intertrust®

INTERTRUST WHITE PAPER

Taking Steps to Protect Financial Mobile Applications

TABLE OF CONTENTS

Executive Summary

With the introduction of Apple's iOS and Google's Android operating systems, the world saw an advent of smart mobile technologies that completely revolutionized the way people and businesses communicate and transact. The success of the smartphone and its ease of use have led consumers to use mobile devices to perform a variety of financial transactions such as mobile banking, remote deposits, mobile commerce, and so on. The mobile payment market worldwide is expected to increase at a compound annual growth rate of 20.5% through 2024. In 2015, it was valued at $338B, with a projected growth to $1.7B by 2024[1]. Another report predicts that in the U.S., mobile wallets are expected to surpass the use of both credit and debit cards by 2020[2].

Although the popularity of mobile technologies has greatly simplified the use of day-to-day financial operations for the end user, it has definitely created headache for companies developing these applications and providing the back end systems to support their operation. The costs of dealing with cybercrime incidents has reached the point where it is now a major threat to the corporate bottom line. For example, a study released in 2017 showed that the average annual cost of cybercrime for companies and institutions providing financial services was over $18M[3]. Given the high potential financial losses associated with software-based attacks on financial institutions, it is imperative that companies take concrete steps to ensure the security of their mobile applications.

This paper describes the latest statistics on cybercrime in the mobile financial industry and the most common types of attacks on mobile applications. Finally, this white paper will focus on Intertrust's robust solution to protecting financial applications—a set of application shielding tools that are intended to increase application-level security and render cyberattacks on financial applications extremely difficult and expensive to execute.

Attacks On Financial Applications-A Menacing Threat

Traditionally, the technical attack vectors which financial companies needed to be concerned about have been either personal computers or servers. However, the introduction of smart mobile devices has opened up an entirely new range of platforms for criminals to exploit. The ever increasing popularity of mobile devices has led to a situation where companies must constantly keep fighting new techniques and advanced persistent threats that try to infiltrate networks. Perimeter security is slowly disappearing, and the emphasis is now placed on endpoint security. The connected world has changed the face of business, and almost every industry, especially the financial sector, must adopt a mobile strategy to address their customers' security needs.

A Global Shift in How We Do Banking

For decades people who needed to do any kind of banking transactions had to go to the local branch and do business in person. People are increasingly choosing to do their banking and other financial transactions online via mobile phone/tablet applications. One report based on data from three European banks states that in 2017 branch usage was down by 40% since 2014[4]. The advantages of choosing a mobile application over a physical visit to a bank are obvious—mobile banking is faster and more convenient to the user.

However, this global paradigm shift creates a problem for banking and financial companies. How can banks ensure that mobile financial applications provide the same level of security as a brick and mortar branch? As will be discussed later in this paper, security has been

---

[1] Global Mobile Technologies Market (2017). Transparency Market Research. www.transparencymarketresearch.com/mobile-payments-market.html
[2] Global Payments Report (2017). Worldplay. www.worldpay.com/global/insight/articles/2017-11/global-payments-report-2017
[3] Cost of Cyber Crime Study (2017). Ponemon Institute LLC.
[4] https://entrackr.com/2018/04/hotstar-vivo-ipl-2018-edition-first-week/

lagging behind when compared to other aspects of the shift from in-person to mobile banking. Partially this is related to the fact that mobile financial services is still a very young industry without clear standards and significant self-regulation. Therefore, to avoid financial losses, legal liability, and damaged brand reputation in the long run, it is absolutely critical that financial companies take application security seriously and implement it right.

How Real Is the Risk?

Although one might claim that mobile application attacks have not yet captured the same publicity as traditional major breaches, it would be unwise for security teams to delay or limit investment in a mobile security program. The number one concern for mobile payment users is that their device will get hacked or their data will get intercepted[5], and there is sound basis for these concerns. Given the ever increasing use of mobile devices and the drive towards mobile banking, it is inevitable that mobile security breaches will become the next headline. For example, according to a report done by Ponemon Institute, 60% of organizations admitted that a security incident resulted from an insecure mobile application[6]. Because mobile applications in the financial industry are very lucrative targets, there is little surprise that financial services have become a particularly threatened sector. In fact, financial services have the highest cost of cybercrime when compared to a range of other industries[7].

The stage has been set for attacks on mobile financial and payment applications to reach the global headlines. But before talking about the ways of solving the mobile application security problem, let's look into the mindset of a cybercriminal and explore the attack tools they typically use.

How Cybercriminals Attack

This section describes some of the typical attack vectors employed by hackers to compromise mobile applications, and the common weaknesses in such applications.

Typical Attack Vectors

Reverse engineering. Almost always reverse engineering is the cornerstone of any attack on a mobile application. Before an application can be attacked, it has to be well understood. Reverse engineering is a tedious process of analyzing the compiled application code in order to identify and understand the parts it is composed of and the way they work. Hackers are familiar with common structures in compiled code. For example, they might look for a string corresponding to an error message related to their objective (e.g. "invalid PIN") and trace where that string is used. They leverage sophisticated techniques such as static analysis and debuggers, which help them understand the overall structure of the code, where the functions are located, how they are called from other functions. Talented reverse engineers can look at assembly language code and immediately recognize that it is, for example, performing cryptographic operations.

Tampering. A cybercriminal might take a legitimate payment application and modify it so that instead of the application performing the tasks originally designed for it, the application actually performs tasks for the cybercriminal such as stealing information from the mobile device. Such modified applications can be fairly easily distributed to a large user base by hijacking a public Wi-Fi hotspot, tricking users into installing applications from malicious websites, or taking advantage of people who prefer to use rooted or jailbroken devices.

---

[5] https://yourstory.com/2018/04/what-driving-hotstar-record-viewership-ipl-2018/
[6] Study on Mobile and IoT Application Security (2017). Ponemon Institute LLC.
[7] Cost of Cyber Crime Study (2017). Ponemon Institute LLC.

Stealing sensitive information. Very often mobile applications, especially those dealing with finance and payments, have to store various kinds of secret and sensitive data that is necessary for its operation. This might include usernames, password hashes, account numbers, PIN codes, cryptographic keys, personally identifiable information, proprietary algorithms, intellectual property and so on. All this information can be subject to exploitation if a hacker were to reverse engineer the application. For example, by leveraging the stolen information an attacker might be able to pretend to be a legitimate user, mimic the way the application communicates with other parties, create unfair competition, or eavesdrop on encrypted network messages.

Rootkits and malware. Malicious software, if present on a device, can directly intercept and modify API calls of other applications and hence manipulate data in transit, such as credentials, payment information, and so on.

Exploiting vulnerabilities. Studies have shown that most software has 15-50 bugs per 1000 lines of code[8]. Even small unintended software bugs can quickly add up to a major breach. One example is Eavesdropper, where a single SDK vulnerability affected hundreds of enterprise applications and exposed millions of call metadata, voice recordings and text messages[9]. Very often these types of vulnerabilities go undetected until they are too big and too late to manage. Hackers look for such vulnerabilities in application code to achieve their goals.

Common Weaknesses in Financial Mobile Applications

Almost always a successful attack on a mobile application results from poorly designed or implemented software. Sadly, financial applications are no exception. For instance, a study performed in 2014 showed that most of the iOS mobile banking applications from the top 60 leading banks have significant security vulnerabilities[10].

Intertrust has decades of experience in software security. Our researchers have been following the advent of mobile technologies from the very beginning, and the most common problems with mobile applications that often lay the groundwork for subsequent attacks are as follows:

- Lack of protection against debugging and reverse engineering of code
- Unobfuscated code that is easy to analyze and exposes vulnerabilities
- No jailbreak/root detection
- Using cryptographic keys and passwords in plaintext
- Sensitive information exposed in log files and crash reports
- Unencrypted local databases used to store sensitive information
- Unprotected resources and metadata, such as images, which can be used to create phishing applications
- Ineffective risk management program or lack thereof

It is imperative for financial institutions to take the necessary steps to protect their applications by making them harder to hack.

---

[8] Ratio of Bugs Per Line of Code, Dan Mayer (2012). Continuously Deployed. www.mayerdan.com/ruby/2012/11/11/bugs-per-line-of-code-ratio
[9] www.appthority.com/mobile-threat-center/blog/eavesdropper-mobile-vulnerability-exposing-millions-conversations
[10] www.computerworld.com/article/2475778/application-security/mobile-ios-banking-apps-are-miserably-insecure-leaky-messes.html

Practical Ways of Protecting Mobile Applications

In this section, we will touch upon the main techniques for protecting mobile financial applications against reverse engineering and other attacks.

Hardware-Based Security

Hardware-based security involves a dedicated piece of hardware that runs separately from the main device processor and cannot be directly manipulated by it. It is commonly used to provide strong protection for cryptographic keys—small pieces of data that are at the core of secure payment transactions and encryption of sensitive data. Some examples of such hardware are hardware security modules (HSM), trusted platform modules (TPM), and trusted execution environments (TEE).

The security of these systems relies on the fact that it is very difficult and expensive for attackers to reverse engineer a hardware module and manipulate its internal data. Generally speaking, hardware security systems can be considered "black box models" because their internal workings are essentially hidden to the observer.

Although the hardware-based approach does provide excellent security advantages, there are also significant downsides:

• Financial companies and institutions have no control over the chipset or the device a customer chooses to use, nor can they persuade customers to upgrade their devices. So, while TEE is secure it does not guarantee all devices are actually protected.

• Hardware-based security adds cost to a system. Manufacturers of platforms might choose cost sensitivity over the security risks of compromised keys—security is usually an afterthought.

• Vulnerabilities in hardware are difficult and potentially expensive to mitigate. Examples like Meltdown and Spectre illustrate that hardware and software manufacturers might need to spend large amounts of money and resources to issue patches to fix vulnerabilities in existing deployments[11].

• Different devices may contain different hardware with varying functionality that require complex logic in applications built to run on a wide range of devices.

• Hardware is not completely immune to attacks. Clever approaches such as differential power analysis can be used to extract keys from hardware by examining indirect patterns in signals emanating from the hardware.

• There are business models which preclude application developers from using secure hardware on a device even when it exists. Such is the case with Apple iPhone, where although it has ARM processors with the TrustZone extension, third-party applications are generally not allowed to use that functionality.

Application Shielding

Considering today's open architecture of mobile devices and the ease at which applications can be obtained and distributed, software vendors cannot rely on the assumption that their mobile applications will be used in a proper manner by the right users within a safe and controlled environment. Instead, the recommended approach is to assume the worst—that the software will eventually end up in the hands of an attacker who will subject it to various analysis tools to attempt to use it in a malicious manner. Hence, a logical conclusion is that security mechanisms have to be embedded into the application itself. This technique is sometimes referred to as application shielding or Runtime Application Self Protection

[11] meltdownattack.com

(RASP). Typically, this would involve code obfuscation, code integrity protection, rooting/jailbreak detection, white box cryptography, anti-debug protection, binary packing and so on.

Intertrust is a leading developer and provider of application shielding solutions, which can turn any software application into a self-contained fortress that can withstand even the most cunning attacks. The next section provides an overview of the Intertrust solution for protecting financial mobile applications.

Protecting Financial Mobile Applications with Intertrust's whiteCryption® Technology

According to a 2016 report by ENISA, [12] minimum requirements for building mobile payment applications include:

- Avoiding hard-coded sensitive information such as passwords or keys
- Employing anti-reversing techniques
- When possible, verifying the integrity of the running code, to ensure that it has not been backdoored

whiteCryption, a subsidiary of Intertrust, has decades of experience with advanced proprietary security technologies and techniques. We are in a unique position to help financial institutions greatly reduce their threat exposure on mobile platforms and mitigate risk.

whiteCryption is a leading provider of application shielding solutions that prevent hackers from reverse engineering and tampering with code. We specialize in world-class advanced obfuscation, tamper resistance, and white box cryptography solutions for mobile applications, desktop applications, firmware and embedded applications.

Application Shielding Portfolio

Our application shielding portfolio consists of two products:

whiteCryption Code Protection™ provides application developers with a comprehensive suite of anti-reverse engineering and runtime application security tools to help protect your applications on all popular target platforms. Code Protection is easy to use, provides simple means for fine tuning the balance between security and performance, and requires no significant changes to the code itself or the existing build chain. Since Code Protection secures source code before it is compiled, protected builds can easily be delivered to a mobile device or any other device in your financial services ecosystem. Some of the security features provided by Code Protection include:

- Code obfuscation
- Integrity protection
- Anti-debug protection
- Rooting/jailbreak detection
- Binary packing

[12] Security of Mobile Payments and Digital Wallets (2016). ENISA Report. https://www.enisa.europa.eu/publications/mobile-payments-security

whiteCryption Secure Key Box™ is an advanced white box cryptographic library that protects cryptographic keys for critical security functions such as device authentication, secure communications, and data encryption. With Secure Key Box, cryptographic keys are never in the clear in use, in transit or at rest, preventing hackers from stealing your keys and using them to masquerade as users, snoop on secure communications, or unlock content that is critical to your business. Secure Key Box is used as a drop in cryptographic library that functions in the same way as common cryptographic libraries, such as OpenSSL or LibTomCrypt.

Use Case: Host Card Emulation

Host Card Emulation (HCE) is a technology that allows a mobile device to perform bank card emulation on an NFC-enabled device without relying on access to a secure hardware element. When an application uses HCE, communications with the contactless terminal are no longer routed to the secure element; instead, it is routed through the NFC controller to the device's host processor on which the application is running. This innovation has opened an alternative path to contactless payments and other services that had no reliance on hardware-based security.

However, this new technology has also introduced security risks and expanded the attack surface of financial applications. For instance, communication between the NFC controller and the HCE-enabled application can be tracked by malware applications. The device can be rooted or jailbroken allowing the attacker to reverse engineer the application and create a malicious copy. Vulnerabilities in the software can be discovered and exploited for further attacks.

This is where whiteCryption technology comes into play. With the use of Code Protection and Secure Key Box, the application can be turned into an obfuscated and tamper resistant software fortress. Integrating whiteCryption technology would involve the following main steps:

1. Replace the application's crypto module with the Secure Key Box white box library.

This would ensure that the secret cryptographic keys are permanently encrypted at all stages of their lifecycle, eliminating the threat of keys ever appearing in plaintext and being stolen.

2. Integrate your build system with Code Protection.

Code Protection hooks into the build chain and makes numerous modifications to source code, such as obfuscation and insertion of various security checks. The end result would be an application that resembles your original application on the functional level (looks and works the same), but internally it would be completely transformed to withstand a wide range of software attacks.

Why Choose whiteCryption?

Our products are backed up by superlative support and professional services to help you achieve your business goals quickly and efficiently. Market leaders have deployed whiteCryption tools in mission critical finance applications around the world. Our technology protects millions of devices and applications, securing sensitive information and crypto algorithms running in hostile environments; we extend the secure perimeter around applications beyond where traditional technologies have gone. whiteCryption is an important weapon in protecting you and your customers from unwanted intrusion and misappropriation of personal information and data.

Learn more about whiteCryption at
https://www.intertrust.com/products/application-shielding

About Intertrust Technologies Corporation

Intertrust provides trusted computing products and services to leading global corporations – from mobile and CE manufacturers and service providers to enterprise software platform companies. These products include the world's leading digital rights management, software tamper resistance and privacy-driven data platforms for digital advertising, marketing technologies, DNA analysis, and IoT.

Founded in 1990, Intertrust is based in Silicon Valley, with regional offices in Beijing, Bengaluru, Hyderabad, Indore, Mumbai, London, Tokyo, Seoul, and Riga. The Company has a legacy of invention, and its fundamental contributions in the areas of computer security and digital trust are globally recognized. Intertrust holds hundreds of patents that are key to Internet security, trust, and privacy management components of operating systems, trusted mobile code and networked operating environments, web services, and cloud computing.

About whiteCryption

whiteCryption, a subsidiary of Intertrust Technologies, is a leading provider of application shielding solutions to prevent hackers from reverse engineering and tampering with code. We specialize in advanced obfuscation, runtime application self-protection (RASP), and white box cryptography solutions for mobile and desktop applications, firmware and embedded systems. whiteCryption protects content for the automotive, banking/finance, healthcare, and entertainment industries.

Copyright Information:

Contact Information:

Intertrust Technologies Corporation

920 Stewart Drive. Suite #100

Sunnyvale, California 94085, USA

https://www.intertrust.com