蚂蚁科技

移动应用安全加固
使用指南

文档版本：20250731

蚂蚁集团
ANT GROUP

法律声明

蚂蚁集团版权所有©2022，并保留一切权利。

未经蚂蚁集团事先书面许可，任何单位、公司或个人不得擅自摘抄、翻译、复制本文档内容的部分或全部，不得以任何方式或途径进行传播和宣传。

商标声明

蚂蚁集团及其他蚂蚁集团相关的商标均为蚂蚁集团所有。本文档涉及的第三方的注册商标，依法由权利人所有。

免责声明

由于产品版本升级、调整或其他原因，本文档内容有可能变更。蚂蚁集团保留在没有任何通知或者提示下对本文档的内容进行修改的权利，并在蚂蚁集团授权通道中不时发布更新后的用户文档。您应当实时关注用户文档的版本变更并通过蚂蚁集团授权渠道下载、获取最新版的用户文档。如因文档使用不当造成的直接或间接损失，本公司不承担任何责任。

通用约定

格式	说明	样例
危险	该类警示信息将导致系统重大变更甚至故障，或者导致人身伤害等结果。	危险 重置操作将丢失用户配置数据。
警告	该类警示信息可能会导致系统重大变更甚至故障，或者导致人身伤害等结果。	警告 重启操作将导致业务中断，恢复业务时间约十分钟。
注意	用于警示信息、补充说明等，是用户必须了解的内容。	注意 权重设置为0，该服务器不会再接受新请求。
说明	用于补充说明、最佳实践、窍门等，不是用户必须了解的内容。	说明 您也可以通过按Ctrl+A选中全部文件。
>	多级菜单递进。	**单击设置> 网络> 设置网络类型。**
粗体	表示按键、菜单、页面名称等UI元素。	**在结果确认**页面，单击**确定**。
`Courier字体`	命令或代码。	执行 `cd /d C:/window` 命令，进入Windows系统文件夹。
斜体	表示参数、变量。	`bae log list --instanceid` *Instance_ID*
[] 或者 [a|b]	表示可选项，至多选择一个。	`ipconfig [-all|-t]`
{} 或者 {a|b}	表示必选项，至多选择一个。	`switch {active|stand}`

目录

1.产品简介

移动应用安全加固（Mobile Security Armor，简称 MSA）为移动应用（下文简称 App）提供稳定、简单、有效的安全保护，提高 App 的整体安全水平，力保应用不被逆向破解。

产品背景

- **Android 应用安全加固**

  由于 Android 系统本身的开源特性，使应用极易遭到盗版侵袭、反编译破解等攻击，严重影响应用的数据与隐私安全。mPaaS 移动应用安全加固对 APK 或 AAB 包进行加固并对加固后的 APK/AAB 包进行兼容性测试和功能回归测试，可最大限度保障应用不被破解。

- **iOS 应用安全加固**

  随着破解、分析技术的发展，苹果自身包括 ipa 加密在内的多种安全防护措施已无法满足 iOS 应用安全要求。mPaaS 移动应用安全加固采用安全编译器的方式对核心代码进行加固，能够极大提高逆向分析的难度，从而有效防护破解和攻击行为。

- **H5 应用安全加固**

  为满足移动安全的监管要求以及提升自身的安全能力，安全加固成为 H5 应用的必然选择。mPaaS 移动应用安全加固通过抹去原来的运行流程，将函数名称、变量名称进行混淆，从而使处理后的 JavaScript 代码难以阅读，防止 H5 应用被破解盗用，保障 H5 开发者的合法权益。

产品优势

- **操作简单，开箱即用**

  Android 通过上传 APK 进行加固；iOS 通过 Xcode 编译器进行加固；H5 通过上传 JS 文件进行加固。

- **高稳定性与兼容性**

  移动应用安全加固依赖于阿里云集团的移动安全加固技术，经历了淘系等上亿级别业务锤炼，兼顾安全性与兼容性，实现极低崩溃率。

  不仅支持 ARM、AARCH64、X86、X64，还支持 Android 4.2 及其以上全线系统版本。

  iOS 安全编译器支持多语言，性能稳定。

- **Java2C，提升安全防护级别**

  字节码被转换为 native 二进制码以提高代码破解难度；代码被编译成由 JNI 调用的 so 文件使攻击者无法实施 Java 逆向分析技术。

- **企业级能力支持**

  提供 OpenAPI 能力，方便对接客户系统（如 Jenkins），提升自动化效率；移动应用安全加固可与热修复功能同时使用，支持的主流热修复能力包括：mPaaS 热修复、阿里热修复、腾讯 Tinker 热修复。

功能特性

以下列出 Android 应用安全加固、iOS 应用安全加固以及 H5 应用安全加固支持的加固能力，有关加固能力具体说明，请参见 基础术语。

- Android 应用安全加固提供的加固能力包括：APK/AAB 包加固、类安全加固。
- iOS 应用安全加固提供的加固能力包括：常量加密、指令替换、控制流平坦化、分支伪造、花指令及坏指令、调用图混淆、符号加密、指针加密。
- H5 应用安全加固提供的加固能力包括：表达式置换、常量字符串加密、代码压缩、对象键名（对象域名）替换、反格式化、防调试、函数变量名混淆、JS 域名绑定、禁止控制台输出、控制流平坦化、虚假控制流、虚拟化保护（VMP）。

2.基础术语

本术语表按拼音首字母顺序对术语进行排序。

A

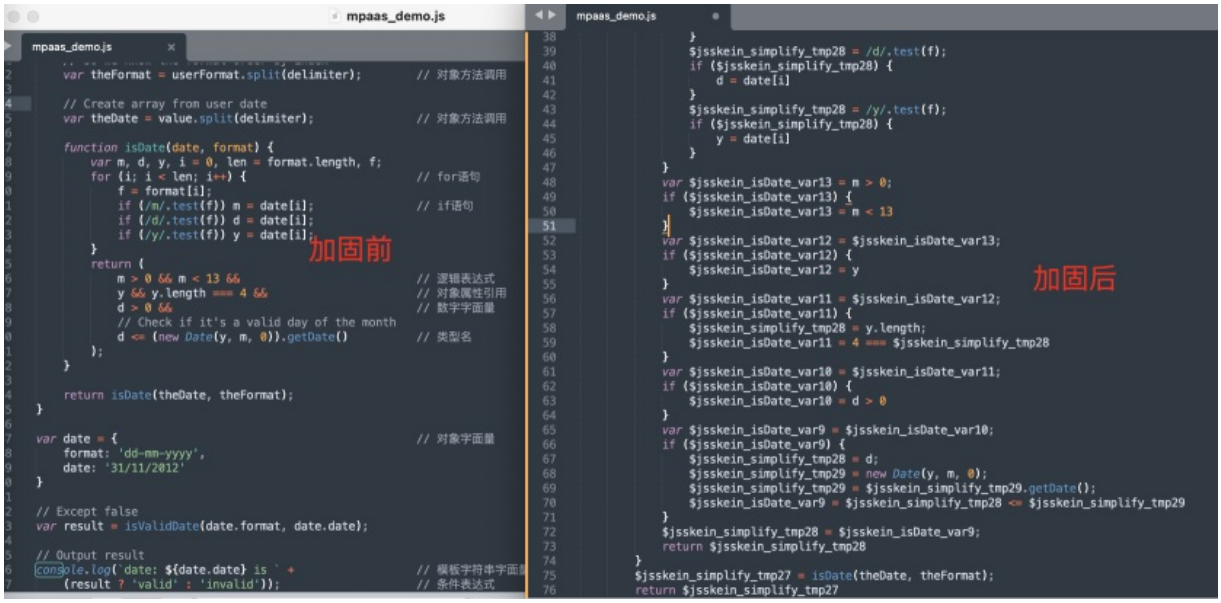
APK/AAB 包加固（Android）

对 APK/AAB 包，整体进行安全保护，提供 APK/AAB 包防反编译保护、DEX 文件整体加壳保护、DEX 文件防篡改保护、防白盒攻击、壳加密算法保护、防调试保护、防内存篡改保护、防 Hook 保护、防模拟器保护、APK/AAB 包防重打包保护、防内存 dump 保护。

B

表达式置换（H5）

对 JavaScript 中的二元表达式等转换成等价函数调用形式，指的是 a + b 这类简单表达式，替换成等价的复杂的表达式，比如 a + (-b)，增大破解者分析难度。



C

常量加密（iOS）

常量加密功能支持对各种类型的数组型常量进行编译期加密。其目的在于隐藏显示字符串如 log 信息以降低信息泄漏风险，及隐藏静态常量数组内容如 AES-SBOX 以提高通用算法静态特征识别难度。

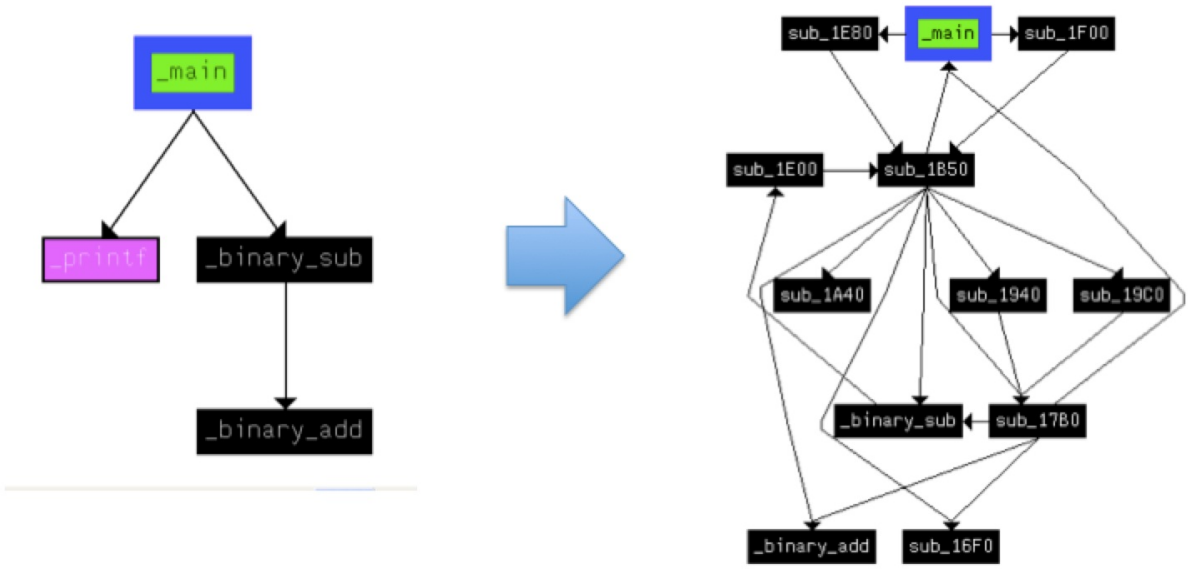
D

代码压缩（H5）

去除 JavaScript 代码中不必要的空格、换行等内容，或把一些可能公用的代码进行处理实现共享，最后输出的结果都压缩为几行内容，降低代码可读性。

调用图混淆（iOS）

调用图指函数间的交叉调用关系，是重要的程序宏观结构描述指标。与过程内针对控制流图的混淆技术相比，调用图混淆模块是一种模块级的、函数间引用关系混淆技术。调用图转化模块通过对源程序所有调用指令的转化处理，可以从宏观层面破坏源程序结构，消除源程序的模块化设计特征。



对象键名替换（H5）

将对象的属性名进行转换，隐藏代码之间的调用关系。

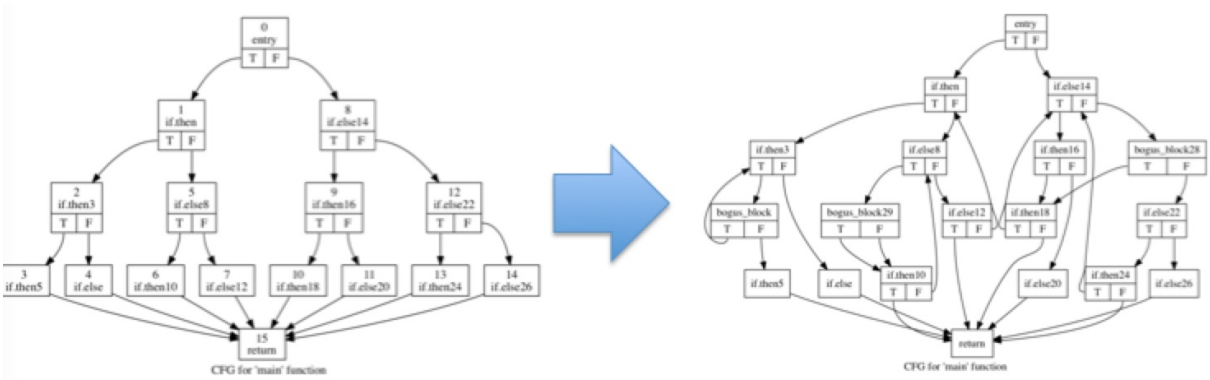
F

反格式化（H5）

强制将代码以单行形式显示，如果对代码进行格式化或者重命名，该段代码将无法执行。

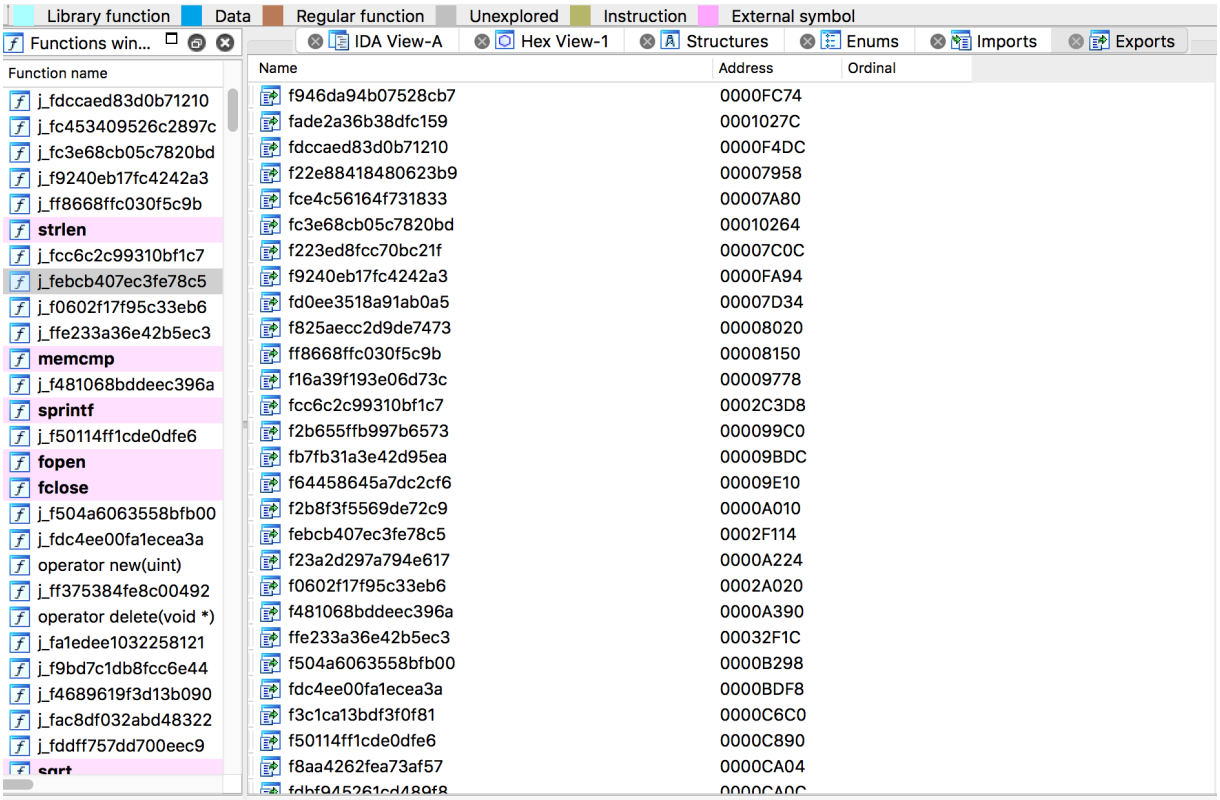
分支伪造（iOS）

分支伪造模块是安全编译器内建的一个基于不透明谓词的控制流混淆方法。结合控制流可达性分析结果及随机不透明谓词构造器，分支伪造模块能以较低的成本（性能影响小，文件膨胀少）达到控制流改造的目的。



符号信息加密（iOS）

符号信息即编译后二进制文件保留的函数名、全局变量名等。符号信息加密，即根据指定规则对指定的、符合特征的函数名、全局变量名等进行加密处理。



H

花指令及坏指令插入（iOS）

花指令指的是可执行但又能干扰分析的一些指令，坏指令指的是不可执行的非法指令。花指令及坏指令插入模块用于对抗 IDA 等反汇编器、反编译器的静态指令流分析能力。基于控制流可达性分析结果，此模块会在程序中随机插入用于干扰分析的花指令和用于阻止分析的坏指令。

```
f797e8ee0476bb108:08003EEE                aaa
f797e8ee0476bb108:08003EEF                push    esi
f797e8ee0476bb108:08003EF0                test    eax, 8B5CF260h
f797e8ee0476bb108:08003EF5
f797e8ee0476bb108:08003EF5 loc_8003EF5:                           ; CODE XREF: .text.f797e8ee0
f797e8ee0476bb108:08003EF5                jmp     loc_80046F2
f797e8ee0476bb108:08003EF5 ; ---------------------------------------------------------------------
f797e8ee0476bb108:08003EFA                dw 868Bh
f797e8ee0476bb108:08003EFC                dd 144h, 8918C083h
f797e8ee0476bb108:08003F04                db 86h
f797e8ee0476bb108:08003F05 ; ---------------------------------------------------------------------
f797e8ee0476bb108:08003F05
f797e8ee0476bb108:08003F05 loc_8003F05:                           ; CODE XREF: .text.f797e8ee0
f797e8ee0476bb108:08003F05                jo      short near ptr byte_8003F0D
f797e8ee0476bb108:08003F05 ; ---------------------------------------------------------------------
f797e8ee0476bb108:08003F07                db 0
f797e8ee0476bb108:08003F08                dd 44868B00h
f797e8ee0476bb108:08003F0C                db 1
f797e8ee0476bb108:08003F0D byte_8003F0D   db 2 dup(0), 8Bh        ; CODE XREF: .text.f797e8ee0
f797e8ee0476bb108:08003F10                dd 6708Eh, 7C8E900h
f797e8ee0476bb108:08003F18                db 2 dup(0)
f797e8ee0476bb108:08003F1A ; ---------------------------------------------------------------------
f797e8ee0476bb108:08003F1A
f797e8ee0476bb108:08003F1A loc_8003F1A:                           ; CODE XREF: .text.f797e8ee0
f797e8ee0476bb108:08003F1A                mov     bl, 80h
f797e8ee0476bb108:08003F1C                add     bl, 1
f797e8ee0476bb108:08003F1F                jo      short near ptr dword_8003F3C
f797e8ee0476bb108:08003F21                jno     short loc_8003F29
f797e8ee0476bb108:08003F23                popf
f797e8ee0476bb108:08003F24                push    8B48F0F0h
f797e8ee0476bb108:08003F29
f797e8ee0476bb108:08003F29 loc_8003F29:                           ; CODE XREF: .text.f797e8ee0
f797e8ee0476bb108:08003F29                jmp     loc_80046F2
f797e8ee0476bb108:08003F29 ; ---------------------------------------------------------------------
f797e8ee0476bb108:08003F2E                dw 80B0h
```

K

控制流平坦化（H5）

控制流平坦化是一种过程内控制流处理技术，能够打乱原有代码执行流程及函数调用关系，使代码逻辑变得混乱无序。旨在消除语义块间的前驱后继关系，提高算法理解成本。

安全编译器内建多种平坦化实现，在处理目标程序时可自动选择；同时安全编译器改进了多处细节处理方法，对基于符号执行技术的控制流分析方法也具备较强的对抗能力。

L

类安全加固（Android）

对 Java 代码进行混淆，隐藏真实运行流程，防止 jadx-gui、jeb 工具的反编译，使加固后的代码难以被人工直接阅读。

Z

指令替换（iOS）

指令替换模块用于等价替换或展开原始代码中的简易二元运算，如四则运算、布尔运算等。替换后的计算过程与代码原意相似度较低且十分繁琐，违背开发人员直觉，难以抽象化理解。指令替换模块内建了数十种替换规则，在混淆过程中随机选用，保证了后端生成的机器指令的随机性和多样性。

指针加密（iOS）

指针加密模块用于消除代码段与数据段间的显式引用关系。

3.计费说明

Android 计费说明

安全加固服务采取预付费模式，需要先购买服务才能使用。对于已经开通了移动开发平台产品的用户，享有时长为七天的免费安全加固服务，自上传 APK/AAB 包时开始计时。七天免费期到期后，系统会提示服务已到期。如需继续使用，则需要付费 购买安全加固服务。

关于安全加固服务的具体定价信息，请参见 预付费模式。

一个安全加固服务只能对一个应用进行加固，安全加固服务会通过 APK/AAB 的包名与应用进行绑定。在上传 APK/AAB 时，安全加固服务会对上传的 APK 或 AAB 的包名进行校验。

- 在上传应用时，如果应用没有绑定过安全加固服务，且上传时有可用的安全加固服务，会将包名绑定到一个安全加固服务。绑定过程在后台自动进行，上传过程不受影响。
- 在上传应用时，如果应用没有绑定过安全加固服务，且上传时没有可用的安全加固服务，系统将会提示购买安全加固服务。
- 如果 APK 或 AAB 的包名绑定过安全加固服务，且绑定的安全加固服务未到期，上传会顺利进行。在服务剩余时长还有三十天时，系统将会提示服务剩余时间。
- 如果 APK 或 AAB 的包名绑定过安全加固服务，且绑定的安全加固服务已到期，系统将会提示服务已到期，需要购买新的安全加固服务。

iOS 计费说明

关于 iOS 的计费问题，欢迎搜索群号 33417739 加入钉钉群进行咨询交流。

H5 计费说明

关于 H5 的计费问题，欢迎搜索群号 33417739 加入钉钉群进行咨询交流。

> **说明**
>
> 由于应用安全加固服务采用的是预付费模式，目前不支持对已购买的加固服务进行续费。请在服务即将到期时重新购买服务并绑定到应用。

4.Android 应用安全加固新工具链（公测）

4.1. 快速开始

本文将引导您使用 Android 安全加固对应用进行快速加固，并获取安全加固包。

前置条件

- 已准备好待防护的 APK，APK 应未加固且 APK 大小 ≤ 300 MB。
- 已购买移动安全加固服务或处于七天免费试用期内。

操作步骤

安全加固新工具链的操作步骤如下：

1. 登录 mPaaS 控制台，并选择目标应用。
2. 单击左侧导航栏中的 **安全服务** > **应用安全加固** > **Android 应用安全加固**。
3. 单击 **创建安全加固**，进入 **上传待加固应用** 页面。
4. 单击 **上传应用**，上传 APK 文件。
5. 上传成功后，页面自动跳转至 **确认安全加固信息** 页面，确认应用信息以及加固信息。
6. 开启 **是否选择新的工具链进行加固** 开关。
7. 根据需要选择 **加固能力**（运行时检测保护）。
8. （可选）在 **添加需要安全保护的类** 栏下选择需要加固的类，选择重要的类即可。
9. （可选）在 **请选择要防护的 So 文件** 栏下选择需要防护的 So 文件。
0. （可选）在 **请选择要防护的 Assets 文件** 栏下选择需要防护的 Assets 文件。
1. 单击 **确认加固** 进行加固操作。
2. 返回 **应用安全加固** 页面，页面上新增相应加固任务的卡片。在卡片中查看相应任务的加固进度。
   - **加固中**：表示加固任务进行中。
   - **已加固**：表示加固任务已完成。
   - **加固失败**：表示加固任务失败。
3. 待卡片中显示 **已加固** 状态时，单击下载加固包图标（ ），即可下载安全加固包（即加固后的 APK 或 AAB 文件）。

> **说明**
>
> 加固后的安装包是没有签名信息的，您需要对下载的加固包重新签名，然后将重新签名后的加固包发布到对应的应用市场。

后续步骤

加固后，请务必检查关键组件功能是否正常，如升级组件、热修复组件。如果安装包在加固后功能异常，请提交工单 或联系 mPaaS 技术支持人员。

> **说明**
> 如果有更多接入相关问题，欢迎搜索群号 33417739 加入钉钉群进行咨询交流。

4.2. 使用指南

安全加固新工具链的使用详情，请参考 Android 移动安全加固使用指南。

> **说明**
> 目前 ABB 文件不支持 **添加需要安全保护的类** 以及其他文件资源保护。

4.3. 查看安全加固日志

若在运行时保护检测到对应风险，安全加固后 App 会退出，退出时会打印如下日志：

```
ashield process runtime info key ---> value
```

key-value 说明

功能	key	value
签名校验	a0	大于 0 的值
防 hook	a1	大于 0 的值
防调试	a2	大于 0 的值
防模拟器	a3	大于 0 的值
防 root	a4	大于 0 的值
防注入	a5	大于 0 的值
防多开	a6	大于 0 的值

5.Android 应用安全加固

5.1. 使用说明

安全加固对加固的 APK/AAB 有以下要求，使用前请阅读以下使用说明，以获得更好的体验。

- 请确认 Provider 的 onCreate 的内容能够多次执行。如果在 onCreate 中有相关的逻辑，请确保它能被执行两次或以上。例如在 Provider 的 onCreate 中进行单例的初始化，则需要判断该实例是否已经被初始化过。
- 不支持 x86、mips 架构，如有配置相关架构，请移除之后重新打包进行加固。
- 目前 minSdkVersion 不支持 24 及以上，minSdkVersion 推荐设置为 23 以下，当 minSdkVersion < 23 时，安全加固默认会对 APK 中的 nativeLibraries 进行压缩存储。如果您需要设置 minSdkVersion = 23，可以从以下两种方式中选择其一进行处理：
  - 在 `application` 这个节点属性中增加 `android:extractNativeLibs="true"` 。
  - 对加固后的 APK 或 AAB 重新进行打包，依据您自己的规则设置是否需要压缩。

> **重要**
>
> 如果要对应用中的 Assets 文件进行加固，则必须确保 minSdkVersion ≥ 21，即 Android 版本不低于 5.0。

- 如果项目中有接入 Arouter，请使用 Gradle 插件实现路由表的自动加载。

```
apply plugin: 'com.alibaba.arouter'//需要在工程的 app module 的 build.gradle 中进行配置

buildscript {
 repositories {
 mavenCentral()
 }

 dependencies {
 classpath "com.alibaba:arouter-register:?"
 }
}
```

通过 ARouter 提供的注册插件进行路由表的自动加载（power by AutoRegister），默认通过扫描 dex 的方式进行加载。通过 Gradle 插件进行自动注册可以缩短初始化时间，解决应用加固导致无法直接访问 dex 文件造成的初始化失败问题。

> **重要**
>
> - 该插件必须搭配 API 1.3.0 以上版本使用。
> - 更多内容请参考 **ARouter** 使用说明。

5.2. 快速开始

本节引导您如何使用 Android 安全加固，对应用进行快速加固并获取安全加固包。

前置条件

- 已准备好待防护的 APK 或 AAB（APK/AAB 应未加固，且大小 ≤ 300 MB）。
- 已购买移动安全加固服务或处于七天免费试用期内。

操作步骤

操作方法如下：

1. 登录 mPaaS 控制台，并选择目标应用。
2. 单击左侧导航栏中的 **安全服务** > **应用安全加固** > **Android 应用安全加固**，进入 **Android 应用安全加固** 页面。
3. 单击 **创建安全加固**，进入 **上传待加固应用** 页面。
4. 单击 **上传应用**，上传 APK 或 AAB 文件。
5. 上传成功后，页面自动跳转至 **确认安全加固信息** 页面，确认应用信息以及加固信息。
6. （可选）在 **添加需要安全保护的类** 栏下选择需要加固的类，选择重要的类即可。
7. （可选）在 **请选择要防护的 So 文件** 栏下选择需要防护的 So 文件。
8. （可选）在 **请选择要防护的 Assets 文件** 栏下选择需要防护的 Assets 文件。
9. 单击 **确认加固** 进行加固。
0. 返回 **应用安全加固** 页面，页面上新增相应加固任务的卡片。在卡片中查看相应任务的加固进度。
   - **加固中**：表示加固任务进行中。
   - **已加固**：表示加固任务已完成。
   - **加固失败**：表示加固任务失败。
1. 待卡片中显示 **已加固** 状态时，单击 **下载**，即可下载安全加固包（即加固后的 APK 或 AAB 文件）。

> **说明**
>
> 加固后的安装包是没有签名信息的，您需要对下载的加固包重新签名，然后将重新签名后的加固包发布到对应的应用市场。

后续步骤

加固后，请务必检查关键组件功能是否正常，如升级组件、热修复组件。如果安装包在加固后功能异常，欢迎搜索群号 33417739 加入钉钉群进行咨询交流。

5.3. 使用指南

5.3.1. 创建安全加固

应用的安全加固指对应用整体以及核心类进行加固，本节引导您完成创建加固任务的完整流程。

移动应用安全加固支持加固的对象包括：

- **APK/AAB 整体**：对 APK/AAB 整体进行安全保护，提供 APK/AAB 防反编译保护、DEX 文件整体加壳保护、DEX 文件防篡改保护、防白盒攻击、壳加密算法保护、防调试保护、防内存篡改保护、防 Hook 保护、防模拟器保护、APK/AAB 防重打包保护、防内存 dump 保护。
- **核心类**：对 Java 代码进行混淆，隐藏真实运行流程，防止 jadx-gui、jeb 工具的反编译，使加固后的代码难以被人工直接阅读。
- **So 文件**：对 So 文件进行加密防护，以增加破解 So 文件的难度和成本。
- **Assets 文件**：对 Assets 资源文件进行加密防护，使其满足监管要求。

> **说明**
>
> 对 APK/AAB 整体的加固是必需的，核心类、So 文件和 Assets 文件的加固是可选的，对 APK 可以根据需求加固项。

前置条件

在开始本任务之前，您需要先准备好待加固的应用，要求如下：

- 文件格式须为 `.apk` 或 `.aab` 。
- 应用须未加固，移动应用安全加固不支持对已加固的安装包进行重复加固。
- APK/AAB 包已有签名。在加固过程中会进行 APK/AAB 的防二次打包处理，因此上传的应用包需要有签名。
- 如果要对应用中的 Assets 文件进行加固，则必须确保 minSdkVersion ≥ 21，即 Android 版本不低于 5.0。
- APK/AAB 大小应 ≤ 300 MB。

> **重要**
>
> 正式下单购买后的首次加固会自动绑定使用的包名，后续只能加固使用此包名的 APK，且绑定后不支持更换包名。试用期间没有此规则限制。

操作步骤

创建加固任务的操作方法如下。

1. 进入 mPaaS 控制台，在应用列表中选择目标应用。
2. 单击左侧导航栏中的 **移动应用安全** > **应用安全加固**，进入 **应用安全加固** 页面。
3. 单击页面上的 **创建安全加固**，进入 **上传待加固应用** 页面。
4. 单击 **上传应用**，上传待加固的安装包。上传过程中，单击页面上的 **取消上传**，可取消上传，**上传待加固应用** 页面将回到初始状态（即未执行上传操作时的状态）。

   > **说明**
   >
   > 当上传的 APK/AAB 不符合要求时，上传将会失败，此时，单击 **重新上传** 后，**上传待加固应用** 页面将回到初始状态。

5. 上传成功后，页面跳转至 **确认安全加固信息** 页面，在该页面上，您需要完成以下操作：
   - **确认应用信息**：在 **应用信息** 栏下查看应用信息。
     - 应用名称
     - 应用包名
     - 应用版本
     - 应用大小
   - **确认加固信息**：在 **加固信息** 栏下，查看对 APK/AAB 整体提供的加固服务。
     - 加壳保护
     - AndroidManifest 文件防篡改保护
     - 签名文件保护
     - 防调试保护

- 防原生应用调试
- 防内存 dump 保护
- 防模拟器运行保护
- 防 Root 设备运行保护
- 防内存数据读取保护
- 防内存数据修改保护
- 防 Hook 攻击保护
- 防内存代码注入保护

- **选择加壳模式**：默认选择 **快速模式**。
  - **快速模式**：经该模式加壳后的应用启动速度较兼容模式加固后的应用快，但是在某些 Android 机型中可能会出现 Crash。
  - **兼容模式**：经该模式加壳后的应用启动速度较快速模式加固后的应用慢，但是兼容性更高，加壳后的应用在运行过程中一般不会出现异常。

    > **说明**
    >
    > 推荐使用 **兼容模式** 对应用进行加壳。

- **添加需要安全保护的类**（可选）：选择需要加固的类，操作方法如下。
  a. （可选）输入类名关键字，单击 **搜索**，搜索目标类。建议输入比较完整的类名进行搜索，当搜索结果超过 1000 个时，平台将无法展示搜索结果，此时，您需要输入完整的类名重新进行搜索。
  b. 单击与目标类对应的复选框，选择目标类。支持多选，最多支持 300 个类。

    > **说明**
    >
    > 已选择的类名会出现在搜索框下方，单击 ×，可取消选择对应类名。

- **请选择要防护的 So 文件**：选择需要加固的 So 文件，操作方法如下。
  a. 输入 So 文件名称中的关键字，单击 **搜索**，搜索目标文件。
  b. 单击待加固 So 文件前的复选框，可以选择一个或多个目标 So 文件。

    > **重要**
    >
    > 在选择待加固的 So 文件时不建议选择第三方的 So 文件进行加固，因为加固第三方的 So 文件以提升应用安全的意义不大，且容易产生兼容性问题

- **请选择要防护的 Assets 文件**：选择需要加固的 Assets 文件，操作方法如下。
  a. 输入 Assets 文件名称中的关键字，单击 **搜索**，搜索目标文件。
  b. 单击待加固 Assets 文件前的复选框，可以选择一个或多个目标 Assets 文件。

6. 单击 **确认加固**，加固应用。 页面上显示 **应用加固中** 提示时，表示加固任务已创建完成。单击 **查看加固列表** 进入 **应用安全加固** 页面查看安全加固列表。列表中新增当前任务的卡片，在卡片中，您可查看任务的加固进度，并下载加固后的 APK/AAB。

后续操作

- 下载安全加固包

5.3.2. 查看安全加固列表

安全加固列表中以卡片形式展示了已成功创建的加固任务信息。通过加固任务卡片，您可查看在任务中加固的应用信息、加固状态并实现下载加固包、查看失败日志以及删除加固任务等操作。

安全加固列表根据各任务的创建时间进行倒序排列，任务卡片中展示的信息包括：

- **应用名称**
  - 加固成功时，名称显示为蓝色文本。
  - 加固失败或加固中时，名称显示为黑色文本。
- **加固状态**

  在 **应用名称** 右侧显示当前任务的加固状态，包括 **已加固**、**加固中** 以及 **加固失败**。
  - **已加固** 表示当前加固任务已成功。
  - **加固中** 表示当前加固任务进行中。
  - **加固失败** 表示当前加固任务已失败。
- **包名**：上传的 APK/AAB 的包名。
- **版本号**：应用的版本号。
- **应用大小**：加固前的 APK/AAB 大小。
- **创建时间**：当前任务创建完成的时间。

操作步骤

查看安全加固列表的操作方法如下：

1. 进入mPaaS 控制台，在应用列表中选择目标应用。
2. 单击左侧导航栏中的 **移动应用安全** > **应用安全加固**，进入 **应用安全加固** 页面，在页面上查看安全加固列表。

相关链接

在任务卡片中，除了查看对应的应用信息和加固状态之外，您还可以进行以下操作：

- 下载安全加固包
- 查看安全加固详情
- 查看加固失败日志

5.3.3. 下载安全加固包

安全加固包（以下简称加固包）指加固后的 APK/AAB。本节引导您通过加固任务卡片，下载任务的安全加固包。

> **说明**
>
> 在状态为 **加固中** 或 **加固失败** 的任务卡片中，不提供加固包的下载入口。

操作方法如下：

1. 登录 mPaaS 控制台，在应用列表中选择目标应用。
2. 单击左侧导航栏中的 **移动应用安全** > **应用安全加固**，进入 **应用安全加固** 页面。
3. 在安全加固列表中，单击目标任务卡片中的 **加固包**，即可下载该任务的加固包。

> **重要**
>
> - 在对应用包加固过程中会删除应用中的签名信息，因此在下载安全加固包后，您需要对其重新签名，然后将重新签名过的加固包发布到对应的发布市场。
> - 下载完成后，请务必再次检查关键组件的功能是否正常，如升级组件、热修复组件。

如果安装包在加固后功能异常，请搜索群号 33417739 加入钉钉群进行咨询交流。

5.3.4. 查看安全加固详情

在创建完加固任务后，您可查看任务的安全加固详情。

- **基本信息**：展示加固的应用信息，包括 **应用名称**、**应用包名**、**应用版本** 和 **应用大小**（该应用大小指未加固时的应用大小）。
- **安全加固包**：提供安全加固包的下载入口。

> **重要**
>
> 在加固过程中会删除应用的签名信息，故在下载安全加固包后，您需要对其重新签名，然后将重新签名过的加固包发布到对应的发布市场。

- **加固详情**：在 **应用大小**、**MD5** 以及 **安全性** 3 个维度上，展示加固前后的对比。
- **安全加固的类**：展示已加固的类，及其加固前后的代码对比。
- **安全加固的 So 文件**：展示已加固的 So 文件。
- **安全加固的 Assets 文件**：展示已加固的 Assets 文件。

> **说明**
>
> 在创建加固任务时，如果选择了对类或 So 文件、 Assets 文件进行加固，则在该任务的 **安全加固详情** 页面上可以看到 **安全加固的类**、**安全加固的 So 文件** 和 **安全加固的 Assets 文件** 的相关信息。

操作步骤

可以查看状态为 **已加固** 任务的加固详情。状态为 **加固中** 或 **加固失败** 的任务无法查看加固详情。操作方法如下：

1. 登录mPaaS 控制台，在应用列表中选择目标应用。
2. 单击左侧导航栏中的 **移动应用安全** > **应用安全加固**，进入 **应用安全加固** 页面。
3. 在安全加固列表中，单击目标任务卡片中的应用名称，进入 **安全加固详情** 页面查看加固详情。
4. 查看安全加固的类、安全加固的 So 文件以及安全加固的 Assets 文件（可选），将鼠标移至 **安全加固的类** 后的问号上，可查看加固前后的代码对比。

> **说明**
>
> 反编译失败时，加固前后的代码截图将为空。

5.3.5. 查看加固失败日志

对于加固失败的任务，您可下载其加固失败的日志。状态为 **加固中** 或 **已加固** 的任务不提供查看加固失败日志的操作入口。

操作方法如下：

1. 登录 mPaaS 控制台，在应用列表中选择目标应用。
2. 单击左侧导航栏中的 **移动应用安全** > **应用安全加固**，进入 **应用安全加固** 页面。
3. 在安全加固列表中，找到目标加固任务，单击任务卡片右上角的 **下载失败日志** 图标按钮，即可下载加固失败的日志。

5.3.6. 搜索及删除任务

在安全加固列表，您可以搜索和删除加固任务。

搜索任务

您可根据应用名称或包名在安全加固列表中搜索目标任务。操作方法如下。

在安全加固列表右上方的搜索框中，输入应用名称或包名关键字，系统将根据输入的内容，实时搜索安全加固列表。

删除任务

您可通过以下方法来删除加固任务。

1. 登录 mPaaS 控制台，在应用列表中选择目标应用。
2. 单击左侧导航栏中的 **移动应用安全** > **应用安全加固**，进入 **应用安全加固** 页面。
3. 在安全加固列表中，单击目标任务卡片中的 **删除**，在弹出的提示框中，单击 **确定**，即可删除该任务。

5.4. 开放接口

5.4.1. 调用准备

对移动应用进行加固的操作可以通过调用 OpenAPI 来实现，实现用户服务端和 mPaaS 服务端的对接。

流程说明

接口调用流程如下：

1. 获取上传到 OSS 的 Token。
2. 将 APK/AAB 上传到 OSS。
3. 通知 MSA 已经上传 APK/AAB。
4. 查询上传结果（轮询），获取到加固任务 ID。
5. 通知 MSA 开始加固。
6. 查询加固结果（轮询），拿到加固后的应用 URL。
7. 下载加固后的包。

限流说明

为防止 OpenAPI 方式调用过于频繁从而对应用的运行产生影响，OpenAPI 在调用的过程中存在限流机制，具体限流策略如下：

- MSA 的 OpenAPI 是单机限流，限流维度是 appId+workspaceId。
- MSA 目前提供两台用于接收 OpenAPI 请求的设备，由负载均衡进行 OpenAPI 请求转发。
- 在单机维度，上传应用包的接口限流为每分钟 10 次，即每 6 秒内只能调用一次；其余接口限流为每分钟 600 次，即每 0.1 秒只能调用一次。

准备工作

在使用 OpenAPI 前，您需要先获取 App ID、 Workspace ID 与 Tenant ID，并配置 Maven 依赖及配置文件上传。

获取 App ID、 Workspace ID 与 Tenant ID

1. 登录 mPaaS 控制台，进入应用。
2. 在 **总览** 页，依次点击 **代码配置**（可视情况选择 Android 或 iOS） > **下载配置文件** > **立即下载**，在右侧弹出的 **代码配置** 窗口中，您可以看到 App ID、Workspace ID 和 Tenant ID 的值。

配置 Maven 依赖

在使用 OpenAPI 之前，您需要完成以下 Maven 依赖配置。

```
<dependency>
  <groupId>com.aliyun</groupId>
  <artifactId>aliyun-java-sdk-mpaas</artifactId>
  <version>3.0.3</version>
</dependency>

<dependency>
  <groupId>com.aliyun</groupId>
  <artifactId>aliyun-java-sdk-core</artifactId>
  <optional>true</optional>
  <version>[4.3.2,5.0.0)</version>
</dependency>
```

配置文件上传

由于在所有的 API 接口中均不允许出现文件流，所以需要上传的文件都应先调用上传工具类来将文件上传至 OSS，再将返回的 OSS 地址作为参数传递到指定的 API 中。

您可下载相关的文件的上传工具类 OssPostObject.java.zip。

使用示例

使用流程示例如下：mpaas-msa-client.zip。

5.4.2. 接口说明

本文详细介绍对移动应用进行安全加固的开放接口。

获取上传文件 token

请求 - GetFileTokenForUploadToMsaRequest

参数名称	类型	说明
appId	String	所属的 App。
workspaceId	String	所属的 Workspace。
tenantId	String	所属的租户。
onexFlag	Boolean	固定传值为 `true`。

返回值 - GetFileTokenForUploadToMsaResponse


```
{
    "resultContent":{
        "content":{
            "accessid":"LTAI****************",
            "dir":"mds/tempFileForOnex/ONEXE9B092D/test/PUQYHL/8b574cb7-3596-403f-a0e9-
208660fc2081/",
            "expire":"1584327372",
            "host":"https://mcube-test.oss-cn-hangzhou.aliyuncs.com",
            "policy":"QwM2YtYTBlOS0yMDg2NjBmYzIwODEvIl1dfQ==",
            "signature":"kisfP5YhbPtmES8+w="
        },
        "resultMsg":"",
        "success":true
    },
    "requestId":"8BAA3288-662E-422C-9960-2EEBFC08369F",
    "resultCode":"OK"
}
```


返回值说明

返回值名称	类型	说明
requestId	String	标识请求的 ID。
resultCode	String	正常情况下，请求返回的 code 是 `OK` 。若有其他情况，则表明 API 请求异常。
ResultContent.Content	Object	返回的具体对象，具体含义见下表。

在返回的对象中，包含的字段含义如下：

名称	类型	说明
resultMsg	String	查询失败后的返回值。
success	Boolean	查询是否成功。

通知 MSA 开始处理已经上传到 OSS 的应用

请求 - UploadUserAppToMsaRequest

参数名称	类型	说明

appId	String	所属的 App。
workspaceId	String	所属的 Workspace。
tenantId	String	所属的租户。
fileUrl	String	APK/ABB 上传后的地址。

返回值 - UploadUserAppToMsaResponse

```
{
    "resultContent":{
        "data":{
            "id": 12345,
            "enhanceTaskId": 12345,
            "progress": 10,
            "status": 0
        },
        "resultMsg":"",
        "success":true
    },
    "requestId":"637D5BE0-0111-4C53-BCEE-473CFFA0DBAD",
    "resultCode":"OK"
}
```

返回值说明

返回值名称	类型	说明
requestId	String	标识请求的 ID。
resultCode	String	正常情况下，请求返回的 code 是 `OK` 。若有其他情况，则表明 API 请求异常。
resultContent	Object	返回的具体对象，具体含义见下表。

在返回的对象中，包含的字段含义如下：

名称	类型	说明
data.id	Integer	上传任务的 ID，如果加固没有完成需要轮询检查。

data.enhanceTaskId	Integer	上传完成之后，会返回一个加固任务 ID，利用这个 ID 启动加固任务。
data.progress	Integer	加固的进度，范围 0 - 100。
data.status	Integer	上传状态，-1 为失败，0 为处理中，1 为上传成功。
resultMsg	String	查询失败后的返回值。
success	Boolean	查询是否成功。

查询处理上传应用状态

请求 - GetUserAppUploadProcessInMsaRequest

参数名称	类型	说明
appId	String	所属的 App。
workspaceId	String	所属的 Workspace。
tenantId	String	所属的租户。
id	Long	上传任务 ID。

返回值 - GetUserAppUploadProcessInMsaResponse

```
{
    "resultContent":{
        "data":{
            "id": 12345,
            "enhanceTaskId": 12345,
            "progress": 10,
            "status": 0
        },
        "resultMsg":"",
        "success":true
    },
    "requestId":"637D5BE0-0111-4C53-BCEE-473CFFA0DBAD",
    "resultCode":"OK"
}
```

返回值说明

返回值名称	类型	说明
requestId	String	标识请求的 ID。
resultCode	String	正常情况下，请求返回的 code 是 `OK` 。若有其他情况，则表明 API 请求异常。
resultContent	Object	返回的具体对象，具体含义见下表。

在返回的对象中，包含的字段含义如下：

名称	类型	说明
data.id	String	上传任务的 ID，如果加固没有完成需要轮询检查。
data.enhanceTaskId	String	上传完成之后，会返回一个加固任务 ID，利用这个 ID 启动加固任务。
data.status	Integer	上传状态，-1 为失败，0 为处理中，1 为上传成功。
resultMsg	String	查询失败后的返回值。
success	Boolean	查询是否成功。

启动加固任务

请求 - StartUserAppAsyncEnhanceInMsaRequest

名称	类型	说明
appId	String	所属的 App。
workspaceId	String	所属的 Workspace。
tenantId	String	所属的租户。
id	Long	加固任务 ID。

taskType	String	任务类型，有 `shell` 和 `enhance_shell` 两种，`shell` 是加壳，`enhance_shell` 搭配 Java2C 使用。
classes	String	设置需要 Java2C 加固的核心类，应当只添加关键核心类，使用英文逗号（,）隔开，如 `com.a.a,com.b.b`，如果设置了这个值，那么 `taskType` 需要为 `enhance_shell`，个别类不适配可能会导致加固失败。
totalSwitch	boolean	总开关，设置为 `true`，下面的开关才生效。
javaHook	Integer	防 Java Hook 能力，0 为 Killself，1 为 Warning。
memoryDump	Integer	防 memory dump 能力，0 为 Killself，1 为 Warning。
emulatorEnvironment	Integer	防模拟器能力，0 为 Killself，1 为 Warning。
nativeHook	Integer	防 Native Hook 能力，0 为 Killself，1 为 Warning。
dalvikDebugger	Integer	防 Java 调试能力，0 为 Killself，1 为 Warning。
nativeDebugger	Integer	防 Native 调试和 Root 能力，0 为 Killself，1 为 Warning。

返回值 - StartUserAppAsyncEnhanceInMsaResponse

```
{
    "resultContent":{
        "data":{
            "afterMd5": "aaaaaaaa",
            "afterSize": 1000,
            "appCode": "ONEXxxxx",
            "appPackage": "com.example.app",
            "beforeMd5": "bbbbbb",
            "id": 1,
            "label": "支付宝",
            "progress": 0,
            "status": 2,
            "taskType": "shell",
            "versionCode": 1,
            "versionName": "1.0.0",
            "enhancedClasses": ["aaa", "bbb"]
        },
        "resultMsg":"",
        "success":true
    },
    "requestId":"F9C681F2-6377-488D-865B-1144E0CE69D2",
    "resultCode":"OK"
}
```


返回值说明

返回值名称	类型	说明
requestId	String	标识请求的 ID。
resultCode	String	正常情况下，请求返回的 code 是 OK。若有其他情况，则表明 API 请求异常。
resultContent	Object	返回的具体对象，具体含义见下表。

在返回的对象中，包含的字段含义如下：

名称	类型	说明
resultMsg	String	查询失败后的返回值。
success	Boolean	查询是否成功。
data.afterMd5	String	加固后 APK/ABB 的 MD5。

data.afterSize	Long	加固后 APK/ABB 的大小。
data.id	Long	加固任务的 ID，后续用来轮询调用。
data.label	String	APK/ABB 的 label 字段。
data.progress	Integer	加固的进度，范围 0 - 100。
data.status	Integer	加固任务的状态： 0 未开始，1 已提交任务，2 加固中，3 加固成功，4 加固失败。
data.taskType	String	加固任务类型。
data.enhancedClasses	String[]	Java2C 选择的类。

查询加固任务进度

请求 - GetUserAppEnhanceProcessInMsaRequest

名称	类型	说明
appId	String	所属的 App。
workspaceId	String	所属的 workspace。
tenantId	String	所属的租户。
id	Long	加固任务 ID。

返回值 - GetUserAppEnhanceProcessInMsaResponse

```
{
    "resultContent":{
        "data":{
            "afterMd5": "aaaaaaaa",
            "afterSize": 1000,
            "appCode": "ONEXxxxx",
            "appPackage": "com.example.app",
            "beforeMd5": "bbbbbb",
            "id": 1,
            "label": "支付宝",
            "progress": 0,
            "status": 2,
            "taskType": "shell",
            "versionCode": 1,
            "versionName": "1.0.0",
            "enhancedClasses": ["aaa", "bbb"]
        },
        "resultMsg":"",
        "success":true
    },
    "requestId":"F9C681F2-6377-488D-865B-1144E0CE69D2",
    "resultCode":"OK"
}
```


返回值说明

返回值名称	类型	说明
requestId	String	标识请求的 ID。
resultCode	String	正常情况下，请求返回的 code 是 OK。若有其他情况，则表明 API 请求异常。
resultContent	Object	返回的具体对象，具体含义见下表。

在返回的对象中，包含的字段含义如下：

名称	类型	说明
resultMsg	String	查询失败后的返回值。
success	Boolean	查询是否成功。
data.afterMd5	String	加固后 APK/ABB 的 MD5。

data.afterSize	Long	加固后 APK/ABB 的大小。
data.id	Long	加固任务的 ID，后续用来轮询调用。
data.label	String	APK/ABB 的 label 字段。
data.progress	Integer	加固的进度，范围 0 - 100。
data.status	Integer	加固任务的状态： 0 未开始，1 已提交任务，2 加固中，3 加固成功，4 加固失败。
data.taskType	String	加固任务类型。
data.enhancedClasses	String[]	Java2C 选择的类。

查询加固后的产物下载链接

请求 - GetUserAppDownloadUrlInMsaRequest

参数名称	类型	说明
appId	String	所属的 App。
workspaceId	String	所属的 workspace。
tenantId	String	所属的租户。
taskId	String	加固任务 ID。

返回值 - GetUserAppDownloadUrlInMsaResponse

```
{
    "resultContent":{
        "data": { "url": "https://xxxx"},
        "resultMsg":"",
        "success":false
    },
    "requestId":"8F76783A-8070-4182-895D-14E5D66F8BA3",
    "resultCode":"OK"
}
```

返回值说明

返回值名称	类型	说明
requestId	String	标识请求的 ID。
resultCode	String	正常情况下，请求返回的 code 是 OK。若有其他情况，则表明 API 请求异常。
checkRsaKeyResult	Object	返回的具体对象，具体含义见下表。

在返回的对象中，包含的字段含义如下：

名称	类型	说明
data.url	String	APK/ABB 下载链接。
data.filename	String	APK/ABB 文件名。
resultMsg	String	查询失败后的返回值。
success	Boolean	查询是否成功。

查询加固日志

请求 - GetLogUrlInMsaRequest

参数名称	类型	说明
appId	String	所属的 App。
workspaceId	String	所属的 Workspace。
tenantId	String	所属的租户。
taskId	String	加固任务 ID。

返回值 - GetLogUrlInMsaResponse

```
{
    "resultContent":{
        "data": { "url": "https://xxxx"},
        "resultMsg":"",
        "success":false
    },
    "requestId":"8F76783A-8070-4182-895D-14E5D66F8BA3",
    "resultCode":"OK"
}
```

返回值说明

返回值名称	类型	说明
requestId	String	标识请求的 ID。
resultCode	String	正常情况下，请求返回的 code 是 OK。若有其他情况，则表明 API 请求异常。
resultContent	Object	返回的具体对象，具体含义见下表。

在返回的对象中，包含的字段含义如下：

名称	类型	说明
data	String	log 下载链接。
resultMsg	String	查询失败后的返回值。
success	Boolean	查询是否成功。

5.5. Android 加固后的问题排查

本文介绍 Android 使用加固后，带来的闪退问题排查。

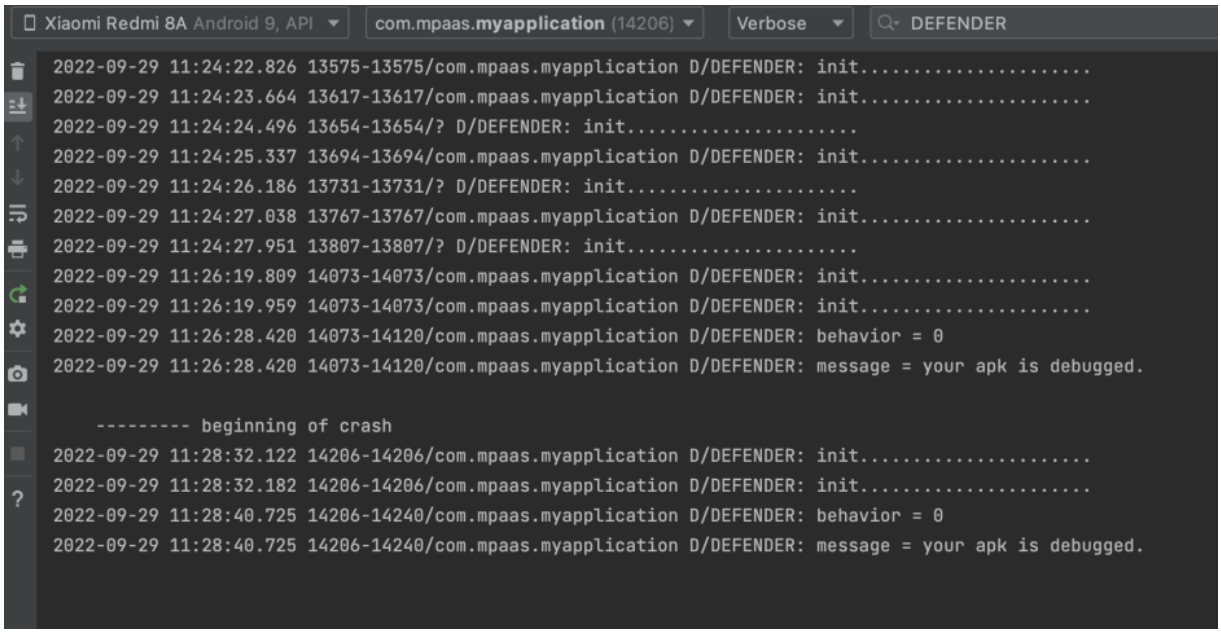
> **说明**
>
> 如果是加固前 App 可以正常运行，加固后启动闪退，那么其原因可能是触发到了加固服务的运行时检测危险项。此时服务会进行 kill，即 App 无法正常使用。加固的闪退可能是加固策略的保护目的，是期望中的闪退。

加固服务的日志中排查

App 启动闪退过滤日志，查看如下关键词：

DEFENDER

代表是我们加固服务自己打的日志



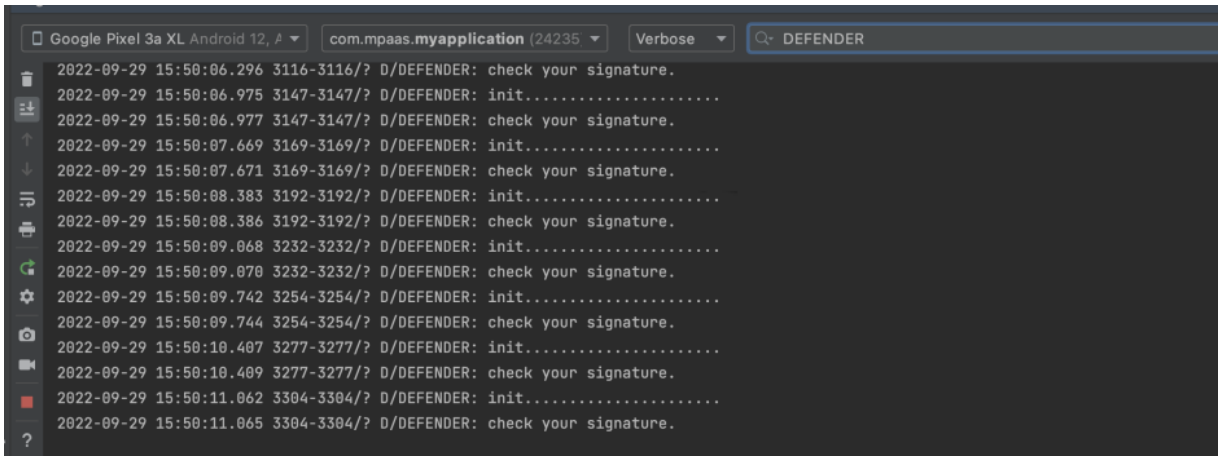
behavior

behavior 之后为处置方式：

- 0 代表退出 App，1 代表打印日志，2 代表弹窗。
- **message** 关键词后面的为闪退原因。

jaffer

jaffer 出现可能是签名问题，比如进行了重签名等。



check your **xxxx**（这个即是检测到了某个危险项的原因）。

应用闪退日志排查

FATAL

代表致命的，下面的堆栈信息就是闪退原因。

5.6. 常见问题

加固失败日志中存在 EnhanceError 的错误

原因分析：

在应用的 `AndroidManifest.xml` 文件中，`application` 标签未声明入口类。

解决方法：

在 `AndroidManifest.xml` 文件的 `application` 标签中，声明入口类。

选择的类未加固

原因分析：

在对类进行加固时，移动应用安全加固会判断目标类在加固后是否存在运行风险，若判断为存在风险，则会自动放弃加固。

解决方法：

不对加固后存在运行风险的类进行加固。

6.iOS 应用安全加固

本文引导您使用移动应用安全加固对 iOS App 进行加固。

使用须知

使用 mPaaS 应用安全加固前请阅读以下使用须知，并保证您的工程满足相关要求，否则可能导致加固失败或加固效果受损。

- 建议相关待加固代码采用 C、C++ 编写。iOS 加固对 C、C++ 的支持更好更稳定，此外部分支持 Ojective-C，不支持 Swift。
- 加固会带来性能损耗和理论上的稳定性风险提高，建议只对需要保护的核心代码进行加固，将需要保护的 C、C++ 代码抽离到一个单独的 Framework 中，然后进行加固，**不支持对整个源码 App 进行加固**。
- 目前支持 X86/M1 机器。您可从屏幕角落的苹果菜单中选取 **关于本机**，查看 Mac 的概览信息，包括处理器信息。如果显示为 Intel 处理器即表明您的 Mac 为 X86 架构。
- 目前支持 Xcode 14.1/14.2/15.0.1。由于 iOS 加固会对编译器进行处理，需要对特定 Xcode 进行适配，所以使用 iOS 加固时需要使用特定版本的 Xcode。

  > **重要**
  >
  > 自 2024 年 4 月 29 日起，上传到 App Store Connect 的 App 必须是使用 Xcode 15 为 iOS 17、iPadOS 17、Apple tvOS 17 或 watchOS 10 构建的 App。

- 请确保 App 工程的工作空间设置为 **New Build System**。检查路径为 **Xcode > File > Project Settings > Build System**。

操作步骤

1. 配置环境文件。按照以下方法生成 `MSAConfig.json` 文件，并放到 `$HOME` 目录下。在 Mac 机器下打开命令行输入 `echo $HOME`，即可得到 `$HOME` 目录，使用时替换为真实值即可，字段如下：

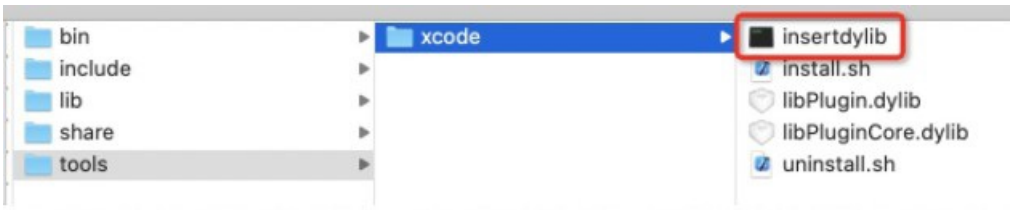
   ```
   {
       "appId": "应用 appId",
       "workspaceId": "应用 workspaceId",
       "tenantId": "应用 tenantId",
       "accessKeyId": "阿里云账号 accessKeyId",
       "accessKeySecret": "阿里云账号 accessKeySecret",
       "license": "目前空即可",
       "domain":"xxx"
   }
   ```

   > **说明**
   >
   > - domain 的值分别为 ap-southeast-1 和 cn-hongkong，分别对应新加坡和中国香港环境。
   > - 其他字段值获取方式请参考 iOS 加固配置文件信息获取方式。

2. 安装加固工具。

i. 下载加固工具，解压后进入目录 **tools> xcode**。

> **说明**
>
> - xcodeplugin-x86_64-5.9.0.zip 适用于 Xcode 15.0.1 + Mac X86。
> - xcodeplugin-arm64-5.9.0.zip 适用于 Xcode 15.0.1 + Mac M1。
> - xcodeplugin-x86_64-5.7.2 适用于 Xcode 14.1/14.2 + Mac X86，操作系统建议 13.2.1。
> - xcodeplugin-arm64-5.7.2 适用于 Xcode 14.1/14.2 + Mac M1，操作系统建议 13.2.1。



ii. 打开 insertdylib 文件，并在确认弹窗中单击 **Open**。

iii. 执行以下命令。

```
sh ./tools/xcode/install.sh
```

> **说明**
>
> 该命令会自动查找与替换 `/Applications/Xcode.app/` 的编译器，如需要恢复可执行 `sh ./tools/xcode/uninstall.sh` 命令。

3. 使用 Xcode 打开 Framework/ipa 工程，执行 `Build/Archive` 命令，暂不支持动态库。

> **重要**
>
> 工程路径名称中不能有空格或中文，否则可能导致编译报错。

4. 处理完成后使用以下命令验证加固，如果加固成功则输出不为空。

```
nm ./BinaryPath | grep obfuscator
```

后续步骤

加固后，请务必检查关键组件功能是否正常，如果安装包在加固后功能异常，请 提交工单 联系 mPaaS 技术支持人员。