Monthly Research

SE for Android Overview

F F R I , Inc
http://www.ffri.jp

Ver 1.00.03

About *SE for Android*

- Security enhancement for Android developed by NSA

- Worked with Android Open Source Project(AOSP)
  - Enforcing for installd, netd, vold and zygote in Android 4.4(KitKat)
  - Permissive for other processes and apps
    - There are labeled, but not enforced
  - In the near future: All android apps enforced by SELinux

Threat model

- Root exploits
  - Linux kernel vulnerability CVE-2012-0056（Mempodipper）
    - Incorrect permission checking with /proc/pid/mem
      - Privilege escalation may cause by code injection
    - ✓Privilege escalation is prevented because SELinux restricts original SELinux contexts
- Incorrect access controls
  - Mobile (LOOK-11-001)
  - Created files without setting umask
    - Information leakage may cause by malicious app
    - ✓SELinux isolates app's resources from other app using SELinux contexts

History of SE for Android

- 2012.01 SE for Android releases
- 2012.03 Samsung collaboration begins
- 2013.04 First device SE ships - Galaxy S4
- 2013.07 First Android releases SE permissive - Android 4.3
- 2013.10 First Android releases SE enforcing(partially) - Android 4.4

Terminology

- Security Enhancements(SE) for Android
  - “Used to describe the overall framework for implementing SELinux mandatory access control (MAC) and Middleware mandatory access control (MMAC) on Android”
- SE Android
  - “The SEAndroid project enhancements are decreasing as more features move into AOSP”
- AOSP
  - “The Android code base distributed by Google”

Terms from http://selinuxproject.org/page/NB_SEforAndroid_1

Details of Security Enhancements(SE) for Android

- Additional kernel space components
  - Implemented security labeling for yaffs2
  - Instrumented Binder for SELinux

- Middleware MAC
  - Install-time MAC
  - Enterprise Ops
  - Intent Firewall

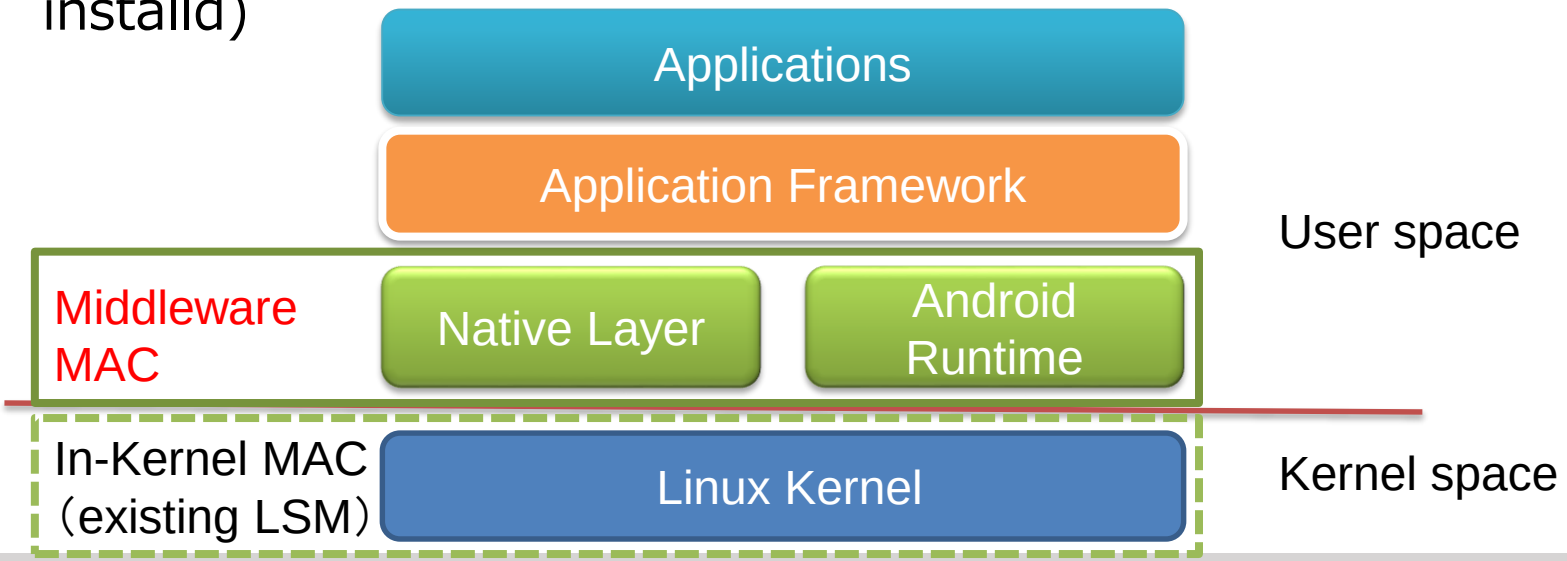
- User space tools
  - Extending Bionic Libc
  - Porting libselinux and policytools

MAC : Mandatory access control

Middleware MAC

- SELinux cannot control user space event
  - In addition, zygote process control model cannot apply exec-based domain transition

- SE for Android integrates access control fundamentals into android middleware(such as zygote, dalvik runtime and installd)



Install-time MAC

- Whitelist/disable app enforced by PackageMangerService
  - policy example: external/sepolicy/mac_permissions.xml
- Linkage to SELinux policy via **seinfo** identifier
  - Installd and zygote uses this linkage information

seapp_contexts

See also AOSP source
sepolicy/seapp_contexts:

```
isSystemServer=true domain=system_server
user=system domain=system_app type=system_app_data_file
user=bluetooth domain=bluetooth type=bluetooth_data_file
user=nfc domain=nfc type=nfc_data_file
user=radio domain=radio type=radio_data_file
user=shared_relro domain=shared_relro
user=shell domain=shell type=shell_data_f
user=_isolated domain=isolated_app
user=_app seinfo=platform domain=platform_app type=app_data_file
user=_app domain=untrusted_app type=app_data_file
```

Identifying platform app using *seinfo*

Labeling 3rd-party app with untrusted_app by default

Enterprise Ops & Intent Firewall（Beta）

- **Enterprise Ops(eops)**
  - Controlling app operations (Extending AppOps)
  - Replaces permission revocation mechanism

- **Intent Firewall**
  - Controlling app interactions
  - Replaces Intent MAC

✓ Both features introduced in Android 4.3
  - Android 4.4 includes its mechanism only (AOSP not contains its policy yet)

```
<?xml version="1.0"?>
<app-ops>
   <debug/>
   <seinfo name="system">
      <op name="CAMERA"/>
   </seinfo>
</app-ops>
```

Example1: The eops policy will stop the camera being used by any system or default app

```
<?xml version="1.0"?>
<rules>
   <service log="true" block="true">
      <not><sender type="system"/></not>
      <intent-filter />
         <component-filter
name="com.se4android.isolatedservice/.DemoIsolatedService"/>
   </service>
</rules>
```

Example2: This will stop any app that is not a system app from running the DemoIsolatedService service

Conclusions

- SE for Android project provides security enhancement mechanism and policy to improve existing android platform security
  - Satisfying extra security requirements for mobile devices usage such as enterprise and government organization
- Android device developers should understand SE for Android functions and policies
  - Even system app is restricted
- App developers should pay attention to SELinux's merged status on AOSP
  - Android 4.4 still grant permissive domains, but 3rd-party apps are restricted by SELinux in the near future

References

- Security Enhancements (SE) for Android™
  http://seandroid.bitbucket.org/
- Security Enhanced (SE) Android: Bringing Flexible MAC to Android, 20th Annual Network and Distributed System Security Symposium (NDSS '13), Feb 2013.
  http://www.internetsociety.org/sites/default/files/Presentation02_4.pdf
- Security Enhancements (SE) for Android, Android Builders Summit 2014, Apr 2014.
  http://events.linuxfoundation.org/sites/events/files/slides/abs2014_seforandroid_smalley.pdf
- The Flask Security Architecture: System Support for Diverse Security Policieshttp://www.nsa.gov/research/_files/publications/flask.pdf
- NB SEforAndroid 1
  http://selinuxproject.org/page/NB_SEforAndroid_1
- NB SEforAndroid 2
  http://selinuxproject.org/page/NB_SEforAndroid_2
- Iintent firewall(unofficial documentation)
  http://www.cis.syr.edu/~wedu/android/IntentFirewall/

Contact Information

E-Mail ： research—feedback@ffri.jp
Twitter： @FFRI_Research