

**MEDIA APP VULNERABILITIES EXPOSED**

93% of Android OTT  
Media Apps Are Not  
Ready for New  
Security Mandates

July 2020

# Table of Contents

**03**

**Introduction**

---

**04**

**Methodology**

---

**05**

**Ranking**

---

**10**

**What's At Risk**

---

**13**

**Important  
Lessons**

---

**16**

**Conclusion**

---



## Introduction

Verimatrix reported in April 2020 that content producers were starting to mandate protection of Over The Top (OTT) Video applications.<sup>1</sup> These apps allow consumers to stream video content – often high value – to their mobile phones and tables.

With this new mandate coming, Verimatrix decided to review the current State of Video App Security to assess the readiness of the market.


**It was discovered that 93% of apps did not meet the expected standards.**

This report is relevant to CISOs, CTOs, risk and security professionals, and product managers responsible for media streaming mobile applications. It discusses the risks that applications expose themselves to along with lessons and best practices to mitigate those risks.

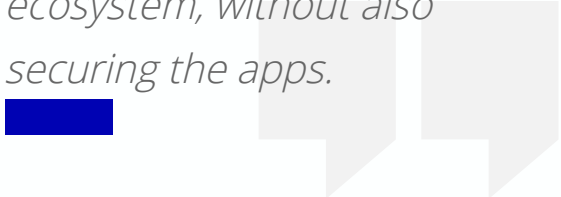
1. <https://www.verimatrix.com/blog/202004/understand-emerging-movie-studio-security-requirements-protect-your-ott-app>



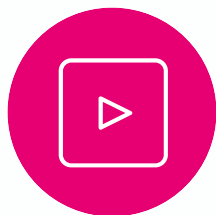
## Major Hollywood studios agree...



*The acceleration in uptake of content streaming this year reinforces the need for studios, affiliates and partners to protect their revenue stream. Content protection solutions like DRM have long been advocated but revenue protection increasingly means going further. The apps being deployed by streaming services are as much a part of the ecosystems as the servers they connect to. You can't secure the ecosystem, without also securing the apps.*



## Methodology



To better understand the current state of security in video apps, Verimatrix selected 14 popular applications and utilized its in-house security lab to analyze them for common security weaknesses.



*The complexity of modern mobile apps means that there is a “guaranteed vulnerability” in their code*

### To create a level playing field

all the apps selected were for the Android platform. They were selected at random from the published set of OTT apps that target European consumers.

Each app was given a shallow dive analysis that was timeboxed to 3 hours. The analysis assessed whether or not the apps implemented a defined list of standard security measures.

To be clear, the research did not look for vulnerabilities within the app code. The complexity of modern mobile apps means that there is a “guaranteed vulnerability” in their code. The presence of standard security measures is an indication of how exploitable a vulnerability would be by an attacker.

The applications all used digital rights management (DRM) and/or a secure video player to keep the valuable video content they play safe. These components were out of scope of the analysis.

# Ranking

Once analyzed, the apps were ranked against the UL-Verimatrix scale.<sup>2</sup>

## A

Highly  
Secure

- Majority of code (including all handling sensitive data and algorithms) is developed in a language that compiles to processor native machine code (i.e. C/C++)
- Strong control flow obfuscation<sup>3</sup> of all critical code
- Strong anti-tamper<sup>4</sup> protection of the application
- Cryptography protected by whitebox<sup>5</sup> (or equivalent technology)
- No sensitive text visible in static analysis of code
- Network traffic encrypted using TLS<sup>6</sup> 1.3 and downgrade not possible
- Certificate pinning<sup>7</sup> applied to networking
- Strong device binding<sup>8</sup>

## B

Payment  
Equivalency<sup>9</sup>

- Code handling sensitive data and algorithms is developed in a language that compiles to processor native machine code (i.e. C/C++)
- Strong control flow obfuscation of all critical code
- Anti-tamper protection of the application
- Cryptography protected by whitebox (or equivalent technology)
- No sensitive text visible in static analysis of code
- Network traffic encrypted using TLS 1.3 and downgrade not possible
- Certificate pinning applied to networking
- Strong device binding

## C

Standard  
Security

- Obfuscation of all critical code
- Anti-tamper protection of the application
- No sensitive text visible in static analysis of code
- Network traffic encrypted using TLS 1.3 and downgrade not possible
- Certificate pinning applied to networking
- Strong device binding

## D

Basic  
Security

- Obfuscation of critical code
- Network traffic encrypted
- Device binding

## E

- None

2. <https://info.verimatrix.com/mobile-security-whitepaper-1>

3. Obfuscation means scrambling computer code to make it less-intelligible to a human.

4. Anti-tamper technology provides a means to ensure the code being run is the intended code.

5. Whitebox technology protects cryptographic operations and keys.

6. TLS (Transport Layer Security) is the standard encryption protocol of the internet.

7. Certificate pinning validates that the end point of communication is the intended end point.

8. Device binding is a technique to lock an application instance to a particular phone.

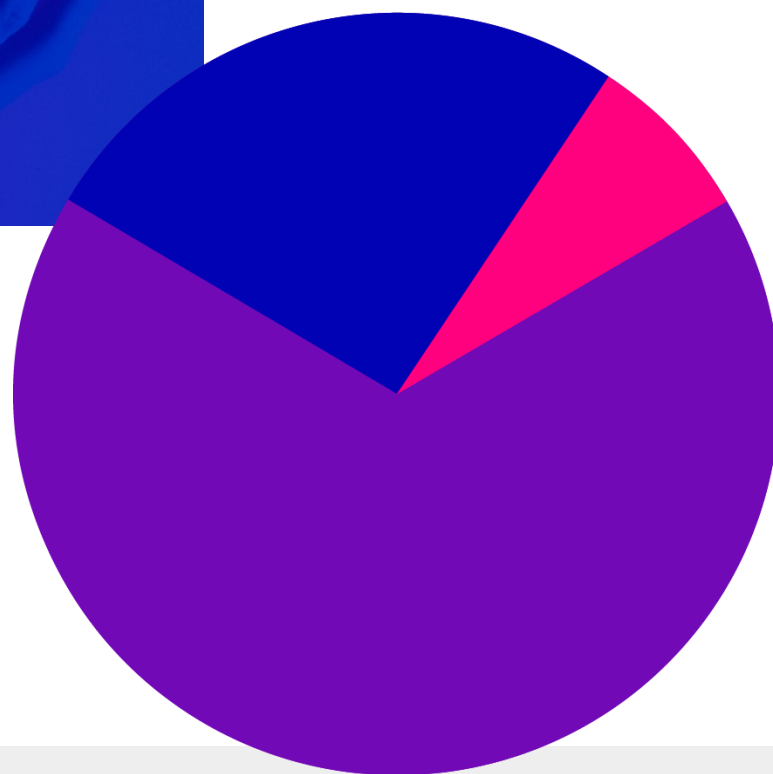
9. Based on standards from Visa and Mastercard for secure mobile payments

It is Verimatrix's expert opinion that a C ranking is required to protect a media application to a reasonable standard. Based on available information and insights, it is expected that the emerging security standards will agree with Verimatrix's assessment of applications requiring at least a C grade. Reaching this level of security will:

- Better guard individual components, such as a secure video player, by tightly binding them into the rest of the app context.
- Protect business logic and rules within the application.
- Secure the privacy and integrity of customer data.
- Avoid the mobile app being a channel to abuse back-end APIs.

***Of the applications analyzed, only 7% achieved the Baseline protection level.***

Worse, 29% did not even make use of the freely available tools that come with the Android development environment. This is a higher percentage than the general mobile app population where 29%<sup>10</sup> of apps do not apply Basic security.



A Rating:  
**0%**

B Rating:  
**0%**

C Rating:  
**7%**

D Rating:  
**64%**

E Rating:  
**29%**

## Man in the Middle



When an attacker sits undetected in the middle of a communication channel, they can monitor and modify messages being sent back and forth. This is known as a “Man in the Middle” attack.

**To avoid a Man in the Middle attack, two precautions must be taken:**

**1**

The communication channel needs to be **encrypted**.

**2**

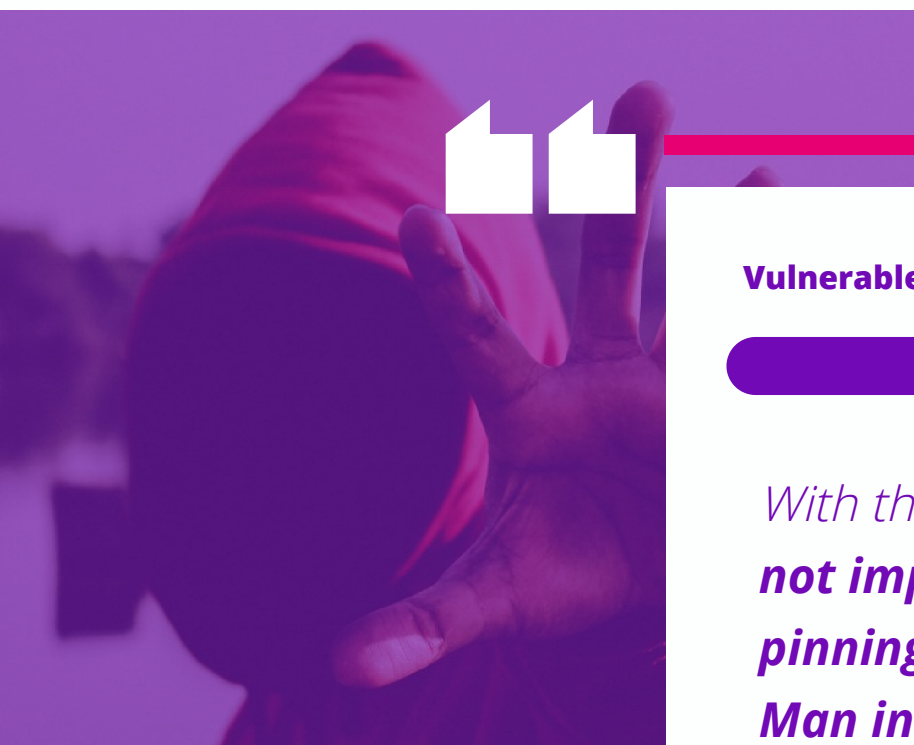
The end point of the communication needs to be **authenticated**.



## Man in the Middle

All of the sample applications used Transport Layer Security (TLS) as the protocol to encrypt their network communication. TLS is a standard encryption protocol used for most internet traffic. It was reassuring to see that all the analyzed apps were using TLS v1.2 or the very latest v1.3. This is good to see; especially as older TLS versions have been deprecated by most technology vendors due to security vulnerabilities.

As well as encrypting the channel, it is crucial to trust the end point of the communication. There is no point encrypting the data if the bad-guy is the one you have established the channel with. Validating the end point is known as “authentication”. With TLS, the established best practice is to use a technique known as certificate pinning. This is based around public key cryptography – where only the legitimate end point knows the associated private key.



**Vulnerable**

**Not Vulnerable**



*With the sample apps, **64% did not implement certificate pinning and were vulnerable to Man in the Middle attacks.** This puts the data, including user authentication credentials, at risk of being exposed.*



## App Modification

Attackers will often modify an application's code or the resources that are packaged within it. This enables them to perform deeper reverse engineering, circumvent security measures within the app, or to create "clone" or "skinned" versions of the application.

Creating a clone application is known as a repackaging attack.

For example, **71% of the applications studied contained some form of root detection.** This is designed to prevent the applications from running on a device that has its Operating System protections broken down. If an application can be modified, then an attacker can locate and remove the root detection code, thus circumventing the protection it offered.

*Being able to modify the application allows resources to be changed. Certificates for pinning network connections are often stored as application resources. If these can be changed, an attacker would be able to un-pin a pinned network connection.*

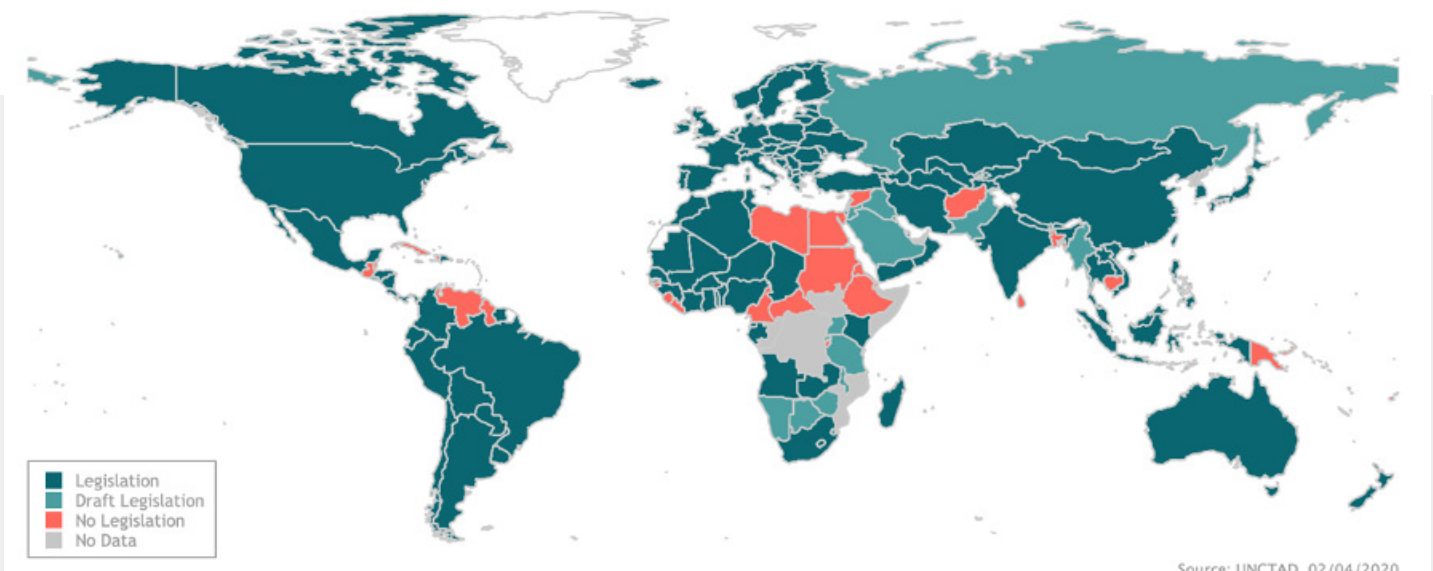
```
on() {
  manRan) { return; }
  Ran = true;
  ment.createElement('script');
  xt/javascript';
  ue;
  l + '&r=' + Math.random();
  elementsByTagName('head')[0]||document.getEle
  0; i < evts.length; i++) {
    t(evts[i], logHuman);
  }
  0; i < evts.length; i++) {
    t(i), logHuman);
  }
  fference_lh=1&hid=A957C9DCB285F0
  xscript>>
  te
```

## What's At Risk: Personal Data

The General Data Protection Regulation (“GDPR”) is part of a global regulatory trend to protect our privacy rights – which are viewed as fundamental rights in the EU. In fact, 66% of countries now have similar data protection and privacy legislation; another 10% are currently preparing for its introduction.<sup>12</sup>

**If an app is collecting, processing or storing any personal data – and let’s be honest, all apps are - then it falls into the scope of privacy regulations.** If payments are being taken through the app, then extra requirements like PCI-DSS and PSD2 come into play.

These new regulations typically require the service provider to use best practises to keep personal data safe. If the service provider fails to keep personal data safe, the immediate consequence is an obligation to report the data breach, which then becomes public information. In addition, the regulators may investigate and impose heavy financial penalties and the harmed individuals may pursue private lawsuits in some jurisdictions. When poor security practises are made public, there is nowhere to hide.



12. [https://unctad.org/en/Pages/DTL/STI\\_and ICTs/ICT4D-Legislation/eCom-Data-Protection-Laws.aspx](https://unctad.org/en/Pages/DTL/STI_and ICTs/ICT4D-Legislation/eCom-Data-Protection-Laws.aspx)

## What's At Risk: **Trusted Server Access**

No media application works in isolation — they must retrieve content from back-end systems. These systems are designed to only provide content to entitled users and trusted applications. The back-end exposes APIs – internet doors – to give apps and users access to systems, services and information. Only trusted apps and users should be allowed through these doors.

The only source of decision making for the back-end is the information it receives through the APIs. If the information can be falsified, then the

back-end will make the wrong decisions – potentially allowing untrusted or unauthorised access.

**It is increasingly seen that hackers reverse engineer applications to learn how they authenticate themselves with back-end systems. This allows the attacker to send requests to the back-end that appear to come from a trusted and legitimate application. Attackers use this approach to hack into back-end systems through their own APIs.**



## What's At Risk: **Business Models**

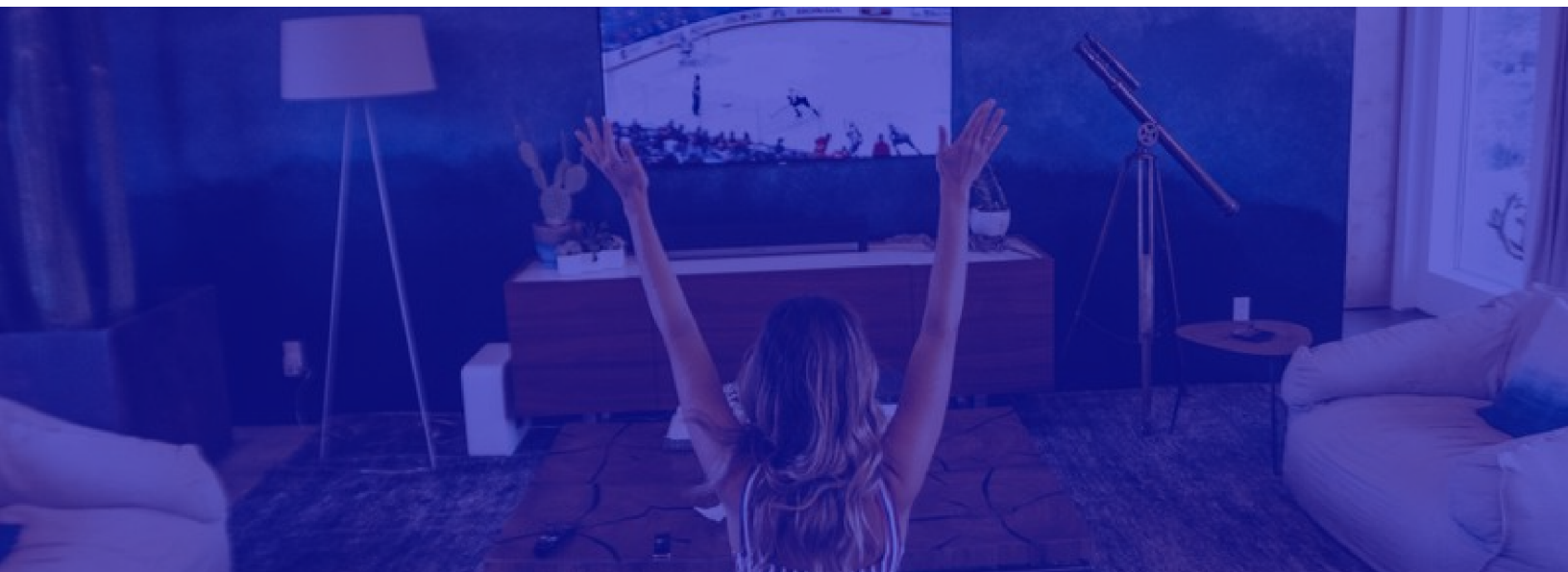
In many cases, the application is also the guardian of the service's business model – ensuring the content is viewed in the manner intended.

**Users often feel entitled to a better experience and will find ways to self-upgrade. Modifying or misusing an application is often the vehicle to achieving a better experience without paying for it.**

For example, it is common to offer a free or lower cost subscription that allows content only to be viewed on small mobile phone screens. If the user wishes to view the content on a larger screen (e.g. a TV) then there are additional costs.

There are plenty of discussions in online forums educating users how to Side Load an Android mobile app onto an Android SmartTV to circumvent those additional costs.

Another common attack on the app is to remove advertisements or to change the advert stream to one controlled by the attacker. This only works for client-side insertion, but it is common to see repackaged versions of popular media apps on third-party app stores offering “advert free viewing” or other enhancements to the viewing experience. As the adverts go, so does the revenue stream.



# Important Lessons

## New Regulations

GDPR and other privacy regulations are nothing to be scared about. Fundamentally, they require that you take sensible steps, in line with industry best practice, to protect information. When that data is within a mobile app, the best way to protect it is to protect the application.

### OWASP Mobile Top 10

OWASP publishes a set of security guidelines for anyone developing a mobile application<sup>13</sup>. These are practical steps that an application developer should take when building their solution.

Any application that is developed following these 10 steps will reach the benchmark C grade.



**M1:** Improper Platform Usage

**M2:** Insecure Data Storage

**M3:** Insecure Communication

**M4:** Insecure Authentication

**M5:** Insufficient Cryptography

**M6:** Insecure Authorization

**M7:** Client Code Quality

**M8:** Code Tampering

**M9:** Reverse Engineering

**M10:** Extraneous Functionality

13. <https://owasp.org/www-project-mobile-top-10/>

# Important Lessons

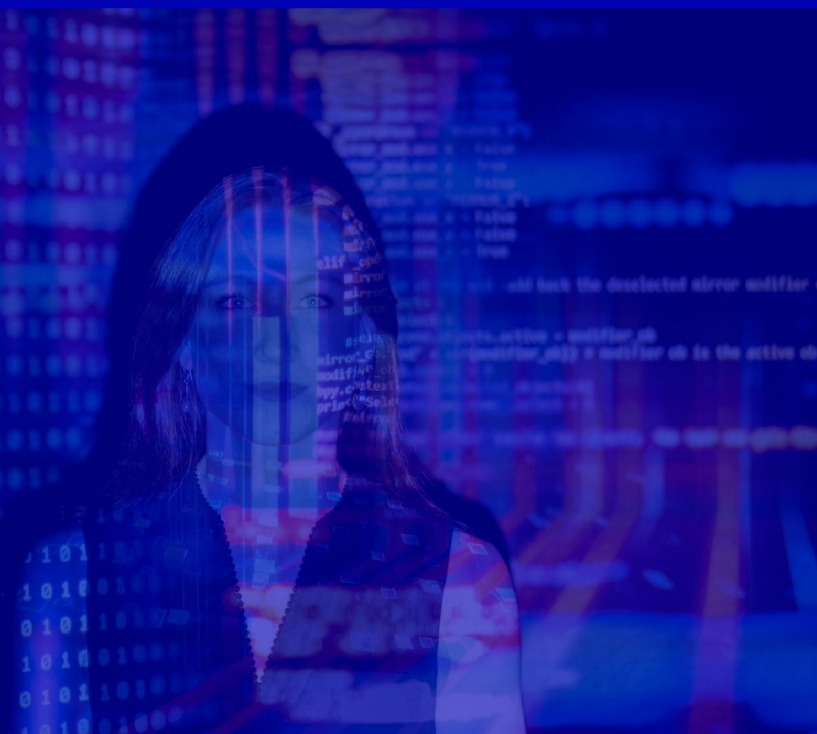
## Vulnerability Assessment

There are different levels of security evaluation that can be conducted on a mobile application – from a deep review carried out by a specialist security lab to an automated scan. While all applications can benefit from a deep review, in practical terms the size of benefit depends on the use case of the application and the sensitivity of the data being processed. That means the cost and effort of commissioning a review needs to be weighed against the functionality of the app.

*Having a good understanding of the data being processed in the application is key.*

It is clear from the analysis that all the applications would benefit strongly from Automated Vulnerability Assessment. This would quickly and easily identify risks such as Man in the Middle attacks.

It should be noted though, Automated Vulnerability Assessment is not a deep analysis. It will not find all weaknesses and typically does not analyze how easy an application is to reverse engineer and modify. This means it will check for the presence of defenses against common risks but not how easy it is for an attacker to work around them.



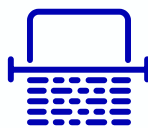
# Important Lessons

## Strong Authentication

Moving away from password-based authentication makes it much harder to spoof or replay user credentials to get unauthorized access to content and other systems. The most high-profile example of this is password sharing. This is becoming an increasing concern for streaming companies who are looking to protect revenue streams and crackdown on freeloaders. The way to do this is to make it more difficult to falsify the information being entered into the mobile app using Strong Authentication methods. If slick passwordless authentication methods are adopted, it can even enhance the user experience.

## App Shielding

App Protection Tools make applications much harder to reverse engineer by shielding their internals. This keeps attackers from learning secrets such as how to access back-end systems. The three complimentary security layers required are:



### Obfuscation

Makes code difficult to read and understand, which protects it from static analysis.



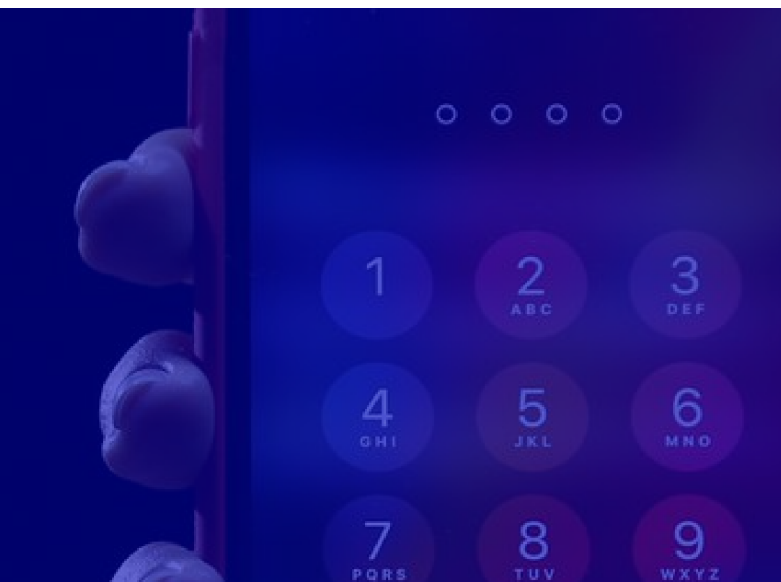
### Environmental Checks

Ensures the application is only running on a trusted device and hasn't been sideloaded elsewhere.



### Anti-tamper technology

Prevents an attacker changing or modifying the application. This prohibits them from using the application code maliciously or from creating "clone" versions of the application.



## Conclusion

With content owners becoming increasingly aware of the need to protect the whole video application, it is expected that rapid improvements will be made across the industry when it comes to mobile application security. The owners of OTT Video apps will have to adopt App Shielding technology if they are to continue to access the content their subscribers expect.

With consumers and regulators becoming more savvy about cybersecurity weaknesses putting personal data at risk, closing off weaknesses - such as those that expose Man in the Middle risks - will become a requirement of trading, otherwise customers will go elsewhere.

The acceleration of these trends will ultimately be a good thing for all ecosystem stakeholders: keeping consumers, businesses, and content safe.



# About Verimatrix

Verimatrix (Euronext Paris: VMX) helps power the modern connected world with security made for people. We protect digital content, applications, and devices with intuitive, people-centered and frictionless security. Leading brands turn to Verimatrix to secure everything from premium movies and live streaming sports, to sensitive financial and healthcare data, to mission-critical mobile applications. We enable the trusted connections our customers depend on to deliver compelling content and experiences to millions of consumers around the world. Verimatrix helps partners to get to market faster, scale easily, protect valuable revenue streams, and win new business.

