



ТЕСТ ЗА ПРОНИКВАНЕ В МОБИЛНО ПРИЛОЖЕНИЕ · СЪСТАВЕН ПРИМЕР

# Съставен примерен Тест за проникване в мобилно приложение

Изготвен за [Клиент] от PXLSecurity LTD.

Това е примерен документ, не предаван на клиент. Той е съставен от формата на няколко анонимизирани ангажимента на PXLSec срещу iOS, Android и кросплатформени приложения и API-тата зад тях. Всяка констатация е преписана със синтетични, илюстративни данни върху резервирани примерни имена, пакети и адреси. Той не описва нито един действителен клиент и в него не присъства действителен клиент, доставчик, продукт, хост, човек или данни за достъп.

|                       |                                                                |
|-----------------------|----------------------------------------------------------------|
| Номер на документа    | PXL-2026-0311                                                  |
| Версия                | 1.0 (окончателна)                                              |
| Класификация          | Конфиденциално — примерен документ, не съдържа клиентски данни |
| Вид на оценката       | Тест за проникване в мобилно приложение (grey box)             |
| Платформи             | Издателски билдове за iOS, Android и Windows                   |
| Оценено спрямо        | OWASP MASVS, с MASTG като ръководство за тестване              |
| Период на ангажимента | [Дати на ангажимента]                                          |
| Изготвен от           | PXLSecurity LTD. · София, България · VAT BG203239331           |

## СЪДЪРЖАНИЕ

# Какво съдържа този отчет

Написан, за да бъде четен по два начина: раздел 1 стои самостоятелно, а подробните констатации носят всичко, което един инженер има нужда, за да възпроизведе и поправи всеки проблем.

| РАЗДЕЛ                                                                                                                     |                                                     | СТР. |
|----------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------|------|
| 1                                                                                                                          | Резюме за ръководството                             | 4    |
| Какво беше тествано, основната картина на риска и трите неща, които да поправите първо — написано за нетехнически читател. |                                                     |      |
| 2                                                                                                                          | Обхват на ангажимента                               | 6    |
| Платформи, билдове, акаунти, API-то зад приложението и допусканията, от които зависи резултатът.                           |                                                     |      |
| 3                                                                                                                          | Методология                                         | 7    |
| Петте етапа, осемте покрити групи контроли на MASVS и правилата за безопасност, при които тествахме.                       |                                                     |      |
| 4                                                                                                                          | Оценки на критичността                              | 9    |
| Скалата и защо заобикалянето на контрол за устойчивост не е автоматично критично.                                          |                                                     |      |
| 5                                                                                                                          | Обобщение на констатациите                          | 10   |
| Всяка констатация с нейния идентификатор, критичност, група по MASVS и статус, в реда на отстраняване.                     |                                                     |      |
| 6                                                                                                                          | Подробни констатации                                | 11   |
| По един раздел за всяка констатация: описание, доказателство, възпроизвеждане, въздействие и конкретна поправка.           |                                                     |      |
| 7                                                                                                                          | План за отстраняване                                | 26   |
| Поправките, групирани на вълни, с отговорник и последователност.                                                           |                                                     |      |
| 8                                                                                                                          | Повторен тест и приключване                         | 28   |
| Какво покрива повторният тест и какво заявява писмото за приключване.                                                      |                                                     |      |
| A                                                                                                                          | Приложение А — Билдове, акаунти и инструменти       | 29   |
| Какво поискахме и защо всяка точка имаше значение.                                                                         |                                                     |      |
| B                                                                                                                          | Приложение Б — Извън обхвата и какво този тест не е | 30   |
| Какво беше изключено и съседната работа, която този ангажимент не извършва.                                                |                                                     |      |
| C                                                                                                                          | Приложение В — Препратки и какво представляват      | 31   |
| MASVS, MASTG и останалите, описани точно, без да се подразбира каквато и да е сертификация.                                |                                                     |      |
| D                                                                                                                          | Приложение Г — Как да четете този отчет             | 32   |
| Кой кой раздел да прочете и как да действа по него.                                                                        |                                                     |      |

Бележка какво представлява този примерен документ. Констатациите F-01 до F-09 са взети от формата на няколко различни анонимизирани ангажимента и представени тук, сякаш идват от един, за да може потенциален клиент да види целия обхват на работата с едно прочитане. Всеки запис, име на пакет, идентификатор, акаунт и оценка е синтетичен и нито модел на устройство, нито версия на рамка, нито търговски инструмент е назован като принадлежащ на клиент.

## РАЗДЕЛ 1

## Резюме за ръководството

PXLSecurity LTD. беше нает от [Клиент] да извърши ръчен тест за проникване на мобилно приложение и API-то зад него. Този раздел е написан за нетехнически читател. Той казва какво беше тествано, колко сериозен е резултатът и какво да се направи първо.

### Какво беше тествано

Старши инженери тестваха приложението на ръка през договорения период, работейки от разпространените издателски билдове, а не от изходния код, защото това е, което нападателят има. Съставният обхват зад този примерен документ покриваше билдовете за iOS, Android и Windows на един кросплатформен продукт, данните, които приложението съхранява на устройството, поведението му под инструментирани върху устройство, контролирано от тестера, и API-то, с което приложението говори. Тестването беше структурирано около OWASP Mobile Application Security Verification Standard, а всяка констатация по-долу назовава групата контроли, в която попада.

Две неща си заслужава да бъдат изречени направо преди резултатите, защото определят как трябва да се четат констатациите. Първо, мобилното приложение е единствената част от една система, която се подава на нападателя: щом бъде публикувано, двоичният файл, данните, които записва, и работещият процес са достъпни за всеки, който го инсталира — така че слабост, компилирана в артефакта, не може да бъде отменена с промяна на сървър. Второ, контролите, които живеят само на устройството — проверки за цялост, обвързване със сертификат, защита от подправяне — повишават цената на една атака, но не я предотвратяват. Очакваме да ги преодолеем и отчитаме какво са защитавали, а не факта, че са паднали.

### Заглавието

|             |          |      |           |             |
|-------------|----------|------|-----------|-------------|
| 9           | 1        | 3    | 3         | 8           |
| констатации | critical | high | платформи | MASVS групи |

Резултатът се определя от една констатация. Приложението се удостоверява пред сървърната си част с клиентска тайна, компилирана в пакета, идентична във всяко инсталирано копие и присъстваща на всичките три платформи. Възстановяването ѝ не изискваше използване на уязвимост и акаунт: издателският пакет беше разпакетиран като архив и стойността прочетена от него. Този единствен проблем превръща публикувано приложение в работещи данни за достъп до вътрешно API за данни, притежавани от всеки, който някога го е изтеглял.

Зад нея стоят три проблема с висока критичност, които споделят един модел: всеки е контрол, който съществува и е почти правилен. Офлайн базата данни е шифрована, но с ключ, който самото устройство може да възпроизведе. Обвързването със сертификат е реализирано, но настройка по подразбиране в профила за управление го изключва мълчаливо в конфигурацията, която повечето клиенти действително внедряват. Локална помощна услуга е защитена с токен, а после подава този токен на всеки извикващ, който поиска. Нито един от тези случаи не е отсъстващ контрол;

всичките три са контроли, които не издържат срещу заплахата, за която са били изградени.



Констатации по критичност. Числата са от този обобщен примерен документ и не са еталон за каквато и да е съвкупност от приложения.

### Трите неща, които да направите първо

- 1 Подменете данните за достъп до сървърната част днес, преди каквато и да е промяна в кода. Те са действащи, публични са, а подмяната е единственото действие, което намалява изложеността незабавно. Очаквайте, че подмяната ще изисква издание на клиента, и планирайте това издание сега, а не след това.
- 2 Спрете да извеждате ключа на офлайн базата данни от устройството. Преместете го в хранилище, подкрепено от хардуер, така че притежаването на устройството да не дава данните върху него. Това е промяната, която превръща загубено или откраднатото устройство в ограничено събитие.
- 3 Обърнете подразбирането за обвързването със сертификат. Когато профилът за управление мълчи, налагайте обвързване. Конфигурацията, до която клиентът достига, следвайки документацията, трябва да е безопасната, не позволяващата.

Какво този отчет не казва. Този отчет описва какво успяхме да направим в рамките на договорения обхват и период. Той не е сертификация, одиторско мнение или твърдение, че приложението е без други слабости, а PXLSecurity LTD. не е одитор, оценител или сертифициращ орган. Структурирането на работата около OWASP MASVS е начин покритието да стане четимо; то не е оценка спрямо стандарта и не предполага членство в каквато и да е схема.

## РАЗДЕЛ 2

## Обхват на ангажимента

Какво беше в обхвата, какво ни беше предоставено и допусканията, на които се опира резултатът. Една констатация е значима само колкото обхвата, в който е открита.

|                 |                                                                                                                                                                                                 |
|-----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Цели            | Издателските билдове за iOS, Android и Windows на едно кроссплатформено приложение, плюс сървърното API, което то ползва.                                                                       |
| Артефакти       | Подписани издателски билдове, предоставени от [Клиент], съответстващи на версиите в публичното и управляваното разпространение към момента на тестването.                                       |
| Вид на оценката | Grey box. Без изходен код; бяха предоставени тестови акаунти и описание на ролите. Артефактът беше третиран както би го третирал нападател.                                                     |
| Акаунти         | По два акаунта за всяка роля в изолиран тестов клиентска организация, така че поведението между акаунти и между клиентски организации да може да бъде доказано, а не изведено по предположение. |
| Устройства      | Физически устройства и емулатори, контролирани от тестера, както в състояние по подразбиране, така и в привилегировано, плюс профил за управлявано внедряване за корпоративната конфигурация.   |
| Период          | [Дати на ангажимента], като тестването на сървърната част беше ограничено до договорената тестова среда, освен където е отбелязано.                                                             |
| Стандарти       | OWASP MASVS като стандарт за проверка; OWASP MASTG като ръководство за тестване; OWASP API Security Top 10 за сървърната страна.                                                                |
| Изключения      | Вижте Приложение Б. Накратко: без отказ на услуга, без тестване на самите операционни системи, без атака срещу магазините за приложения или платформата на клиента за управление на устройства. |

### Допускания, от които зависи този резултат

- Тестваните билдове са билдовете, които се разпространяват. Ако процесът по издаване построява или подписва отново след артефакта, който ни беше предоставен, констатациите описват различен двоичен файл от онзи, който клиентите ви инсталират.
- Профилът за управлявано внедряване, който ни беше предоставен, отразява как клиентите действително внедряват. F-03 зависи от настройките по подразбиране на този профил, така че съществено различен профил променя критичността на тази констатация.
- Тестовият клиентска организация се държи като производствен. Там, където контрол от страна на сървъра се различава между двата, констатация, доказана в тестовата среда, може да се държи различно в производствената — отбелязваме всяка констатация, при която това е под въпрос, и нито една от констатациите по-долу не е такава.
- Нищо в този отчет не допуска компрометирана операционна система. Там, където една констатация изисква привилегировано състояние на устройството, констатацията го посочва и е оценена съответно.

## РАЗДЕЛ 3

## Методология

Как протече ангажиментът, какво покри тестването и правилата, при които се проведе.



|                                |                                                                                                                                                                     |
|--------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Обхват и билдове               | Платформи, издателски артефакти, акаунти, устройства и правила на ангажимента, договорени писмено, преди да бъде докоснато каквото и да е.                          |
| Статичен анализ                | Разпространеният артефакт, разпакетиран и подложен на обратно инженерство: текстови низове, ключов материал, крайни точки, манифест, изнесена повърхност и логика.  |
| Динамично и по време на работа | Инструментирано тестване върху устройство, контролирано от тестера: прихващане на функции, прихващане на трафик, проверка на хранилището и заобикаляне на контроли. |
| API и навързване               | Това, което клиентът излага, се проследява навътре в услугата зад него, а констатациите се навързват, за да се установи действително въздействие.                   |
| Отчет и повторен тест          | Констатациите, съотнесени към групите контроли на MASVS и приоритизирани, после проверени при повторния тест и потвърдени писмено.                                  |

## Защо редът има значение

Последователността по-горе е на нападателя, не на одитор. Прочитането на артефакта е първо, защото е безплатно, безшумно и не изисква акаунт: всичко, възстановено там, е достъпно за всеки, който инсталира приложението — затова F-01 е най-сериозната констатация в този отчет, въпреки че е най-малко изискващата технически. Инструментирането е второ, защото установява какво контролите на страната на устройството действително защитават. API-то е последно, защото дотогава знаем в какво вярва клиентът и можем да тестваме дали сървърът вярва в същото.

## Покритие спрямо групите контроли на MASVS

OWASP Mobile Application Security Verification Standard организира сигурността на мобилните приложения в групи контроли. Структурираме както тестването, така и този отчет около тях, за да бъде покритието четимо и пропуските видими. Дясната колона показва кои констатации в този примерен документ са попаднали във всяка група — група без констатации въпреки това е била тествана.

| Контролна група | Какво покрива                                                                     | Констатации      |
|-----------------|-----------------------------------------------------------------------------------|------------------|
| MASVS-STORAGE   | Данни, съхранявани на устройството, и какво го напуска в журнали и резервни копия | F-01, F-02, F-06 |

| Контролна група           | Какво покрива                                                              | Констатации |
|---------------------------|----------------------------------------------------------------------------|-------------|
| <b>MASVS - CRYPTO</b>     | Управление на ключовете и правилната употреба на криптографските примитиви | F-02        |
| <b>MASVS - AUTH</b>       | Удостоверяване и оторизация, включително локална и биометрична защита      | F-08        |
| <b>MASVS - NETWORK</b>    | Сигурност на транспорта и проверка на сертификати                          | F-03        |
| <b>MASVS - PLATFORM</b>   | Взаимодействие с платформата: IPC, изнесени компоненти, WebView            | F-04, F-07  |
| <b>MASVS - CODE</b>       | Конфигурация на билда, зависимости и какво носи артефактът                 | F-01        |
| <b>MASVS - RESILIENCE</b> | Проверки за цялост, защита от подправяне и цената на обратното инженерство | F-05, F-09  |
| <b>MASVS - PRIVACY</b>    | Работа с лични данни и какво приложението разкрива за своя потребител      | F-06        |

Структурирането на тестването около MASVS не е оценка спрямо него. PXLSec не е сертифициращ орган и тази таблица е помощно средство за покритие, не твърдение за съответствие.

### Как се използват инструментите

Инструментите разпакетират, декомпилират, инструментируют и прихващат; те не решават какво има значение. Статичният анализ на мобилен артефакт произвежда голям обем изход с ниска стойност — всяко разрешение, всеки изнесен компонент, всеки текстов низ, който наподобява ключ — а работата е в разделянето на трите, които са действителни, от трестата, които не са. Всяка констатация в този отчет беше потвърдена на ръка срещу работещо приложение, а констатация, която не можа да бъде демонстрирана, не е в отчета.

### Правила за безопасност, при които тестваме

- Потвърждение само с четене навсякъде, където едно прочитане доказва твърдението. Там, където валидността на данни за достъп е констатацията, статусният код я установява, а тялото на отговора се отхвърля непрочетено.
- Не се извличат, копират или запазват клиентски данни. Там, където едно хранилище беше отворено, беше прочетен броят на записите и нищо друго.
- Промените в състоянието се извършват срещу синтетични записи в изолиран тестов клиентска организация и се връщат обратно.
- Без отказ на услуга и без техника, чийто режим на провал е счупено устройство или повредено хранилище.
- Техниките за заобикаляне не се публикуват. Там, където контрол беше преодолян, отчетът посочва какво контролът е защитава и защо е паднал, а не скрипта, който го е направил.

## РАЗДЕЛ 4

## Оценки на критичността

Какво означава всяка оценка тук и как CVSS я информира, без да я определя.

| Оценка        | Какво означава в този отчет                                                                                                                                                                     |
|---------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Critical      | Пряк, неустойчив достъп до клиентски данни или до власт, която ги достига, наличен за всеки, който инсталира приложението. Поправете преди следващото издание и подменете сега всичко разкрито. |
| High          | Контрол, който съществува, но не издържа, водещ до излагане на данни или достъп до акаунт при условия, които нападателят може реалистично да подреди.                                           |
| Medium        | Слабост, която съществено намалява цената на атака или усилва друга констатация, но сама по себе си не достига клиентски данни.                                                                 |
| Low           | Тясна изложеност, изискваща условия, които нападателят не може лесно да подреди, или функция, която не осигурява защитата, която подразбира.                                                    |
| Informational | Наблюдение, по което си заслужава да се действа, но не е уязвимост. Не се присвоява оценка.                                                                                                     |

### Защо CVSS не определя оценката

Вектор по CVSS се дава за всяка оценена констатация, защото читателите го искат и защото се пренася добре между инструментите. Той е входни данни, не отговорът. CVSS няма начин да изрази факта, че тайна вътре в разпространен артефакт не може да бъде отменена, което е точно това, което прави F-01 констатацията, която е; нито че критичността на F-03 произлиза от конфигурацията, която повечето клиенти внедряват, а не от трудността на атаката. Там, където нашата оценка и аритметиката се разминават, оценката отразява бизнес последствието и този отчет казва защо.

Бележка, специфична за мобилните приложения. Отчетите в тази област често оценяват всяко заобикаляне на контрол за устойчивост като критично, защото заобикалянето е драматично и лесно за демонстриране. Ние не го правим. Откриването на root, обвързването със сертификат и защитата от подправяне повишават цената на атака върху устройство, което нападателят вече контролира; преодоляването им се очаква. F-05 е оценена като Medium именно по тази причина и оценката описва до какво заобикалянето е дало достъп, а не факта, че е проработило. Отчет, който извежда заобикаляне на jailbreak като водеща констатация, обикновено е запълване.

## РАЗДЕЛ 5

## Обобщение на констатациите

Всяка констатация, в реда, в който бихме ги поправили.

| ID   | Констатация                                                          | Критичност    | MASVS             | Статус    |
|------|----------------------------------------------------------------------|---------------|-------------------|-----------|
| F-01 | Backend API credentials compiled into the shipped application bundle | Critical      | STORAGE · CODE    | Отворен а |
| F-02 | Offline database key derivable from material readable on the device  | High          | STORAGE · CRYPTO  | Отворен а |
| F-03 | Certificate pinning disabled by a management-profile default         | High          | NETWORK           | Отворен а |
| F-04 | Embedded local service discloses its own authorisation token         | High          | PLATFORM          | Отворен а |
| F-05 | Root and jailbreak detection defeated at a single decision point     | Medium        | RESILIENCE        | Отворен а |
| F-06 | Session token written to the device log and included in backups      | Medium        | STORAGE · PRIVACY | Отворен а |
| F-07 | Exported component performs a state change without authentication    | Medium        | PLATFORM          | Отворен а |
| F-08 | Biometric prompt gates the interface rather than the credential      | Low           | AUTH              | Отворен а |
| F-09 | Obfuscation applied, but not across the security-relevant code       | Informational | RESILIENCE        | Отворен а |

Разпределение по критичност: 1 critical, 3 high, 3 medium, 1 low, 1 informational. Статусът е към датата на този отчет; писмото от повторния тест отразява приключването.

## РАЗДЕЛ 6

## Подробни констатации

По един раздел за всяка констатация: критичност и илюстративен CVSS, засегнат компонент, какво беше открито, доказателството, път за възпроизвеждане, въздействието и поправка.

За доказателствата по-долу. В действителен отчет всеки запис е заличена снимка на екрана от собствената сесия на инженера, с пълните двойки заявка и отговор в приложението. Всеки запис тук е синтетична реконструкция — операциите са представителни; пакетите, хостовете, идентификаторите, ключовете и изходът са измислени. В този документ никъде не присъства работещ полезен товар, hook скрипт или рецепта за заобикаляне: възпроизвеждането е написано на нивото, което използва един бюлетин на доставчик.

**CRITICAL****F-01** Backend API credentials compiled into the shipped application bundle

| КРИТИЧНОСТ | CVSS v3.1 (ИЛЮСТРАТИВНО)                | ЗАСЕГНАТ КОМПОНЕНТ                                          | MASVS          |
|------------|-----------------------------------------|-------------------------------------------------------------|----------------|
| Critical   | 9.1 AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:N | Release builds for all three platforms — application bundle | STORAGE · CODE |

Class: Use of hard-coded credentials (CWE-798) · insufficiently protected credentials (CWE-522) · Reference: OWASP MASVS-STORAGE-2 and MASVS-CODE-4 · MASTG static analysis of the application package · OWASP Mobile Top 10 M1 (improper credential usage) · Status: Open · Composite element: cross-platform application distributed to other organisations

**Описание**

The application authenticates to its backend with a client identifier and secret that are compiled into the bundle. They are not in a configuration file fetched at runtime, not held in the platform keystore and not issued per installation: they are constants in the shipped artefact, identical in every copy, and the same three values appear in the iOS, Android and Windows builds.

Recovering them required no exploitation. The release package was unpacked as an archive and its string table read. The credential proved live against the production API, returning an authenticated response to a read-only probe. This is the finding that defines the engagement, because it is not reachable by the usual remedies: the secret cannot be revoked by a server-side configuration change without breaking every installed copy, and it remains valid on every device that does not take an update, which in a distributed application is a population the vendor does not control.

## Доказателство

## Backend API credentials compiled into the shipped application bundle

SYNTHETIC EXAMPLE

Strings recovered from the public release build authenticate against an internal data API.

```
analyst@wkstn ~$ bash

analyst@wkstn:~/mobile$ unzip -q haldon-release.apk -d ./unpacked
Synthetic transcript — invented package, host and key material
analyst@wkstn:~/mobile$ strings unpacked/assets/app.bundle | grep -iE 'api_(key|secret)'

api_base      = https://api.example.test/v3
api_client_id = haldon-mobile-prod
api_secret    = sk_live_7f3a...<truncated - fabricated value>

# the same three values appear in the iOS and Windows builds

analyst@wkstn:~/mobile$ curl -s -o /dev/null -w '%{http_code}' \
-u haldon-mobile-prod:sk_live_7f3a...<truncated - fabricated value> \
https://api.example.test/v3/records?limit=1

200 - credential is live and authenticated

Read-only check only. No record was retrieved, stored or exfiltrated.
```

i Illustrative example — synthetic data, not client evidence.

Evidence for F-01 — the release package is unpacked, the credential is read out of the bundle, and a read-only probe confirms it authenticates against the production API. Illustrative reconstruction using synthetic data on the reserved example.test domain and the documentation package namespace com.example.haldon; not client evidence.

## Стъпки за възпроизвеждане

- 1 Obtain the release artefact the way any user would — from the public listing or the managed distribution catalogue. No device compromise, no account and no prior access is required.
- 2 Unpack the package as an archive and search the bundled assets and string tables for credential-shaped constants. In the sample transcript this returns the API base, the client identifier and the client secret directly.
- 3 Confirm the credential is live with a single read-only request that asks for the smallest possible response. The sample shows **200** — authenticated against the production API.
- 4 Stop there. The response body was discarded unread and no record was retrieved, stored or exfiltrated; the finding is the credential's validity, and that is established by the status code alone.

## Въздействие

Anyone who downloads the application holds a working credential for an internal data API. That is the entire population of the app's distribution, which for an application published to other organisations is unbounded and unauthenticated. The access is indistinguishable from legitimate application traffic, so it produces no anomaly for anyone to detect, and it persists independently of the vendor's own session handling: revoking a user's account does not revoke this. Because the same values ship on all three platforms, there is no partial exposure and no platform that can be treated as unaffected.

## Препоръка

- 1 Treat the credential as compromised and rotate it now, ahead of any code change. Review the API's own access logs for use from outside the expected client population, and expect that a rotation will require a client release.
- 2 Remove the shared secret from the artefact entirely. The application should authenticate as the user, with a short-lived token issued on sign-in, rather than as itself with a constant the bundle carries.

- 3 Where the client genuinely needs its own identity, issue it per installation at first run and store it in the platform keystore or keychain, so that recovering one device's credential does not yield every device's.
- 4 Add a release gate that scans the built artefact — not the source tree — for credential-shaped constants, so this class of defect is caught in the thing that actually ships.

HIGH

**F-02** Offline database key derivable from material readable on the device

| КРИТИЧНОСТ | CVSS v3.1 (ИЛЮСТРАТИВНО)                | ЗАСЕГНАТ КОМПОНЕНТ                                    | MASVS            |
|------------|-----------------------------------------|-------------------------------------------------------|------------------|
| High       | 7.4 AV:P/AC:L/PR:N/UI:N/S:U/C:H/I:N/A:N | Android and iOS builds — encrypted offline data store | STORAGE · CRYPTO |

Class: Use of a risky cryptographic algorithm (CWE-327) · insufficient key derivation (CWE-916) · Reference: OWASP MASVS-CRYPTO-1 and MASVS-STORAGE-1 · MASTG testing of data storage and key management · OWASP Mobile Top 10 M10 (insufficient cryptography) · Status: Open · Composite element: application caching records on the device

**Описание**

The application supports working offline and caches records in an encrypted local database. The encryption is real and the library behind it is sound; the weakness is in where the key comes from. It is derived by hashing a device identifier concatenated with a fixed application string — no salt, no key-derivation function and a single iteration.

Every input to that derivation is readable on the device the database sits on, which means the key travels with the thing it is supposed to protect. Anyone with the device, a backup of it or filesystem-level access can reproduce the key and open the store. The protection holds against someone who copies the database file alone and nothing else, which is not the threat the feature was designed against.

**Доказателство**

**Offline database key derivable from material readable on the device** SYNTHETIC EXAMPLE

The at-rest encryption key is derived from an unsalted device identifier, so possession of the device is enough to reproduce it.

```
analyst@wkstn ~$ ls -l /data/data/com.example.haldon/databases/
-rw-rw-r-- 1 analyst 4.1M encrypted

# key derivation recovered by decompiling the storage class
key = SHA256( ANDROID_ID + "haldon" )    no salt, no KDF, 1 iteration

analyst@wkstn ~$ python3 derive_key.py --android-id 9a1c...<truncated - fabricated value>
derived key: 4e9b...<truncated - fabricated value>

analyst@wkstn ~$ sqlcipher offline.db
sqlite> PRAGMA key = '4e9b...<truncated - fabricated value>';
sqlite> SELECT count(*) FROM contacts;

ok - store opened; row count read, no rows retrieved

! Hardware-backed key storage would make the key unreproducible off the device.
```

i Illustrative example — synthetic data, not client evidence.

Evidence for F-02 — the derivation is recovered from the decompiled storage class, the key is reproduced from a device identifier, and the store opens. Illustrative reconstruction using synthetic identifiers and fabricated key material; not client evidence.

## Стъпки за възпроизвеждане

- 1 Locate the encrypted store in the application's private data directory and confirm it is encrypted rather than merely inaccessible.
- 2 Decompile the storage class and read the derivation. In the sample it is a single unsalted hash over a device identifier and a constant string.
- 3 Reproduce the key from the device identifier alone, with no secret from the user and no material from the server.
- 4 Open the store with the derived key and confirm access by reading a row count. Testing stopped at the count: no rows were read, extracted or retained.

## Въздействие

The offline store holds the records the user has worked with, which is often a substantial subset of what their account can reach. A lost or stolen device, a device handed on without a wipe, or an unencrypted backup extracted from a workstation all yield that data in plaintext. Because the key derivation is identical across installations, the technique is not specific to one device and scales to any device recovered. The encryption satisfies a checklist that asks whether data at rest is encrypted while providing very little of the protection that question is meant to establish.

## Препоръка

- 1 Move the key into hardware-backed storage — the Android Keystore or the iOS Keychain with an appropriate protection class — so that it is generated on the device, never leaves the secure element and cannot be reproduced from anything readable.
- 2 If a user secret must take part in the derivation, use a current password-based key-derivation function with a per-installation random salt and a deliberately costly iteration count. Never derive a key from device identifiers alone.
- 3 Bind the key's release to user authentication, so that unlocking the device is required before the store can be opened, and re-derive on credential change.
- 4 Reduce what is cached. The strongest control available here is to hold fewer records offline and expire them sooner; the data that is not on the device needs no key at all.

HIGH

**F-03** Certificate pinning disabled by a management-profile default

| КРИТИЧНОСТ | CVSS v3.1 (ИЛЮСТРАТИВНО)                | ЗАСЕГНАТ КОМПОНЕНТ                                          | MASVS   |
|------------|-----------------------------------------|-------------------------------------------------------------|---------|
| High       | 7.4 AV:A/AC:H/PR:N/UI:N/S:U/C:H/I:H/A:N | All platforms — transport security under managed deployment | NETWORK |

Class: Improper certificate validation (CWE-295) · insecure default initialisation (CWE-1188) · Reference: OWASP MASVS-NETWORK-1 and MASVS-NETWORK-2 · MASTG testing of network communication · OWASP Mobile Top 10 M5 (insecure communication) · Status: Open · Composite element: application deployed through managed device fleets

## Описание

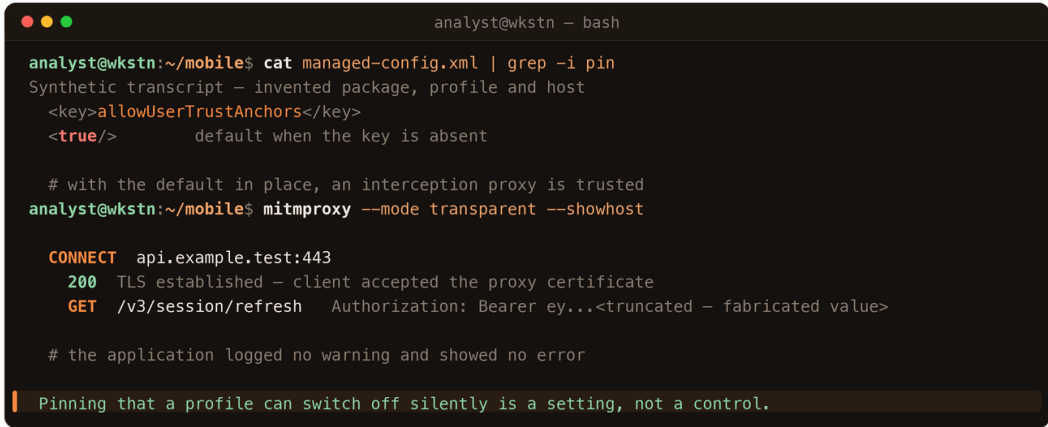
The application implements certificate pinning, and in a default installation it works. The problem appears in the configuration the product is actually deployed under. When the application is installed through a customer's device-management platform, a managed-configuration key governs whether user-installed trust anchors are honoured, and the value that applies when the key is absent is the permissive one.

Customer administrators deploying the application do not normally set a key they have not been told about, so the permissive default is the configuration the product runs under in the field. With it in place the pinning is not enforced, an interception proxy is trusted, and the full session — including the bearer token on the refresh call — is readable. Nothing in the application surfaces the change: no warning, no error, no indicator that a control the vendor believes is active has been switched off by a default nobody chose.

### Доказателство

**Certificate pinning disabled by a management-profile default**SYNTHETIC EXAMPLE

A managed deployment default turns pinning off without surfacing the change, allowing full interception of the session.



*Illustrative example — synthetic data, not client evidence.*

Evidence for F-03 — the managed-configuration default is read from the profile, and with it in place the client accepts an interception proxy and the session is readable. Illustrative reconstruction using a synthetic profile and the reserved example.test domain; not client evidence.

### Стъпки за възпроизвеждане

- 1 Read the managed-configuration schema the application publishes and identify the key governing trust anchors, together with the value that applies when it is unset.
- 2 Deploy the application under a management profile that does not set the key — the configuration a customer administrator produces by following the documentation as written.
- 3 Place an interception proxy in the path and present a certificate chaining to a trust anchor installed on the device. In the sample the client completes the handshake and proceeds.
- 4 Observe the session traffic, confirming that the authorisation material is readable. Testing was performed against a test account on an isolated network segment; no production session was intercepted.

### Въздействие

On any network the attacker can position themselves on — a shared office wireless network, a captive portal, a compromised home router — the session is readable and modifiable. That yields the bearer token, which is reusable away from the device and away from the network, so a transient position on the network converts into persistent account access. The severity is driven less by the technical difficulty, which is real, than by the silence: the deployment most customers run is the vulnerable one, and neither the customer nor the vendor has any signal that pinning is not in effect.

### Препоръка

- 1 Change the default. When the managed-configuration key is absent the application should enforce pinning, so that the configuration a customer gets by not acting is the safe one.

- 2 Surface the state. If a profile does disable pinning, the application should say so in a place an administrator will see, and refuse to handle credentials while a user-installed anchor is in the chain.
- 3 Document the key explicitly in the deployment guidance, with the security consequence stated plainly rather than described as a compatibility option.
- 4 Make the token less valuable on its own: shorten its lifetime, bind it to the device or installation where the platform allows, and require re-authentication for sensitive operations rather than treating a valid bearer token as sufficient.

HIGH

**F-04** Embedded local service discloses its own authorisation token

| КРИТИЧНОСТ | CVSS v3.1 (ИЛЮСТРАТИВНО)                | ЗАСЕГНАТ КОМПОНЕНТ                                                      | MASVS    |
|------------|-----------------------------------------|-------------------------------------------------------------------------|----------|
| High       | 7.1 AV:L/AC:L/PR:L/UI:N/S:U/C:H/I:L/A:N | Android and Windows builds — loopback bridge for the embedded web layer | PLATFORM |

Class: Exposure of sensitive information to an unauthorised actor (CWE-200) · missing authentication for a critical function (CWE-306) · Reference: OWASP MASVS-PLATFORM-1 and MASVS-PLATFORM-3 · MASTG testing of platform interaction and local IPC · OWASP Mobile Top 10 M4 (insufficient input/output validation) · Status: Open · Composite element: hybrid application with a native-to-web bridge

**Описание**

The hybrid architecture runs a small HTTP service on the loopback interface so the embedded web layer can reach native capability. The service is protected by a bearer token, which is the correct instinct. The defect is that one of its own endpoints returns that token to any caller, without requiring it.

A loopback port is not a privacy boundary on a mobile device: any installed application can connect to it, with no special permission and no user prompt. So the sequence available to an unprivileged application on the same device is to connect, ask the bridge for its configuration, receive the token, and then use the token to reach everything the bridge exposes — which is the authenticated user's data, through a component that was never designed to be an external interface.

## Доказателство

## Embedded local service discloses its own authorisation token

SYNTHETIC EXAMPLE

A loopback helper the app runs for its web layer returns the token that protects it, to any unauthenticated caller on the device.

```
analyst@wkstn ~ bash

analyst@wkstn:~/mobile$ adb shell netstat -tlnp | grep 127.0.0.1
Synthetic transcript — invented package, port and token
tcp 127.0.0.1:18642 LISTEN com.example.haldon

# any app on the device can reach a loopback port
analyst@wkstn:~/mobile$ curl -s http://127.0.0.1:18642/bridge/config

{"version":"3.2.1","bridgeToken":"brg_c41f...<truncated - fabricated value>"}

# that token is the only thing guarding the bridge
analyst@wkstn:~/mobile$ curl -s -H 'X-Bridge-Token: brg_c41f...<truncated - fabricated value>' \
http://127.0.0.1:18642/bridge/contacts?limit=1

200 - authorised; response discarded unread

! The endpoint that hands out the token needs the same authorisation as the data behind it.
```

i Illustrative example — synthetic data, not client evidence.

Evidence for F-04 — the loopback service is enumerated, its configuration endpoint returns the bridge token unauthenticated, and the token then authorises a data call. Illustrative reconstruction using a synthetic port and fabricated token; not client evidence.

## Стъпки за възпроизвеждане

- 1 Enumerate listening sockets on the device and identify the loopback service, confirming which application owns it.
- 2 Request the service's configuration endpoint with no credential. In the sample the response carries the bridge token in full.
- 3 Replay a data request using the token just obtained. The sample shows **200** — the request is authorised.
- 4 Stop at the status code: the response body was discarded unread, and the interaction was performed from a purpose-built test helper against a test account.

## Въздействие

Any application the user installs — including one with no permissions of consequence — can read the data this bridge fronts, for as long as the target application is running. The user is given no prompt, because from the platform's point of view nothing requiring consent has happened. On the Windows build the same loopback service is reachable by any process in the user's session, which widens the caller population further. The practical effect is that the application's own data boundary is delegated to a token it publishes.

## Препоръка

- 1 Stop returning the token. The configuration endpoint should expose the version and capability information the web layer needs and never the credential that protects the bridge.
- 2 Require authorisation on every endpoint of the bridge, including the ones that appear to be metadata. An endpoint that can be reached without the token must not disclose anything the token protects.
- 3 Bind the token to the caller rather than to the port: generate it per application launch, pass it to the web layer through the native channel that already exists, and reject any caller that cannot demonstrate it came from that channel.

- 4 Prefer the platform's own native-to-web bridge over a loopback HTTP service where one is available, since it carries an identity the operating system enforces and no port that other applications can reach.

MEDIUM

**F-05** Root and jailbreak detection defeated at a single decision point

| КРИТИЧНОСТ | CVSS v3.1 (ИЛЮСТРАТИВНО)                | ЗАЧЕГНАТ КОМПОНЕНТ                               | MASVS      |
|------------|-----------------------------------------|--------------------------------------------------|------------|
| Medium     | 5.3 AV:P/AC:L/PR:L/UI:N/S:U/C:L/I:L/A:N | Android and iOS builds — device integrity checks | RESILIENCE |

Class: Reliance on a client-side security control (CWE-602) · insufficient anti-tampering protection · Reference: OWASP MASVS-RESILIENCE-1 and MASVS-RESILIENCE-3 · MASTG resiliency testing · OWASP Mobile Top 10 M7 (insufficient binary protections) · Status: Open · Composite element: application with device-integrity gating

**Описание**

The application performs several independent integrity checks and then funnels all of them into one method that returns a single boolean. Everything downstream — whether the application starts, whether the offline store unlocks, whether certain features appear — reads that one value.

That design makes the mechanism exactly as strong as its weakest point and no stronger, because the checks themselves never have to be defeated. Overriding one return value at runtime disables the entire mechanism at once. We record this as Medium rather than High deliberately. A resilience control of this kind is not a security boundary and is not expected to stop a determined attacker with the device in hand; it raises cost. The finding is that the cost it raises here is close to zero, not that a bypass was possible at all.

**Доказателство**

**Root and jailbreak detection defeated at a single decision point**SYNTHETIC EXAMPLE

Every check funnels into one boolean, so one runtime hook disables the whole mechanism.

```
analyst@wkstn ~$ frida-trace -U -f com.example.haldon \
-j '*!*integrity*/isu'
Synthetic transcript — invented package and symbol names

instrumenting 3 matching methods:
  IntegrityCheck.hasSuBinary()
  IntegrityCheck.hasTestKeys()
  IntegrityCheck.isDeviceCompromised()

# the first two feed the third; the third gates the whole app
isDeviceCompromised() -> true (rooted test device)

# one return-value override applied at that single method
isDeviceCompromised() -> false
app continues to the authenticated session

The hook script is withheld. What matters is the single point of failure, not the technique.
```

Illustrative example — synthetic data, not client evidence.

Evidence for F-05 — tracing shows three checks collapsing into one decision method, and a single return-value override carries the application into the authenticated session. The instrumentation script is deliberately not published. Illustrative reconstruction using synthetic symbol names; not client evidence.

**Стъпки за възпроизвеждане**

- 1 Run the application on a device in a state the checks are meant to reject and confirm it refuses to proceed.

- 2 Trace the integrity-related methods to establish the relationship between them. In the sample, two checks feed a third, and the third is what the rest of the application consults.
- 3 Apply a single return-value override at that one method. The application proceeds to the authenticated session.
- 4 Record the structural weakness rather than the technique: the hook script is not included in this report, because the single point of failure is the finding and the method of reaching it is not.

#### Въздействие

On a device under the attacker's control, every protection that depends on this boolean is available to be turned off together, which includes the gate on the offline store examined in F-02. On its own the finding does not expose another user's data; its significance is that it removes the obstacle in front of the findings that do, and it does so in one step. For a customer relying on integrity checks as a compensating control in a risk assessment, that reliance is not supported by what the control actually provides.

#### Препоръка

- 1 Distribute the decision. Have the protections that matter consult the underlying signals independently rather than a single shared boolean, so that no one override disables everything.
- 2 Move what can be moved to the server. Where the platform offers an attested device signal that a backend can verify, a server-side decision is worth more than any client-side check, because the attacker does not control the verifier.
- 3 Make the checks inline and varied rather than centralised and uniform, and treat the result as one input to a risk decision rather than a gate that is either open or shut.
- 4 Calibrate expectations in the product's own documentation: describe these checks as raising the cost of attack, not as preventing use on a compromised device, so that customers do not record them as a control they are not.

MEDIUM

### F-06 Session token written to the device log and included in backups

| КРИТИЧНОСТ | CVSS v3.1 (ИЛЮСТРАТИВНО)                | ЗАСЕГНАТ КОМПОНЕНТ                                          | MASVS             |
|------------|-----------------------------------------|-------------------------------------------------------------|-------------------|
| Medium     | 6.2 AV:P/AC:L/PR:N/UI:N/S:U/C:H/I:N/A:N | Android build — diagnostic logging and backup configuration | STORAGE · PRIVACY |

Class: Insertion of sensitive information into log file (CWE-532) · exposure through backup (CWE-530) · Reference: OWASP MASVS-STORAGE-2 and MASVS-PRIVACY-1 · MASTG testing for sensitive data in logs · OWASP Mobile Top 10 M9 (insecure data storage) · Status: Open · Composite element: release build retaining development diagnostics

#### Описание

A diagnostic log line written during token refresh records the bearer token itself, and it is present in the release build rather than only in debug. On its own that would be a contained problem on modern platforms, where application logs are not readable by other applications.

What makes it material is the second half: the application's manifest still permits platform backup, so the log is copied into backup archives that leave the device and are not encrypted in the same way the device is. The token therefore outlives both the session and the device, and ends up in a location — a user's workstation, a backup service — where neither the vendor's nor the customer's mobile controls apply.

## Доказателство

## Session token written to the device log and included in backups

SYNTHETIC EXAMPLE

A diagnostic log line left enabled in the release build records the bearer token, which the platform backup then copies off the device.

```
analyst@wkstn ~ bash

analyst@wkstn:~/mobile$ adb logcat -d | grep -i 'session'
Synthetic transcript — invented package, tag and token
I/Haldon.Net refresh ok, bearer=eyJ...<truncated — fabricated value>

# the release manifest still allows platform backup
android:allowBackup = true

analyst@wkstn:~/mobile$ adb backup -f out.ab com.example.haldon
analyst@wkstn:~/mobile$ grep -c 'bearer=' out.ab.unpacked/log/app.log

14 token occurrences recovered from an unencrypted backup

Tokens in logs outlive the session they belong to and travel wherever the log travels.
```

Illustrative example — synthetic data, not client evidence.

Evidence for F-06 — the token appears in the device log, backup is permitted by the manifest, and the token is recovered from the resulting archive. Illustrative reconstruction using a fabricated token; not client evidence.

## Стъпки за възпроизвеждане

- 1 Read the device log while the application refreshes its session and identify the line recording the token.
- 2 Confirm the release manifest permits platform backup.
- 3 Take a backup of the application through the platform's own tooling and unpack it.
- 4 Count the occurrences of the token within the extracted log. Testing used a test account's token throughout; no production session material was handled.

## Въздействие

A bearer token in a log is valid until it expires, wherever the log travels. The backup path takes it off the device by design and places it somewhere with a different and generally weaker set of controls, where it may persist far longer than the session. Anyone with access to a user's workstation or backup store holds the session without needing the device, the password or the second factor. The combination is what matters: either the logging or the backup setting alone would be a lesser finding.

## Препоръка

- 1 Remove credential material from logging entirely, in every build configuration. Log the fact of a refresh and its outcome, never the token.
- 2 Disable platform backup for the application, or define an explicit backup rule set that excludes logs, caches and any store holding credential material.
- 3 Add a build-time check that fails the release when logging calls reference credential-shaped values, so that a debugging line cannot reach production again.
- 4 Shorten token lifetime and support server-side revocation, so that a token recovered from an old archive has a much smaller window in which it is worth anything.

## MEDIUM

## F-07 Exported component performs a state change without authentication

| КРИТИЧНОСТ | CVSS v3.1 (ИЛЮСТРАТИВНО)                | ЗАСЕГНАТ КОМПОНЕНТ                                  | MASVS    |
|------------|-----------------------------------------|-----------------------------------------------------|----------|
| Medium     | 5.9 AV:L/AC:L/PR:L/UI:N/S:U/C:N/I:H/A:N | Android build — deep-link handler for quick actions | PLATFORM |

Class: Missing authentication for critical function (CWE-306) · improper export of application component (CWE-926) · Reference: OWASP MASVS-PLATFORM-1 · MASTG testing of app permissions and IPC · OWASP Mobile Top 10 M4 (insufficient input/output validation) · Status: Open · Composite element: application offering deep-linked shortcuts

## Описание

A handler that implements shortcut actions is exported, which is necessary for the shortcuts to work from outside the application. It acts on the currently signed-in user's session, which is also reasonable. What is missing is any check on who invoked it.

Because the component is exported, any application on the device can invoke it with parameters of its choosing, and the action is carried out under the signed-in user's authority with no prompt and no re-authentication. The read paths on the same component are gated correctly, which places this clearly as an oversight on the write path rather than an absent model: the application knows how to check, and does not check here.

## Доказателство

**Exported component performs a state change without authentication**  
A deep-link handler exported to any application on the device acts on the signed-in user without checking who called it.

**SYNTHETIC EXAMPLE**

```
analyst@wkstn ~$ aapt dump xmltree haldon-release.apk \\  
AndroidManifest.xml | grep -A2 'activity-alias'  
Synthetic transcript — invented package and component  
  
activity .ui.QuickActionActivity  
  android:exported = true  
  scheme = "haldon" host = "action"  
  
# invoked from an unprivileged helper app on the same device  
analyst@wkstn ~$ am start -a android.intent.action.VIEW \\  
-d 'haldon://action/reassign?record=1042&owner=carol'  
  
Status: ok  
record 1042 reassigned — no prompt, no re-authentication  
  
Performed against a synthetic record in an isolated test tenant.
```

Illustrative example — synthetic data, not client evidence.

Evidence for F-07 — the manifest shows the component exported, and an invocation from an unprivileged helper application changes record ownership without a prompt. Performed against a synthetic record in an isolated test tenant. Illustrative reconstruction; not client evidence.

## Стъпки за възпроизвеждане

- 1 Inspect the manifest of the release package and identify components marked as exported, together with the URI scheme and host they respond to.
- 2 Build an unprivileged helper application that requests no permissions of consequence.
- 3 Invoke the exported component from that helper with parameters naming a record and a new owner. In the sample the action completes with no prompt and no re-authentication.

- 4 Verify the state change and then revert it. The record was synthetic and the tenant was an isolated test tenant created for the engagement.

#### Въздействие

Any installed application can take actions as the signed-in user, which is an integrity problem rather than a confidentiality one: records can be reassigned, modified or actioned without the user's knowledge. On a device managed by a customer organisation the attacking application need not be malicious in any obvious way — a compromised utility, or an application with an aggressive advertising library, is sufficient. Because the action is performed through the legitimate application, server-side logs attribute it to the user, which complicates any subsequent investigation.

#### Препоръка

- 1 Require an explicit user confirmation in the application for any state-changing action arrived at through an external invocation, so the authority is the user's rather than the caller's.
- 2 Verify the caller where the platform permits it, and treat every parameter arriving from an external invocation as untrusted input to be validated against the session, not as an instruction.
- 3 Export only what has to be exported, and split read-only entry points from state-changing ones so the exported surface carries the smallest possible capability.
- 4 Re-check the authorisation server-side as well. An action that changes ownership should be authorised by the backend on its own terms rather than trusted because it arrived from an authenticated client.

LOW

#### F-08 Biometric prompt gates the interface rather than the credential

| КРИТИЧНОСТ | CVSS v3.1 (ИЛЮСТРАТИВНО)                | ЗАЧЕГНАТ КОМПОНЕНТ                        | MASVS |
|------------|-----------------------------------------|-------------------------------------------|-------|
| Low        | 3.9 AV:P/AC:H/PR:L/UI:N/S:U/C:L/I:N/A:N | Android and iOS builds — biometric unlock | AUTH  |

Class: Reliance on a client-side security control (CWE-602) · improper authentication (CWE-287) · Reference: OWASP MASVS-AUTH-2 and MASVS-AUTH-3 · MASTG testing of local authentication · OWASP Mobile Top 10 M3 (insecure authentication) · Status: Open · Composite element: application offering biometric re-entry

#### Описание

The application offers a biometric unlock on re-entry. The prompt is presented correctly and the platform API is used as intended, but the result is consumed as a boolean that the application then acts on: success reveals the interface, failure hides it. The stored credential is not bound to that decision — the key is created without a user-authentication requirement, so it is released whenever the application asks for it.

The practical consequence is that the prompt is a user-interface state rather than a cryptographic gate. An attacker who can influence the application's own execution — which, given F-05, is not a high bar on a device they control — obtains the credential without satisfying the prompt. We rate this Low because reaching it requires physical possession of an unlocked or compromised device, and because the credential it yields is already protected by the server-side session model.

## Доказателство

## Biometric prompt gates the interface rather than the credential

SYNTHETIC EXAMPLE

The unlock decision is a boolean the application acts on, so the stored credential is released whether or not the prompt succeeds.

```
analyst@wkstn ~$ frida-trace -U -f com.example.haldon -j '*!*biometric*/isu'
Synthetic transcript — invented package and symbol names

BiometricGate.onAuthSucceeded() -> unlocks the view
CredentialStore.load()          no auth binding

# the key is requested without a user-authentication requirement
KeyGenParameterSpec.setUserAuthenticationRequired( false )

# so the stored credential loads with the prompt dismissed
CredentialStore.load() -> returned

A biometric gate is only a gate when the key itself is released by it.
```

Illustrative example — synthetic data, not client evidence.

Evidence for F-08 — tracing shows the unlock callback gating the view while the credential store loads independently, with the key created without an authentication requirement. Illustrative reconstruction using synthetic symbol names; not client evidence.

## Стъпки за възпроизвеждане

- 1 Trace the biometric callback and the credential-loading path and establish whether any data dependency exists between them.
- 2 Inspect the key's generation parameters to determine whether release is bound to user authentication. In the sample it is not.
- 3 Load the stored credential with the prompt dismissed rather than satisfied, confirming the two are independent.
- 4 Record the result against a test account. No production credential was involved at any point.

## Въздействие

Someone holding the device can reach the stored credential without presenting the user's biometric, which defeats the expectation the feature sets. The practical exposure is bounded — physical access is required, and the credential is still subject to server-side session controls — but the feature is relied on by users and described in the product as protecting exactly this case, and it does not.

## Препоръка

- 1 Bind the key to user authentication at creation, so that the platform releases it only after a successful biometric or device-credential check and the gate is enforced by the operating system rather than by application logic.
- 2 Use the authenticated cryptographic object the platform provides, so that the result of the prompt is the ability to perform an operation rather than a boolean the application interprets.
- 3 Re-key on biometric enrolment change, so that adding a new biometric does not silently grant access to credentials stored under the previous set.
- 4 Describe the feature accurately in the product documentation: convenience re-entry and a cryptographic gate are different promises, and customers make risk decisions from the description.

## INFORMATIONAL

## F-09 Obfuscation applied, but not across the security-relevant code

| КРИТИЧНОСТ    | CVSS v3.1 (ИЛЮСТРАТИВНО)                  | ЗАЩЕГНАТ КОМПОНЕНТ                          | MASVS      |
|---------------|-------------------------------------------|---------------------------------------------|------------|
| Informational | n/a не е уязвимост; не е присвоена оценка | All platforms — release build configuration | RESILIENCE |

Class: Observation · insufficient binary protection coverage · Reference: OWASP MASVS-RESILIENCE-2 · MASTG resiliency testing · OWASP Mobile Top 10 M7 (insufficient binary protections) · Status: Open · Composite element: release build with name obfuscation enabled

## Описание

Name obfuscation is enabled and covers the great majority of the application. The remainder, however, is concentrated in precisely the classes an attacker reads first: the credential store, the integrity checks, the bridge token provider and the offline key derivation all retain readable names, almost certainly because they are referenced reflectively or from the web layer and were added to a keep list to stop the build breaking.

This is recorded as informational, not as a vulnerability. Obfuscation is not a security boundary and no finding in this report depends on its absence — the credential in F-01 would have been recoverable from an entirely obfuscated build, because strings are not names. It is included because it tells the team something useful: the effort spent on obfuscation is not landing where it was intended to, and a keep list is quietly documenting the sensitive parts of the application.

## Доказателство

**Obfuscation applied, but not across the security-relevant code** SYNTHETIC EXAMPLE

Name obfuscation covers most of the application while the classes handling keys, tokens and integrity keep readable names.

```
analyst@wkstn ~$ jadx -d out haldon-release.apk && cd out
Synthetic transcript — invented package and class names

classes total      4,118
obfuscated names   3,977  96.6%
readable names     141

# the readable remainder is the part worth reading
CredentialStore
IntegrityCheck
BridgeTokenProvider
OfflineDbKeyDeriver

Obfuscation raises the cost of reading code. It is not a boundary, and it did not protect these paths.
```

i Illustrative example — synthetic data, not client evidence.

Evidence for F-09 — a count of obfuscated against readable class names in the release build, with the readable remainder listed. Illustrative reconstruction using synthetic class names; not client evidence.

## Как беше установено това

- 1 Decompile the release build and count obfuscated against readable type names to establish overall coverage.
- 2 List the readable remainder and classify it by what the classes do.
- 3 Compare that list against the project's keep rules to confirm the cause.
- 4 Record as an observation. No exploitation is involved and none is implied.

### Въздействие

No direct impact, and no finding in this report relies on it. The effect is on cost: a reader looking for the security-relevant parts of the application is directed straight to them, which lowers the effort required to find everything else in this report. Treat it as a signal about where the build configuration and the threat model have drifted apart rather than as a defect to be fixed in its own right.

### Какво да направите с това наблюдение

- 1 Narrow the keep rules to the specific members that genuinely need stable names, rather than whole classes, and re-check coverage over the security-relevant packages after each change.
- 2 Do not treat the result as a control. Obfuscation raises the cost of reading code and nothing else; it should never appear in a risk assessment as a mitigation for a secret the artefact contains.
- 3 Fix the underlying issues first. Once F-01 and F-02 are addressed there is far less in the binary worth reading, which is a better outcome than making it harder to read.

Напомняне за този раздел. F-01 до F-09 са взети от формата на няколко анонимизирани ангажмента и представени, сякаш идват от един. Всеки запис, име на пакет, хост, идентификатор, акаунт и оценка по-горе са синтетични.

## РАЗДЕЛ 7

## План за отстраняване

Същите констатации, подредени като работа. Групирани според кога трябва да се случи, а не по критичност, защото някои от най-спешните действия не са промени в кода.

Вълна 1 — сега, преди каквато и да е промяна в кода

Тези действия намаляват изложеността незабавно и не чакат цикъл на издаване.

| Действие                                                                                                                                    | Констатация | Отговорник             |
|---------------------------------------------------------------------------------------------------------------------------------------------|-------------|------------------------|
| Подменете клиентската тайна към сървърната част и прегледайте журналите за достъп до API за употреба извън очакваната съвкупност от клиенти | F-01        | Платформа / API        |
| Планирайте изданието на клиента, което подмяната налага, и решете как се обслужват клиентите със стари билдове                              | F-01        | Продукт                |
| Документирайте конфигурационния ключ за обвързването със сертификат за администраторите на клиента, със заявено последствие за сигурността  | F-03        | Продукт / Документация |

Вълна 2 — следващото издание

Промени, които премахват констатациите, вместо да намаляват обхвата им, и които трябва да излязат заедно, защото няколко от тях докосват едни и същи пътища в кода.

| Действие                                                                                                                                           | Констатация | Отговорник |
|----------------------------------------------------------------------------------------------------------------------------------------------------|-------------|------------|
| Премахнете споделената тайна от артефакта; удостоверявайте се като потребителя с краткотраен токен                                                 | F-01        | Мобилни    |
| Преместете ключа на офлайн базата данни в хранилище, подкрепено от хардуер, и обвържете освобождаването му с удостоверяването на потребителя       | F-02, F-08  | Мобилни    |
| Обърнете подразбирането за обвързването със сертификат и отказвайте работа с данни за достъп под инсталирано от потребителя доверено удостоверение | F-03        | Мобилни    |
| Спрете конфигурационната крайна точка на моста да връща токена; изисквайте оторизация на всяка крайна точка на моста                               | F-04        | Мобилни    |
| Премахнете материала с данни за достъп от журналите и изключете резервното копиране на платформата за приложението                                 | F-06        | Мобилни    |
| Изисквайте потвърждение в приложението за промени в състоянието, пристигащи през изнесени компоненти                                               | F-07        | Мобилни    |

### Вълна 3 — структурна, през следващите цикли

Работа, която променя как се предотвратява следващият клас констатации, вместо да поправя този. Нищо от нея не е спешно; всичко от нея намалява колко често този отчет трябва да бъде писан.

| Действие                                                                                                                                          | Констатация | Отговорник        |
|---------------------------------------------------------------------------------------------------------------------------------------------------|-------------|-------------------|
| Разпределете решението за цялост, така че едно единствено отменяне да не изключва всяка защита                                                    | F-05        | Мобилни           |
| Добавете проверка при издаване, която сканира построения артефакт за константи с формата на данни за достъп                                       | F-01, F-06  | Билд / CI         |
| Стеснете правилата за запазване при обфускацията и проверете отново покритието върху кода, свързан със сигурността                                | F-09        | Мобилни / Билд    |
| Проверявайте оторизацията отново от страна на сървъра за действия, които клиентът може да инициира, вместо да се доверявате на удостоверен клиент | F-07        | Платформа / API   |
| Намалете какво се кешира офлайн и го правете невалидно по-рано                                                                                    | F-02        | Продукт / Мобилни |

Бележка за последователността. F-02 и F-08 са изброени заедно умишлено: обвързването на освобождаването на ключа с удостоверяването на потребителя поправя биометричната защита и укрепва офлайн хранилището с една и съща промяна. Изпълнението им поотделно означава хранилището за данни за достъп да бъде докоснато два пъти и от първото преминаване да се получи по-малко.

## РАЗДЕЛ 8

## Повторен тест и приключване

Какво се случва след поправките и какво писмото за приключване казва и не казва.

### Какво покрива повторният тест

Всяка констатация в този отчет се тества повторно срещу билд, който вие посочите, със същия метод, който я е открил. Там, където една поправка променя подхода — данни за достъп, преместени в хранилището за ключове, не се проверяват така, както данни за достъп в пакет — повторният тест използва каквото установява, че поправката действително издържа, а писмото отразява метода.

Повторният тест е включен като стандартна част от услугата. Той не е втори ангажимент и не преопределя обхвата: отговаря на единствения въпрос дали всяка констатация е затворена.

### Какво заявява писмото за приключване

Писмото назовава всяка констатация и я отразява като затворена, частично адресирана или отворена, с едно изречение за това какво е било проверено. То е написано, за да бъде показано на клиент или оценител.

То не заявява, че приложението е сигурно, не го сертифицира спрямо каквото и да е стандарт и не изразява мнение за нищо извън изброените констатации. Ние сме фирма за тестване, не одитор или сертифициращ орган, и писмото го казва с тези думи.

### Какво означава затворена констатация за всяка от тези

| Констатация | Какво проверяваме при повторния тест                                                                               |
|-------------|--------------------------------------------------------------------------------------------------------------------|
| F-01        | Артефактът не носи данни за достъп; старата тайна вече не се удостоверява.                                         |
| F-02        | Ключът не може да бъде възпроизведен извън устройството от материал, четим от устройството.                        |
| F-03        | Профил, който не задава ключа, налага обвързване със сертификат.                                                   |
| F-04        | Нито една крайна точка на моста не разкрива токена; всички го изискват.                                            |
| F-05        | Нито едно единствено отменяне не изключва защитите наведнъж.                                                       |
| F-06        | Няма материал с данни за достъп в журналите; резервното копие изключва журналите и хранилищата за данни за достъп. |
| F-07        | Външното извикване на промяна в състоянието изисква потвърждение в приложението.                                   |
| F-08        | Съхранените данни за достъп не могат да бъдат заредени без изпълнение на подканата.                                |
| F-09        | Покритие върху класовете, свързани със сигурността, отчетено като число.                                           |

## ПРИЛОЖЕНИЕ А

## Билдове, акаунти и инструменти

Какво поискахме, преди тестването да започне, и защо всяка точка имаше значение.

|                          |                                                                                                                                                                                                                                                                                                        |
|--------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Артефакти от изданието   | Подписани издателски билдове за всяка платформа в обхвата, съответстващи на публикуваното. Билдове за отстраняване на грешки биха произвели констатации, които не описват това, което клиентите използват.                                                                                             |
| Тестови акаунти          | По два акаунта за всяка роля, в изолиран клиентска организация. Един акаунт за всяка роля не може да демонстрира проблем между акаунти; два могат.                                                                                                                                                     |
| Управляван профил        | Профилът за внедряване, който се дава на клиентите. F-03 съществува заради настройка по подразбиране в него и не би била открита без него.                                                                                                                                                             |
| Сървърна среда           | Непроизводствена среда, съответстваща на производствената конфигурация, така че поведението от страна на сървъра да може да бъде тествано, без да бъдат докосвани клиентски данни.                                                                                                                     |
| Посочено лице за контакт | Инженер, който можеше да отговори на въпрос за предназначението. Няколко часа обратно инженерство рутинно се спестяват от един отговор за какво служи даден компонент.                                                                                                                                 |
| Инструменти              | Стандартни публични мобилни инструменти за разпакетиране, декомпилиране, инструментиране по време на работа и прихващане на трафик, заедно с ръчен анализ. Нито един инструмент не е назован като принадлежащ на клиент и изходът на търговски инструмент не е възпроизведен в този примерен документ. |

## ПРИЛОЖЕНИЕ Б

## Извън обхвата и какво този тест не е

---

Заявено изрично, защото стойността на един отчет зависи от това да се знае какво той не покрива.

- Не е одит или сертификация. PXLSecurity LTD. е фирма за офанзивно тестване по сигурността. Нищо тук не сертифицира приложението спрямо MASVS, оценка по MASA или каквато и да е друга схема и от този документ не може да бъде изведено такова твърдение.
- Не е преглед на изходния код. Артефактът беше третиран както би го третираше нападател. Прегледът на кода за сигурност е различен ангажимент и намира различни неща.
- Не е тест на операционните системи. Уязвимостите на платформата, фърмуерът на устройствата и самите магазини за приложения бяха извън обхвата.
- Не е тест на управлението на устройствата на клиента. Използвахме профил за управление, за да възпроизведем внедряване; не тествахме платформата за управление.
- Без отказ на услуга и без техника, чийто режим на провал е счупено устройство или повредено хранилище за данни.
- Не е пълен тест на сървърната част. API-то беше тествано така, както приложението го достига. Пълният тест за проникване в API покрива крайни точки и роли, които мобилният клиент никога не докосва.
- Не е оценка на поверителността или защитата на данните. F-06 докосва това, което напуска устройството, но оценка по GDPR или на потоците от данни е отделна работа с различен метод.

## ПРИЛОЖЕНИЕ В

## Препратки и какво представляват

Описано точно. Позоваването на стандарт не е съответствие с него.

|                           |                                                                                                                                                                                                                                                                                             |
|---------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| OWASP MASVS               | Mobile Application Security Verification Standard: публичен списък от изисквания за сигурност на мобилни приложения, организиран в групи контроли. Структурираме тестването и отчитането около него. Той е стандарт, спрямо който се тества, не сертификация, която притежаваме или даваме. |
| OWASP MASTG               | Mobile Application Security Testing Guide: съпътстващото ръководство, описващо как се тества всяко изискване. То определя метода, не критичността.                                                                                                                                          |
| OWASP Mobile Top 10       | Документ за осведоменост, изброяващ най-честите категории мобилен риск. Полезен за предаване на формата на една констатация на неспециалист; не е план за тестване.                                                                                                                         |
| OWASP API Security Top 10 | Използва се за сървърната страна на ангажимента, където API-то на приложението е в обхвата.                                                                                                                                                                                                 |
| CWE                       | Общ речник за видовете слабости, използван в реда „Клас“ на всяка констатация, за да могат констатациите да бъдат групирани и проследявани.                                                                                                                                                 |
| CVSS v3.1                 | Система за оценяване на характеристиките на уязвимостите. Дава се за всяка оценена констатация като входни данни за приоритизирането. Раздел 4 обяснява защо тя не определя нашата оценка.                                                                                                  |

## ПРИЛОЖЕНИЕ Г

## Как да четете този отчет

Кой какво да прочете и какво да направи с него.

| Ако сте...                     | Прочетете                   | И направете                                                                                      |
|--------------------------------|-----------------------------|--------------------------------------------------------------------------------------------------|
| Ръководител или член на борда  | Раздел 1                    | Осигурете средства за вълна 1 днес. Това са три действия и нито едно от тях не е промяна в кода. |
| Продуктов отговорник           | Раздели 1, 5 и 7            | Решете как се обслужват клиентите със стари билдове, щом данните за достъп бъдат подменени.      |
| Мобилен инженер                | Раздели 4 и 6               | Работете по списъка с препоръки; F-02 и F-08 са една промяна.                                    |
| Инженер по платформата или API | F-01, F-04 и F-07           | Проверете отново какво свървърт се доверява, че клиентът е решил.                                |
| Отговорник по сигурността      | Раздели 2, 3 и 8            | Проверете дали допусканията за обхвата все още са в сила, после запазете повторния тест.         |
| Одитор или оценител            | Раздели 2, 4 и Приложение Б | Отбележете какво е и какво не е този документ; той е доказателство за тестване, не сертификация. |

Мобилното приложение е единствената част от вашата система, която подавате на нападателя. Всичко в този отчет следва от това.

За този документ. Съставен примерен документ, синтетични данни, без действителен клиент. PXLSecurity LTD., София, България · VAT BG203239331 · hello@pxlsec.com · pxlsec.com. Скалата ни за критичност, правилата на ангажимента и политиката за координирано разкриване са публикувани на pxlsec.com, а услугата за сигурност на мобилни приложения, която този отчет илюстрира, е на pxlsec.com/bg/services/mobile-application-security.