

密级：指定客户



移动应用加固工具（Android）

使用说明

网易（杭州）网络有限公司

2021 年 02 月

目 录

一、加固方法.....	1
二、配置文件.....	1
2.1 配置 appkey.....	1
2.2 配置签名.....	2
2.3 配置 so 文件名	2
三、常用加固示例.....	2
3.1 仅对 app 加固	2
3.2 app 加固与 so 加固	3
3.3 app 加固，so 加固，签名与文件对齐	3
3.4 打渠道包.....	4
3.5 查看 apk 包是否加固.....	6
四、参数说明.....	6
五、FAQ.....	7

一、加固方法

解压工具包，内容包括：NHPProtect.jar 加固工具， config.ini 为加固的功能配置文件。下面对工具的使用及其配置方法进行介绍。

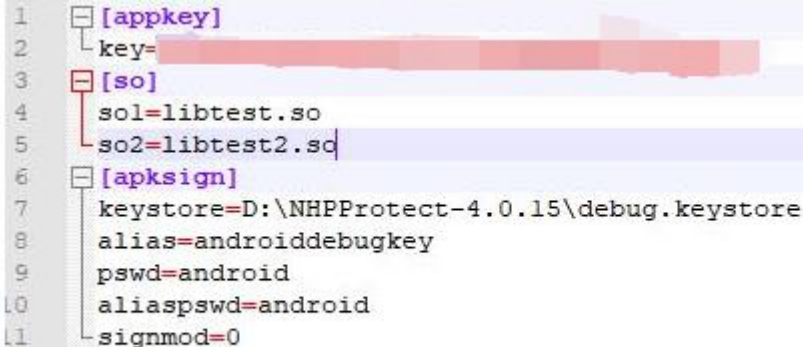
在 config.ini 配置完签名后，将下面的命令中%inputapk%替换为实际的 apk 路径，即可加固。

```
java -jar NHPProtect.jar -yunconfig -fullapk -apksign -zipalign -input %inputapk%
```

（注：如已开通 so 加密功能，需配置 so 名称，具体参见 2.3 节内容。）

二、配置文件

appkey，签名文件，待加固的 so 文件名需要在此设置。



```
1 [appkey]
2 key=
3 [so]
4 so1=libtest.so
5 so2=libtest2.so
6 [apksign]
7 keystore=D:\NHPProtect-4.0.15\debug.keystore
8 alias=androiddebugkey
9 pswd=android
10 aliaspswd=android
11 signmod=0
```

图 1

2.1 配置 appkey

在[appkey]字段填入 key，appkey 可通过易盾后台获取，或是找对接人员索要。

2.2 配置签名

如果加固参数使用-apksign, 需要在 [apksign]字段, 填入 keystore (签名文件路径), alias (签名别名), pswd (签名密码), aliaspswd (签名别名密码), signmod(签名版本), 参考下图

```
[apksign]
keystore=D:\android\jre\bin\my-release-key.keystore
alias=my-release-key
pswd=android
aliaspswd=android
signmod=1
```

其中 signmod:0 表示同时使用 v1 和 v2 签名, 1 表示只使用 v1 签名, 2 表示只使用 v2 签名。默认值为 1, 只使用 v1 签名。(附: v2 签名只支持 Android 7.0 以上系统)

2.3 配置 so 文件名

如有 so 文件需要加固, 需要在配置文件增加[so], 并按照一定的格式填入 so 文件名, 配置方式参考图 1。

三、常用加固示例

3.1 仅对 app 加固

在 config.ini 写入 appkey (参考 2.1), 然后使用如下加固命令即可:

```
java -jar NHPPProtect.jar -yunconfig -fullapk -input D:\test.apk (注意: apk 使用实际文件路径)
```

3.2 app 加固与 so 加固

在 config.ini 写入 appkey, 并写入 so 文件名称(建议仅加固自研 so, 第三方 so 请不要加固), 例如 apk 有两个 so, 分别为 libtest1.so 和 libtest2.so, config.ini 的配置如下:

```
[appkey]
key=此处输入您的key

[so]
so1=libtest1.so
so2=libtest2.so
```

加固命令:

```
java -jar NHPProtect.jar -yunconfig -input D:\test.apk
```

 (注意: apk 使用实际文件路径)

3.3 app 加固, so 加固, 签名与文件对齐

在 config.ini 写入 appkey, 写入 so 文件名称, 并写入签名文件信息。配置如下:

```
[appkey]
key=此处输入您的key

[apksign]
keystore=签名文件路径
alias=签名别名
pswd=签名密码
aliaspswd=别名密码

[so]
so1=libtest1.so
so2=libtest2.so
```

加固命令:

```
java -jar NHPProtect.jar -yunconfig -apksign -zipalign -input D:\test.apk
```

 (注意: apk 使用实际文件路径)

加固成功后，生成的 apk 文件路径为 D:\test_encrypted.apk

3.4 打渠道包

在 config.ini 写入渠道配置文件路径(相对路径或绝对路径皆可)。配置如下：



渠道配置文件示例如下：



说明：每一行代表一个渠道信息，每个渠道信息的内容用 “|” 隔开。其中：

- 第一项：渠道编号(value)，必填。对应<meta-data>标签的 value 字段
- 第二项：渠道名称(name)，可选。不填，则默认为 CHANNEL。对应<meta-data>标签的 name 字段
- 第三项：市场名称，用于标识输出 app，可选。不填，则默认为渠道编号 (value)。仅用作渠道包的识别名称 如上渠道配置示例文件的配置信息解读如下：

渠道配置信息	value	Name	市场名称
360 CHANNEL 奇虎	360	CHANNEL	奇虎
baidu CHANNEL_BAIDU	baidu	CHANNEL_BAIDU	默认为 baidu
google	google	默认为 CHANNEL	默认为 google

加固命令：

`java -jar NHPPProtect.jar -yunconfig -channel -input D:\test.apk`（注意：apk 使用实际文件路径）运行上述命令，会在加固后 apk 所在目录下，生成 channel 目录，目录包含配置文件中配置的渠道包，效果如下所示：

```

</uses-feature>
<uses-feature
    android:name="android.software.vr.high_performance"
    android:required="false"
>
</uses-feature>
<application
    android:theme="@7F040001"
    android:label="@7F030017"
    android:icon="@7F020001"
    android:allowBackup="true"
    android:banner="@7F020000"
    android:isGame="true"
>
    <meta-data
        android:name="CHANNEL_360"
        android:value="m360"
    >
    </meta-data>
    <activity
        android:label="@7F030017"
        android:name="com.unity3d.player.UnityPlayerActivity"
        android:launchMode="2"
        android:screenOrientation="1"
        android:configChanges="0x40003FFF"
    >

```

3.5 查看 apk 包是否加固

可以通过以下加固命令查看当前输入的 apk 包是否已经被易盾加固过。

加固命令：

```
java -jar NHPProtect.jar -htpinfo -input D:\test.apk (注意：apk 使用实际文件路径)
```

四、参数说明

将下面的命令中%inputapk%替换为实际的 apk 路径, 即可加固。 java -jar NHPProtect.jar
-yunconfig -fullapk -apksign -zipalign -input %inputapk%

表 1 加固参数说明

加固参数	说明
-yunconfig	加固工具会根据 appkey 自动配置对应的加固参数。
-fullapk 【必选】	非游戏类 app 加固必填, 会将 apk 全包传至后台进行加固。
-apksign 【可选】	对加固后的 apk 文件进行签名, 签名文件的配置参见下节的文件配置。
-zipalign 【可选】	对加固后的 apk 文件进行对齐, 通常与 apksign 一起使用, 会在签名后进行对齐操作。
-channel 【可选】	对加固后的 apk 文件打渠道包, 会根据配置的渠道配置路径在加固包目录下生成 channel 文件夹, 并生成对应的渠道包。
-htpinfo	输出当前输入的 apk 包是否加固的信息。
-input	参数后面跟待加固的 apk 文件路径。
%inputapk%	待加固的 apk 文件路径。

五、FAQ

1. 工具提示 check appkey failed!

`appkey` 不正确，或者 `app` 包名与申请时提供的不一致。

2. 加固后 apk 安装失败

`app` 可能未签名。

3. 加固后 app 启动后崩溃

`app` 签名文件与原包可能一致。如果是系统 `app`，请将加固后的 `lib` 下 `so` 文件拷贝到系统 `lib`。

4. 加固提示 Align New Apk failed!

此问题常见于 `linux` 和 `unix` 系统，请查看 `tools` 目录下的 `zipalign_linux` 或 `zipalign_mac` 是否有可执行权限。如果没有权限，可通过 `chmod a+x` `zipalign_linux` 或 `xattr -c -r %加固工具目录%` 增加可执行权限。

5. 其它加固失败的问题

日志文件位于 `jar` 包客户端同一目录下的 `Log` 目录内，可以将此日志发给我方，由我方工作人员为您解决。