THE STATE OF MOBILE SECURITY

95% OF BANKING APPLICATIONS LACK THE SECURITY REQUIRED

A joint research between Verimatrix and UL

verimatrix
DRIVING TRUST

TABLE OF CONTENTS

Executive Summary

THE STATE OF MOBILE SECURITY

Security breaches and cyber-attacks are big news. Organisations that hit the headlines take a long time to recover. The market is increasingly complex, none more so than the one operated in by mobile banking applications. Within this market, it is unclear where banks and financial institutions are with regards to the risk assessment of their mobile applications and the potential exposure it brings.

UL and Verimatrix have undertaken joint research to assess the market – analysing publicly available m-banking applications with the aim of bringing clarity about the state of security within mobile banking applications available today. With increased pressure on banks to ensure that they Know Their Customer and GDPR looming on the horizon, this is a topic that is a high priority at every bank.

The results of the research should be a wake-up call to every bank.

There are no security standards for banking applications. To find a benchmark, the research uses the standards defined by the payment networks. For this use-case, the payment application level of assurance is quite harmonized and standard, thanks to the global schemes that are mandating evaluation by 3rd party accredited laboratories. In Mobile Banking, it is more of a greenfield, and in the absence of compliance requirements, free for each to do as they please. Very few rely on external risk assessment and security evaluation.

The intent of the research was to measure where the mobile banking applications stand compared to the mobile payment applications (used as a reference point guiding us to better illustrate the comparison in security). **Only 5% of the applications analysed came close to this standard.**

Not reaching this benchmark puts banks at unnecessary risk of:

- Brand damage;
- Fraud through uncontrolled access to back-end systems;
- GDPR fines.

A few simple steps can be taken to greatly improve the security of mobile banking applications.

Do not reinvent the wheel, and use partners that have experience, having done it before so they can support you in validating your decisions and scenarios. It is also safer to have your solution assessed as early as possible and get an external security evaluation to get an unbiased report, from a team of experts that are dedicated to continuous security. This will bring you both a higher confidence level on security and the economies of scale attached.

Equally, it is just as important to address process and secure development life cycle as it is about the absolute security. The real risk remediation is in how quickly you react and adapt to the new attacks and flaws that will differentiate you in the market. When you select your software protection architecture and solutions, keep in mind to evaluate beyond the level of assurance of the module and architecture, but also the design and the attention and reactivity you receive. It is not recommended to build security solutions in-house but rather rely on a proven solution, that gets exposed via their broad install base, and has teams dedicated to securing mobile applications. When you look at the pace of new operating systems, hardware and new attack techniques, it is increasingly challenging to keep up to date with the in-house team.

Make sure you **select the right partner** that will be on your side during challenging times.

INTRODUCTION

Not a day goes by without another security breach or cyber-attack, to the point where it is not news anymore.

The growing complexity of the current market we live in is only increasing the exposure of organisations. Complexity is coming from many areas: evolving development approaches such as the expanding use of outsourcing and more powerful end points (e.g. mobile devices) connecting to networks are two examples.

Yet, it is unclear where banks and financial institutions are with regards to the risk assessment of their mobile applications and the potential exposure it brings. This brings uncertainty to the market: **banks do not know what standards they should be aiming for**, while customers naïvely trust that their bank has taken care of security.

Contrast this with Mobile Payments, where the payment application level of assurance is harmonized and standard, thanks to the global schemes (MasterCard, Visa, etc.). These schemes are mandating security evaluation by 3rd party accredited laboratories against a defined standard. **Mobile Banking** is more of a greenfield; in the absence of compliance requirements, free for each to do as please. Very few do rely on external risk assessment and security evaluation.

To give clarity on the current state of the market, Verimatrix and UL security have undertaken joint research – analysing publicly available m-banking applications. This paper presents the finding of that research, assesses how the banks stack up against the industry as whole, and provide guidelines on how to strengthen applications with just a few simple steps.

With increased pressure on banks to ensure that they Know Their Customer and GDPR[1] looming on the horizon, this is a topic that should be a high priority at every bank.

[1] https://www.eugdpr.org

THE RESEARCH: PROCESS, SAMPLE & RANKING

To better understand the current state of security in mobile banking applications, UL and Verimatrix selected **19 applications** and utilised their in-house security labs to analyse them for common security weaknesses.

The applications were selected based on the following criteria:

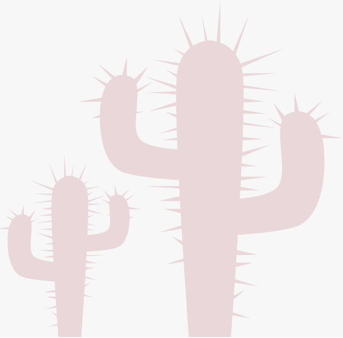
1 **Standard consumer facing applications from banks** that allow account management ("m-banking apps");

2 **Limited to Android applications** to allow comparison with Mobile Cloud-based Payment security baseline;

3 **Spread geographically** for a global view;

4 Even **split** between challenger banks and established incumbent banks;

5 **Selected randomly**, from the total available market.



Throughout the research, **the security standards defined by MasterCard and Visa for Mobile Payment Applications were used as a benchmark.** The standards from the card schemes are the only mobile security standards. They provide an excellent benchmark to assess the wider mobile financial industry.

The authors of this paper, would like to see all mobile banking applications be **at least as secure as mobile payment applications**. The reason for this is simple: both use cases handle banking customers' money. Given that there is typically more money handled by the general banking application than the payment application – security needs to be taken at least as seriously. These standards have been shown to be achievable and give the right level of protection.

> **Security** needs to be taken at least as seriously.

Even more so at the advent of PSD2[2] and open banking, which will ease the process of payment and money transactions outside the card rails. Whether it is SEPA[3], instant payment, or account to account transfers, the **same level of assurance should be required** at all times.

The European Commission is mandating the reporting on fraud rates, and may require higher level of security and second factor authentication when the minimum threshold is not met.

Ranking

To easily assess the findings, the researchers defined a **ranking system for mobile banking application security**. This uses the mobile payment standards as the benchmark rating and gives each application scoring (A to E) similar to rating systems used for car CO_2 emissions.

The ranking system is defined in the table on the next page. The researchers analysed each application using a combination of automated tools and manual inspection. This allowed them to assess each application against the ranking criteria.

An application that achieves the benchmark and is protected to mobile payment standards would score a B in the rankings.

[2] https://ec.europa.eu/info/law/payment-services-psd-2-directive-eu-2015-2366_en

[3] https://ec.europa.eu/info/business-economy-euro/banking-and-finance/consumer-finance-and-payments

Ranking	Criteria
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 obfuscation[4] of all critical code • Strong anti-tamper[5] protection of the application • Cryptography protected by whitebox[6] (or equivalent technology) • No sensitive text visible in static analysis of code • Network traffic encrypted using TLS[7] 1.2 and downgrade not possible • Certificate pinning[8] applied to networking • Strong device binding
B **Payment Equivalency** (Visa and MasterCard's standards for cloud based payments are used as a benchmark)	• Code handling sensitive data and algorithms is developed in a language that compiles to processor native machine code (i.e. C/C++) • Strong 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 1.2 and downgrade not possible • Certificate pinning applied to networking • Strong device binding
C **Standard** (Should be minimum reached by all banking applications)	• 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 1.2 and downgrade not possible • Certificate pinning applied to networking • Strong device binding[9]
D **Basic security**	• Obfuscation of critical code • Network traffic encrypted • Device binding
E **Little or no security**	None

[4] Obfuscation means scrambling computer code to make it less-intelligible to a human.

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

[6] Whitebox technology protects cryptographic operations and keys.

[7] TLS (Transport Layer Security) is the standard encryption protocol of the internet.

[8] Certificate pinning validates that the end point of communication is the intended end point.

[9] Device binding is a technique to lock an application instance to a particular phone.

RESULTS

The collated results should be a wakeup call for the mobile banking industry.

Only 5% of the applications investigated came close to reaching the standard required to pass a security evaluation from the payment schemes. In fact, from the sample set, 95% of the applications offered Little or Basic security.

The researchers know from past experience that some banks do take security of their mobile applications seriously. These are the banks the researchers work with on a daily basis. Given no application in the sample set achieved an A or B rating, it emphasises how much in the minority these banks are.

To try and get a better understanding of the results, it is interesting to **break them down by geography**; and also, to see if established banks are achieving better security than challenger banks.

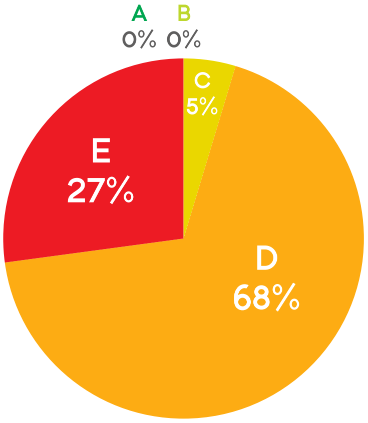
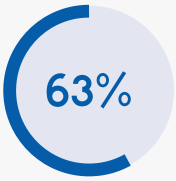


Figure 1 - Collated Results



of Gen X and Millenials are using mobile banking applications

of the banking industry **does not reach the security benchmark** laid down by the payment schemes

of Europe's mobile banking applications are all that achieve appropriate levels of security

Compared to mobile payment app security, **mobile banking app security is severely lagging behind***

*Source: Verimatrix & UL Mobile Banking App Research 2017

Geography

From the sample set, it can be argued that the European banks are slightly more security aware than other geographies; but in reality, **there is little correlation between a bank's location and the security of its mobile banking application**. This could simply be because most banks work within the global context.

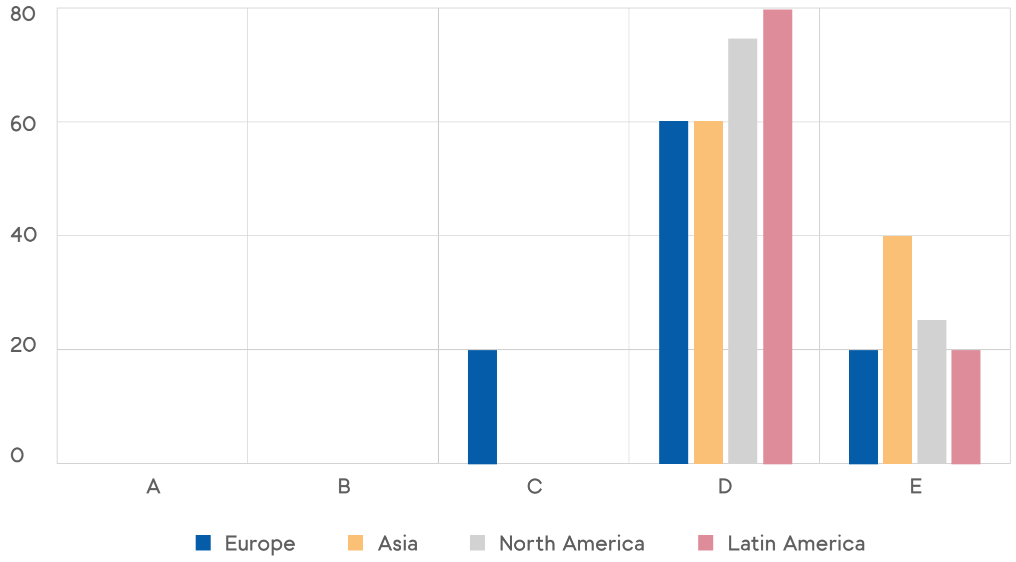


Figure 2 *- Geographic Breakdown*

Established vs Challenger Banks

36%
of the established banks
offer **Little security**

When trying to understand different attitudes of long established banks versus their newer challengers, it is quite interesting. The challenger banks all follow a similar pattern: tending towards Basic security.

The established banks are more spread: some offering reasonable security, while a high proportion (36%) offer Little security.

The reason for this is due to the way the two groups typically build applications. **The challenger banks build applications in a manner similar to any mobile development team.** This means they apply the same Basic security techniques that are considered best practice for **normal** application development. There is no extra consideration given to protect the application; even though it is connecting into a banking infrastructure.

The established banks have built their mobile banking applications from a different base – their application development has longer routes and was initially a side project of the bank's internet banking product. This legacy means that - unless the bank has recognised the specific and unique risks posed by its mobile application - the security is often below that of challenger banks.

Banks trade on trust. They run advertising campaigns to emphasise their security credentials and to "educate" their customer base.

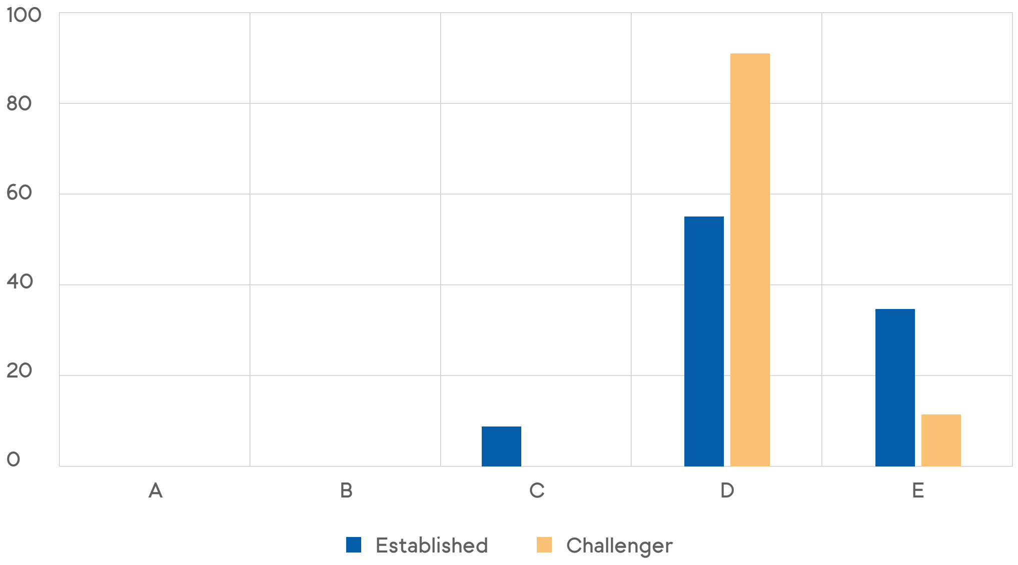


Figure 3 *- Challenger vs Established Breakdown*

Banks trade on trust. They run advertising campaigns to emphasise their security credentials and to "educate" their customer base. The best practice they present to their customer base is to recommend **stronger authentication** (e.g. multi-factor, higher complexity passwords, etc.) for banking credentials than for social media; however, based on the result of this investigation, this seems to be a futile recommendation unless the authentication methods are properly secured.

Pockets of Resistance

Within the applications analysed, there were **pockets of security resistance** – small parts of the application that on their own would reach B (Payment Equivalency) or in some case A (Highly Secure) standards.

One obvious "pocket" is when an application contains **payment functionality that has been certified by Visa and/or MasterCard**. By definition, to reach this certification, the payment portion of the application will be at least a **B ranking**.

These "pockets" tend to be third party libraries that come from security companies – providing functionality like authentication and device binding. The suppliers of these libraries take security seriously – they understand the risks.

Networking

One area that was generally taken seriously within the applications was connections to back-end servers. Good practice was applied where all applications use encrypted communication, **58% implementing certificate pinning** (guarding against man-in-the-middle attacks) and **21%** even forcing the use of the very latest TLS v1.2.

Of course, if the application **is not protected from tampering** (none of the applications tested showed good resistance here), then it is possible for an attacker to undo the networking encryption and pinning.

If the application is **not protected from tampering**, then it is possible for an attacker **to undo the networking encryption and pinning**.

This perhaps starts to explain a typical bank's view point when it comes to security. They have a long history and a lot of expertise in securing backend systems. Mobile is the new kid on the block and the depth of understanding around the risks it poses are yet to be fully understood within banks.

BUSINESS RISK

It is relatively surprising to see a bank application developer buying a third-party security library for a component only to leave the rest of the application unsecured. It is important to look at the **security from a holistic end-to-end approach** instead of just components.

The old adage stating that a solution **is only as secure as the weakest link in the chain** is valid in this case. Doing partial security is like locking only the front wheel of your bicycle to the railing.

We have established that mobile banking applications typically have **security weaknesses**. That is only a problem if these weaknesses can be exploited. The reality is that they can be. At the end of the day, the application is a token linking the user to the bank's infrastructure. It is important to look into security by **design principles, social engineering, threat models and logical attacks**. Once these principles are applied, you will naturally come to the conclusion of the necessity in looking at the end-to-end security and not components individually. There are great documentation and material available from groups working on the current threat and attack models that could be a good starting point (some of it available freely on the UL-TS.com website).

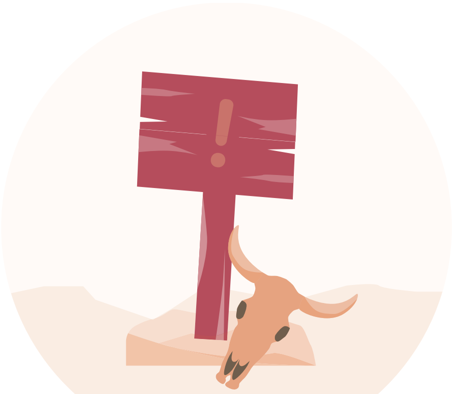
A common belief that is illustrated by the results is that security can be implemented on the back-end – similar to protecting traditional IT systems. This approach works when access to the back-end systems is **highly controlled** and no sensitive data passes across the security boundary.

When those systems are opened up to mobile applications, security experts tell us back-end systems are very good at detection and *containment*; but **they cannot prevent attacks that leverage weak mobile clients**.

Uncontrolled access to servers

Mobile applications access banking services through APIs. **Attackers can dissect applications to learn how to use these APIs** for their own means and to extract the secrets that are meant to stop unauthorised access.

Sometimes this is for "*legitimate*" reasons where a company is effectively screen scraping through private APIs to provide aggregation services. Other times it is **criminals gaining access to bank accounts**. In either case, the bank is ultimately liable for any fraud caused by misuse of its APIs.



Brand damage

Any publically revealed attack on an organisation can leave **lasting damage to brand** – particularly if that organisation trades on trust and security. In the short term, C-level executives are called to explain themselves to government before **being fired by the organisation**. Long term, exposed organisations find their share price suppressed.

A criminal attack is obviously more damaging, but there are many academics and hackers who are looking for weaknesses just for the kudos.

Brand damage is also important to look at, not only from a security perspective, but also in ensuring customer expectations are met. **User Experience** does not have to be traded to improve security. For example, some application developers will set the assumption of not allowing the application to be used on rooted devices, which may alienate some customers and put the application at risk since it will happen anyway. Another approach is to assume that the application **will be installed on a compromised device**, and get the application developer to secure it even in that scenario, which is a much safer and secure approach.

GDPR

New European regulations are putting more requirements on organisations to keep their customers' data safe. The basis of the requirement is that the customer owns their data and an organisation "borrowing" that data has a duty of care over the information and needs to use "state of the art" security **to keep customer data from fallings into unauthorised hands**. Any lapses in that duty can result in large fines (up to €20m or 4% of turnover, whichever is higher).

Mobile is not exempt from the regulations. That means that any personal data used or stored within a mobile banking application must be properly protected from prying eyes.

Beyond the application security, it is also important to look at how the information is stored and logged on the device. This is becoming absolutely crucial with GDPR kicking in this year. There is **a need to identify all Personal Identifiable Information** (PII) and make sure that all data is **encrypted and secured**, this may seem common sense and trivial however the researchers have seen logs stored on the device in clear, freely available to access.

> There is a need to **identify all Personal Identifiable Information** (PII) and make sure that all data is **encrypted and secured**.

RECOMMENDATIONS

This research should serve as a wake-up call for the mobile banking industry. **It does not mean the west is lost**. Banks have a very strong understanding of security risk. Some simple steps can tame the **hostile environment**.

As highlighted earlier, there are already quality material available for free as a starting point. It is important to take these concepts into consideration as early as possible, **the later in the process you start, the more complex and slow your solution will be**; and will significantly decrease the chances of success.

Make sure you surround yourself with the right experts in order to differentiate between perception and reality. MobeyForum[10], for example, is a great platform.

If you don't know where to start, contact the authors of this paper, they will be happy to point you to educational material.

Bring in the experts early

Banks have a strong history of partnership when it comes to technology. Recognise that mobile changes the game and bring in experts from outside to extend the bank's security expertise. There is no need to reinvent the wheel; **use partners that have experience** - having done it before, they can support you in validating your decisions and scenarios.

The right partners can also bring economies of scale and wider visibility of the risks – operating across multiple organisations and industries.

[10] https://www.mobeyforum.org

Have apps (and wider ecosystem) pen tested

Security is not easy and it is important to validate that there are no gaps. External security labs have skilled testers that can find gaps left in the security.

It is safer to have your solution assessed as early as possible and get an external security evaluation to get an unbiased report, from a team of experts that are dedicated to continuous security. This will bring you a higher confidence level on security.

Beyond this, **it is good practice to build that knowledge internally**, get your own pen testing (following the right processes and best practices) but still rely on external experienced pen testers. They will always find something to improve and challenge you in the right way – helping to build your internal expertise. Do not wait for the bad guys to do it, because that is how you end up in the news.

Secure the Application

Security cannot be achieved by securing just part of bank's infrastructure. A customer's mobile device may not be under the bank's control and needs to be considered a hostile environment; but the application running on it is under the bank's control. **Securing the application secures the entry point** into the bank's infrastructure and protects any sensitive data processed and stored by the application.

Properly securing an application requires a great deal of expertise. To attempt it solely within the bank can be a **very costly and time-consuming process**; the fragmentation of Android devices alone is getting increasingly difficult to manage in functionalities and exponentially difficult to secure. Organisations that specialise in securing applications **bring benefits** from their expertise and also economies of scale.

> **Securing the application secures the entry point** into the bank's infrastructure and protects any **sensitive data processed** and **stored** by the application.

Development Methodology

Security is not just tactically applying technology. It requires the right mind-set and processes throughout the development cycle. **This does not have to be onerous or time consuming**; many of the techniques are good development practice that will smooth any software development project.

It is not the absolute security of the application at a point in time that matters, but the continuous security. **Be mindful that you will never know when a new threat or attack will occur and your reaction time is even more important to manage.** You need to have not only the right mind-set, but the processes and measures to mitigate that risk as fast as possible.

When you are sourcing an external solution, make sure you cover the reactivity in fixes and ask the right questions: who they partner with, how they keep up to date with best practices, do they have a continuous training program, which industry organisations they participate in.

Long story short, don't do it on your own, and make sure you surround yourself with the right **external experts to guide you.**

> **Security is not just tactically applying technology.** It requires the right **mind-set** and **processes** throughout the development cycle.

CONCLUSION

It is a good strategy to accept that the worst is going to happen and to prepare your organisation for the breach, just like a fire drill.

It is important to plan ahead so that the risk is reduced and a remediation plan is in place to minimise any fallout both in terms of image and PR but also technically. Otherwise you take the risk of making rash, hasty decisions instead of a rational course of action based on informed decisions.

Do not reinvent the wheel, and use partners that have experience, having done it before so they can support you in validating your decisions and scenarios. It is also safer to have your solution assessed as early as possible and get an external security evaluation to get an unbiased report, from a team of experts that are dedicated to continuous security. This will bring you both a higher confidence level on security and the economies of scale attached.

It is equally important to address process and secure development life cycle as it is to drive for absolute security. The real risk remediation is in how quickly you react and adapt to the new attacks and flaws that will differentiate you in the market. When you select your software protection architecture and solutions, keep in mind to evaluate beyond the level of assurance of the module and architecture, but also the design and the attention and reactivity you receive.

It is not recommended to build security solutions in-house; rely on a proven solution that gets exposed via their broad install base, and has teams dedicated to securing mobile applications. When you look at the pace of new operating systems, hardware and new attack techniques, it is increasingly challenging to keep up to date with the same team.

Make sure you select the right partner that will be on your side during challenging times.

EXAMPLES

While the aim of the research was to establish clarity on the overall the state of the industry, it is interesting to drilldown into some of the specific findings.

While specific to particular applications, these highlight general issues that contribute to the overall state.

Device Binding

Device binding is a **technique to lock an instance of the mobile application to a particular phone**. This stops the application being cloned. It also helps control access to back-end servers as user credentials can be locked to a given device.

The example application was 100% developed in Java. It was easy to extract the compiled Java classes from the application package using freely available reverse engineering tools, such as Baksmali[11]. The source code was then recovered using JAD[12]. Visual inspection of the recovered source code showed that there was no obfuscation applied.

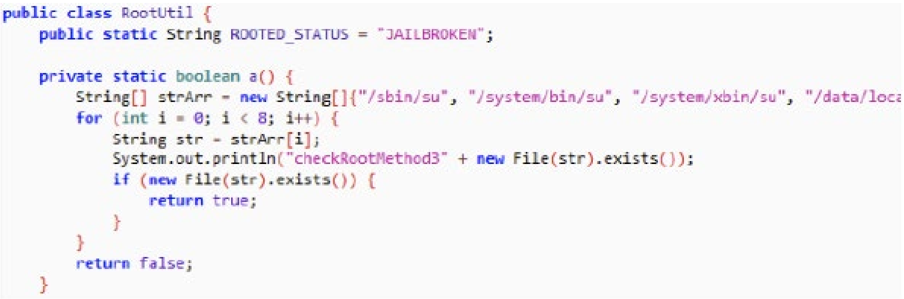
The only barrier to attacking the application was a **weak root detection mechanism.** There were four root detection methods all located within one class. One example is shown in the code sample below. It would be easy to circumvent it just by slightly modifying an open source module of the Xposed framework: RootCloak[13] (RootCloak just hooks some specific calls to Android API to intercept/tamper with some common checks performed at the Java level).

> The only barrier to attacking the application was a **weak root detection mechanism.**

[11] https://github.com/JesusFreke/smali

[12] http://www.javadecompilers.com/jad

[13] https://github.com/devadvance



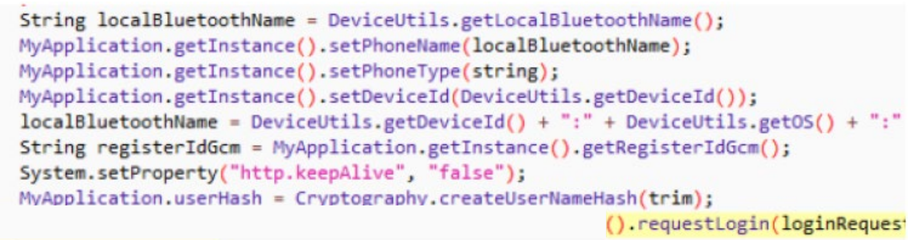
```
public class RootUtil {
    public static String ROOTED_STATUS = "JAILBROKEN";

    private static boolean a() {
        String[] strArr = new String[]{"/sbin/su", "/system/bin/su", "/system/xbin/su", "/data/loca
        for (int i = 0; i < 8; i++) {
            String str = strArr[i];
            System.out.println("checkRootMethod3" + new File(str).exists());
            if (new File(str).exists()) {
                return true;
            }
        }
        return false;
    }
```

Once the root detection has been disabled, it is possible to attack the device binding. It is possible to see in the code that there are different mechanisms to perform the device binding.

During the launch/registration different identifiers and data are retrieved, checked and sent to the server:

- Instance ID;
- IMEI number;
- Android ID;
- Devices/app characteristic: manufacturer, model, OS version, app version...;
- MAC address;
- Geolocation mechanism;
- Registration ID (which is a custom identifier set during registration).

It was also noticed that during the login, those identifiers are sent to the server together with the credentials (actually a SHA-256 hash of the password). This is shown in the code snippet below.

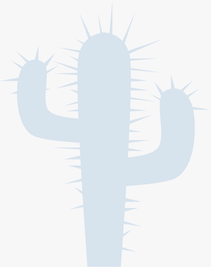


```
String localBluetoothName = DeviceUtils.getLocalBluetoothName();
MyApplication.getInstance().setPhoneName(localBluetoothName);
MyApplication.getInstance().setPhoneType(string);
MyApplication.getInstance().setDeviceId(DeviceUtils.getDeviceId());
localBluetoothName = DeviceUtils.getDeviceId() + ":" + DeviceUtils.getOS() + ":"
String registerIdGcm = MyApplication.getInstance().getRegisterIdGcm();
System.setProperty("http.keepAlive", "false");
MyApplication.userHash = Cryptography.createUserNameHash(trim);
                                                ().requestLogin(loginReques
```

Understanding the device binding algorithm allows the application instance to be easily cloned. It would also allow an attacker to automate an attack with a powerful server pretending to be a user's mobile device.

Some simple steps would have made this attack much harder:

- **Obfuscate the application** so the root detection and device binding code would be harder to find;
- **Use string encryption** to hide key words like "jailbroken" from static analysis;
- **Do not rely solely on hookable system calls** as the inputs to a device binding algorithm;
- **Protect the device binding algorithm** within a strong cryptographic boundary such as a white-box.



Unpinning Certificates

Certificate pinning is a technique to **ensure the mobile client is talking to a trusted server**. This defends against man-in-the-middle attacks.

The example application was 100% developed in Java. As with the first example, it was easy to extract the complied Java classes from the application package using Baksmali. The source code was then recovered using JAD. In this case, visual inspection of the recovered source code showed that there was obfuscation applied. The code was restored to a human readable state using Deguard[14].

Once the obfuscation was undone, it was possible to remove the root detection in a manner similar to the first example.

From the source code, it can be seen that some network connections are using some certificate pinning from a third-party package. It implements a custom trust manager and certificate's chain checking.

Code snippets are not provided for this example so as to not reveal the name of the thirdparty package.

[14] http://apk-deguard.com/

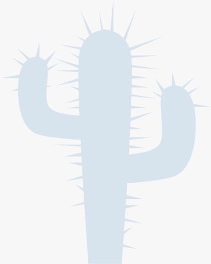
[15] https://quixxi.com

It should be noted that custom implementation of such mechanisms is considered to be tricky and it is generally recommended to reuse proven libraries for certificate pinning. Indeed analysis using automated security scanning tool QUIXXI[15] has raised a medium severity finding of the implementation used in this example: it ignores all SSL certificate validation errors, making the app vulnerable to a man-in-the-middle attack (this was confirmed via a manual code review).

Given that there is no anti-tamper protection in the application, it would be easy for an attacker to remove the certificate pinning, redirect network communication through a proxy, re-build the application and deploy onto the application stores as an imposter application. **The user would be unaware that the application was an imposter but the attacker would be able to snoop on all network traffic**.

Some simple steps would have made this attack much harder:

- **Protect the certificate pinning** by implementing it in more secure native code (i.e. C++) not Java;
- **Apply strong obfuscation** to the application;
- **Apply anti-tamper technology** to the application.



About Verimatrix

Verimatrix (Euronext Paris: VMX), formerly known as Inside Secure, is a trusted business partner providing software security and business intelligence solutions that protect content, applications, and devices across multiple markets. Many of the world's largest service providers and leading innovators trust Verimatrix to protect systems that people depend on every day.

With more than 20 years of experience and the top minds in the industry, the company is uniquely positioned to understand and proactively anticipate security and business challenges for customers. Verimatrix partners provide innovative, customer-friendly solutions that are cost-effective, easy to deploy and supported with responsive customer service teams based worldwide.

To learn more, visit verimatrix.com.

About UL's Identity Management and Security services

UL's Identity Management and Security division guides companies within the mobile, payments, and transit domains through the complex world of electronic transactions. UL is the global leader in safeguarding security, compliance, and global interoperability. Offering advice, training, compliance and interoperability services, security services, and test tools, during the full life cycle of your product development process or the implementation of new technologies. UL's people proactively collaborate with industry players to define robust standards and policies. Bringing global expertise to your local needs. UL has accreditations from industry bodies including Visa, MasterCard, Discover, JCB, American Express, EMVCo, UnionPay, PCI, GCF, GlobalPlatform, NFC Forum, and many others.

For more information, visit ims.ul.com.

About UL

UL helps create a better world by applying science to solve safety, security and sustainability challenges. We empower trust by enabling the safe adoption of innovative new products and technologies. Everyone at UL shares a passion to make the world a safer place. All of our work, from independent research and standards development, to testing and certification, to providing analytical and digital solutions, helps improve global well-being. Businesses, industries, governments, regulatory authorities and the public put their trust in us so they can make smarter decisions.

To learn more, visit UL.com.

To learn more about our nonprofit activities, visit UL.org.