Rockchip Android Root Introduction

ID: RK-SM-YF-E16

Release Version: V1.0.0

Release Date: 2023-01-29

Security Level: □Secret □Internal □Public ■Public

Rockchip Electronics Co., Ltd.

No.18 Building, A District, No.89, software Boulevard Fuzhou, Fujian,PRC

Website: www.rock-chips.com

Customer service Tel: +86-4007-700-590

Customer service Fax: +86-591-83951833

Customer service e-Mail: fae@rock-chips.com

Preface

Overview

This document provides a method to achieve root access on the Android system.

Intended Audience

This document (this guide) is mainly intended for

Technical Support Engineer

Software Development Engineer

Revision History

Version	Author	Date	Change Description
V1.0.0	Wu Liangqing	2023-01-29	Initial version release

Contents

This method is applicable to Android userdebug and user builds.

1. Disable SELinux

system/core

```
diff --git a/init/selinux.cpp b/init/selinux.cpp
index 5a0255acd..787917274 100644
--- a/init/selinux.cpp
+++ b/init/selinux.cpp
@@ -104,6 +104,8 @@ EnforcingStatus StatusFromCmdline() {
 }

 bool IsEnforcing() {
+    return false;
+
     if (ALLOW_PERMISSIVE_SELINUX) {
         return StatusFromCmdline() == SELINUX_ENFORCING;
     }
```

2. Modify su.cpp, comment out group permission check

system/extras/su/su.cpp

```
diff --git a/su/su.cpp b/su/su.cpp
index 1a1ab6bf..af3d2a68 100644
--- a/su/su.cpp
+++ b/su/su.cpp
@@ -80,8 +80,8 @@ void extract_uidgids(const char* uidgids, uid_t* uid, gid_t*
gid, gid_t* gids, i
 }

 int main(int argc, char** argv) {
-    uid_t current_uid = getuid();
-    if (current_uid != AID_ROOT && current_uid != AID_SHELL) error(1, 0, "not
allowed");
+    //uid_t current_uid = getuid();
+    //if (current_uid != AID_ROOT && current_uid != AID_SHELL) error(1, 0, "not
allowed");

     // Handle -h and --help.
     ++argv;
```

3. Grant Default Root Permissions to the su File

system/core/libcutils/fs_config.cpp

```
diff --git a/libcutils/fs_config.cpp b/libcutils/fs_config.cpp
index 5805a4d19..92e93e76f 100644
--- a/libcutils/fs_config.cpp
+++ b/libcutils/fs_config.cpp
@@ -86,7 +86,7 @@ static const struct fs_path_config android_dirs[] = {
     { 00751, AID_ROOT,         AID_SHELL,        0, "system/bin" },
     { 00755, AID_ROOT,         AID_ROOT,         0, "system/etc/ppp" },
     { 00755, AID_ROOT,         AID_SHELL,        0, "system/vendor" },
-    { 00750, AID_ROOT,         AID_SHELL,        0, "system/xbin" },
+    { 00755, AID_ROOT,         AID_SHELL,        0, "system/xbin" },
     { 00751, AID_ROOT,         AID_SHELL,        0, "system/apex/*/bin" },
     { 00751, AID_ROOT,         AID_SHELL,        0, "system_ext/bin" },
     { 00751, AID_ROOT,         AID_SHELL,        0, "system_ext/apex/*/bin" },
@@ -190,7 +190,7 @@ static const struct fs_path_config android_files[] = {
     // the following two files are INTENTIONALLY set-uid, but they
     // are NOT included on user builds.
     { 06755, AID_ROOT,      AID_ROOT,      0, "system/xbin/procmem" },
-    { 04750, AID_ROOT,      AID_SHELL,     0, "system/xbin/su" },
+    { 06755, AID_ROOT,      AID_SHELL,     0, "system/xbin/su" },
```

frameworks/base/core/jni/com_android_internal_os_Zygote.cpp

```
diff --git a/core/jni/com_android_internal_os_Zygote.cpp
b/core/jni/com_android_internal_os_Zygote.cpp
index 9eede83e21e5..694eec2a40ac 100644
--- a/core/jni/com_android_internal_os_Zygote.cpp
+++ b/core/jni/com_android_internal_os_Zygote.cpp
@@ -656,6 +656,7 @@ static void EnableKeepCapabilities(fail_fn_t fail_fn) {
 }

 static void DropCapabilitiesBoundingSet(fail_fn_t fail_fn) {
+/*
   for (int i = 0; prctl(PR_CAPBSET_READ, i, 0, 0, 0) >= 0; i++) {;
     if (prctl(PR_CAPBSET_DROP, i, 0, 0, 0) == -1) {
       if (errno == EINVAL) {
@@ -666,6 +667,7 @@ static void DropCapabilitiesBoundingSet(fail_fn_t fail_fn) {
       }
     }
   }
+  */
 }
```

kernel/security/commoncap.c

```
diff --git a/security/commoncap.c b/security/commoncap.c
index f86557a8e43f6..19124dd6239a1 100644
--- a/security/commoncap.c
+++ b/security/commoncap.c
@@ -1147,12 +1147,12 @@ int cap_task_setnice(struct task_struct *p, int nice)
 static int cap_prctl_drop(unsigned long cap)
 {
        struct cred *new;
-
+/*
```

```
        if (!ns_capable(current_user_ns(), CAP_SETPCAP))
                return -EPERM;
        if (!cap_valid(cap))
                return -EINVAL;
-
+*/
        new = prepare_creds();
        if (!new)
                return -ENOMEM;
```

4. The user version requires integrating 'su' into the system.

```
build/core
```

```
diff --git a/target/product/base_system.mk b/target/product/base_system.mk
index 4569bceff9..5c8eaaa87c 100644
--- a/target/product/base_system.mk
+++ b/target/product/base_system.mk
@@ -273,6 +273,7 @@ PRODUCT_PACKAGES += \
     wificond \
     wifi.rc \
     wm \
+    su \

 # VINTF data for system image
 PRODUCT_PACKAGES += \
@@ -378,7 +379,6 @@ PRODUCT_PACKAGES_DEBUG := \
     ss \
     start_with_lockagent \
     strace \
-    su \
     sanitizer-status \
     tracepath \
     tracepath6 \
```