执行 m4 命令检查策略冲突并编译生成二进制文件 sepolicy,再与 file_contexts,seapp_contexts 一起通过 adb push 放入手机/data/security/current/目录下,通过命令 setprop selinux.reload_policy 1 使策略重载生效。

4 实验效果验证

为了保证测试效果的通用性,使用安卓开源操作系统 AOSP 进行试验并刷入 Nexus5 真机测试,通过 adb shell 连接以 root 访问查看效果。此时 root 权限可以访问任何区域。

```
root@hammerhead:/data # echo test > secData/test
echo test > secData/test
root@hammerhead:/data # cat secData/test
test
```

将隐私安全保护系统刷入 AOSP 中,以 root 权限的 shell 尝试访问安全存储区,secData 文件夹:

```
root@hammerhead:/data # cd secData
sh;cd:/data/secData;Permission denied
```

尝试对子文件进行读写:

```
root@hammerhead:/data # echo 123 > secData/test
sh;can't create secData/test;Permission denied
root@hammerhead:/data # cat secData/test
sh;cat;secData/test;Permission denied
```

可以看到 root 进程尝试访问 secData 文件夹的操作都被禁止,输出内核相关审计记录:

```
root@hammerhead:/data # dmesg | grep avc
<5>[6.617062] type=1400 audit(2918703.190:5):avc:denied {read}
for pid=1 comm="init" name="secData" dev="mmcblk0p28"
ino=122161 scontext=u:r:init;s0 tcontext=u:object_r:sec_data_file;s0 tclass=dir
<5>[6.617187] type=1400 audit(2918703.190:6):avc:denied {open}
for pid=1 comm="init" name="secData" dev="mmcblk0p28"
ino=122161 scontext=u:r:init;s0 tcontext=u:object_r:sec_data_file;s0 tclass=dir
<5>[6.617278] type=1400 audit(2918703.190:7):avc:denied {setattr}
for pid=1 comm="init" name="secData" dev="mmcblk0p28"
ino=122161 scontext=u:r:init;s0 tcontext=u:object_r:sec_data_file;s0 tclass=dir
<5>[183.492806] type=1400 audit(2918879.590:9):avc:denied {getattr}
for pid=1948 comm="sh" path="/data/secData"
dev="mmcblk0p28" ino=122161 scontext=u:r:su;s0 tcontext=u:object_r:sec_data_file;s0 tclass=dir
```

以第一条记录为例,init 进程尝试在文件上下文为 sec_data_file 的 secData 目录下读取文件,由于 init 进程所属的进程上下文 init 没有读取权限,因此请求被拒绝。由后面几条记录也可以看到,root 进程尝试遍历目录等操作均被捕获并禁止。

具有 platform 签名的系统应用尝试访问安全存储区:

```
<5>[48.457168] type=1400 audit(2924904.759:10):avc:denied {search}
for pid=1777 comm="example.testapp" name="secData"
dev="mmcblk0p28" ino=122161 scontext=u:r:platform_app;s0 tcontext=u:object_r:sec_data_file;s0 tclass=dir
```

启动专属应用后查看对应进程域:

```
root@hammerhead:~# ps -Z |grep sec
u:r:sec_app:s0 u0_a53 2221 177 com.example.secData
```

可以看到专属应用已经获得了 sec_app 的进程域上下文。由策略可知,当进程上下文为 sec_app 的进程访问安全存储区时,由于具有访问安全存储区权限,因此允许其进行任意操作。

结束语 通过对 SEAndroid 框架的扩展,在维护操作系统底层安全的同时还有效保护了关键数据的静态存储安全,为隐私安全的保护提供了一个新的思路和方向。在采用 SE-Android 的新版本 Android 操作系统中,此设计具有轻量级、透明化的特性。但是同时由于建立安全存储区时涉及到 init.rc 等 Android 系统文件的修改、不同厂商 Android 系统版本的实现设计不完全相同以及策略实施过程中需要频繁测试编译和运行验证策略的有效性,对 SEAndroid 的定制修改比较复杂繁琐,相对适合于手机生产厂商进行相关定制,从系统级别维护手机的数据存储安全,守护 root 后的系统安全,从而增强对用户隐私的保护。

参考文献

- [1] Aukwy,Zhou Yi-fan,Huang Zhen, et al. PScout: analyzing the Android permission specification [C]//Proc of ACM Conference on Computer and Communications Security. New York: ACM Press,2012:217-228
- [2] Chen K Z, Johnson N, D'silva V, et al. Contextual policy enforcement in Android programs with permission event graphs [C]//Proc of the 20th Annual Network and Distributed System Security Symposium. San Diego:Internet Society,2013:455-464
- [3] Grace M,Zhou Ya-jin,Wang Zhi, et al. Systematic detection of capability leaks in stock Android smartphones[C]//Proc of the 19th Annual Symposium on Network and Distributed System Security. San Diego:Internet Society,2012:235-244
- [4] Di C F,Girardell A,Michahelles F, et al. Detection of malicious applications on Android OS [C]//Proc of the 4th International Conference on Computational Forensics. Berlin:Springer,2011:138-149
- [5] Felt A P,Chin E,Hanna S, et al. Android permissions demystified [C]//Proc of the 18th ACM Conference on Computer and Communications Security. New York: ACM Press,2011:627-638
- [6] Meurer S,Wismül R. APEFS:an infrastructure for permission-based filtering of Android apps[C]//Security and Privacy in Mobile Information and Communication Systems. Berlin:Springer,2012:1-11
- [7] Wu Dong-jie,Mao C H,Wei T E, et al. DroidMat: Android malware detection through manifest and API calls tracing[C]//Proc of the 7th Asia Joint Conference on Information Security. 2012:62-69
- [8] Nauman M,Khan S,Zhang Xin-wen. Apex: extending Android permission model and enforcement with user-defined runtime constraints[C]//Proc of the 5th ACM Symposium on Information, Computer and Communications Security. New York: ACM Press,2010:328-332
- [9] Benats G, Bandara A, Yu Yi-jun, et al. PrimAndroid: privacy policy modelling and analysis for Android applications [C]//Proc of IEEE International Symposium on Policies for Distributed Systems and Networks. 2011:129-132
- [10] Beresford A R,Rice A,Skehin N, et al. MockDroid trading privacy for application functionality on smartphones[C]//Proc of the 12th Workshop on Mobile Computing Systems and Applications. New York: ACM Press,2011:49-54
- [11] Dietz M,Shekhar S,Pisetsky Y, et al. Quire: lightweight provenance for smart phone operating systems[C]//Proc of the 20th USENIX Security Symposium. Berkeley:USENIX Association,2011:232-241
- [12] Bugiel S,Davi L,Dmitrienko A, et al. Towards taming privilege escalation attacks on Android[C]//Proc of the 19th Annual Network & Distributed System Security Symposium. San Diego: Internet Society,2012:18-25
- [13] Lu Long,Li Zhi-chun,Wu Zhen-yu, et al. CHEX: statically vetting Android apps for component hijacking vulnerabilities[C]//Proc of ACM Conference on Computer and Communications Security. New York: ACM Press,2012:229-240
- [14] Bugiel S,Davi L,Dmitrienko A, et al. XManDroid: a new Android evolution to mitigate privilege escalation attacks; TR-2011-04[R]. Darmstadt: Technische Universität Darmstadt,2011
- [15] Smalley S, Craog R. Security enhanced (SE) Android: bringing flexible MAC to Android[C]//Proc of the 20th Annual Network & Distributed System Security Symposium. San Diego: Internet Society,2013:75-84
- [16] Russell G,Crispo B,Fernandes E, et al. YAASE: yet another Android security extension[C]//Proc of the 3rd International Conference on Privacy, Security, Risk and Trust Social Computing. USA: IEEE Press,2011:1033-1040
- [8] L Jian-Hui,H Chang-Jun. A RSSI-Based Localization Algorithm in Smart Space [M]//Intelligent Decision Technologies. Springer Berlin Heidelberg,2011:671-681
- [9] Babic Z,Ljubojevic M,Risojevic V. Indoor RFID localization improved by motion segmentation [C]//2011 7th International Symposium on Image and Signal Processing and Analysis (ISPA). IEEE,2011:271-276
- [10] 高锐,程良伦,苏海武. 三维空间 RFID 定位系统方法及其应用研究[J]. 计算机应用研究,2013,30(11):3336-3338

(上接第 309 页)

- [4] Wu K,Xiao J, Yi Y, et al. Fila: Fine-grained indoor localization [C]//2012 Proceedings IEEE INFOCOM. IEEE,2012:2210-2218
- [5] 高锐,程良伦,胡莘. 一种基于空间分割的无源 RFID 室内定位方法[J]. 计算机应用研究,2012,29(1):184-186
- [6] 龙易,黄际彦,杜江. 基于最小均方误差估计的 RFID 室内定位算法[J]. 南京邮电大学学报(自然科学版),2013,33(6):69-75
- [7] 张金艺,张晶晶,李若涵,等. 流水线型局部加权回归 RFID 室内定位[J]. 应用科学学报,2014,32(2):125-132