OneSpan
Be bold. Be secure.

MOBILE APP SHIELDING: HOW TO REDUCE FRAUD, SAVE MONEY, AND PROTECT REVENUE

WHITE PAPER

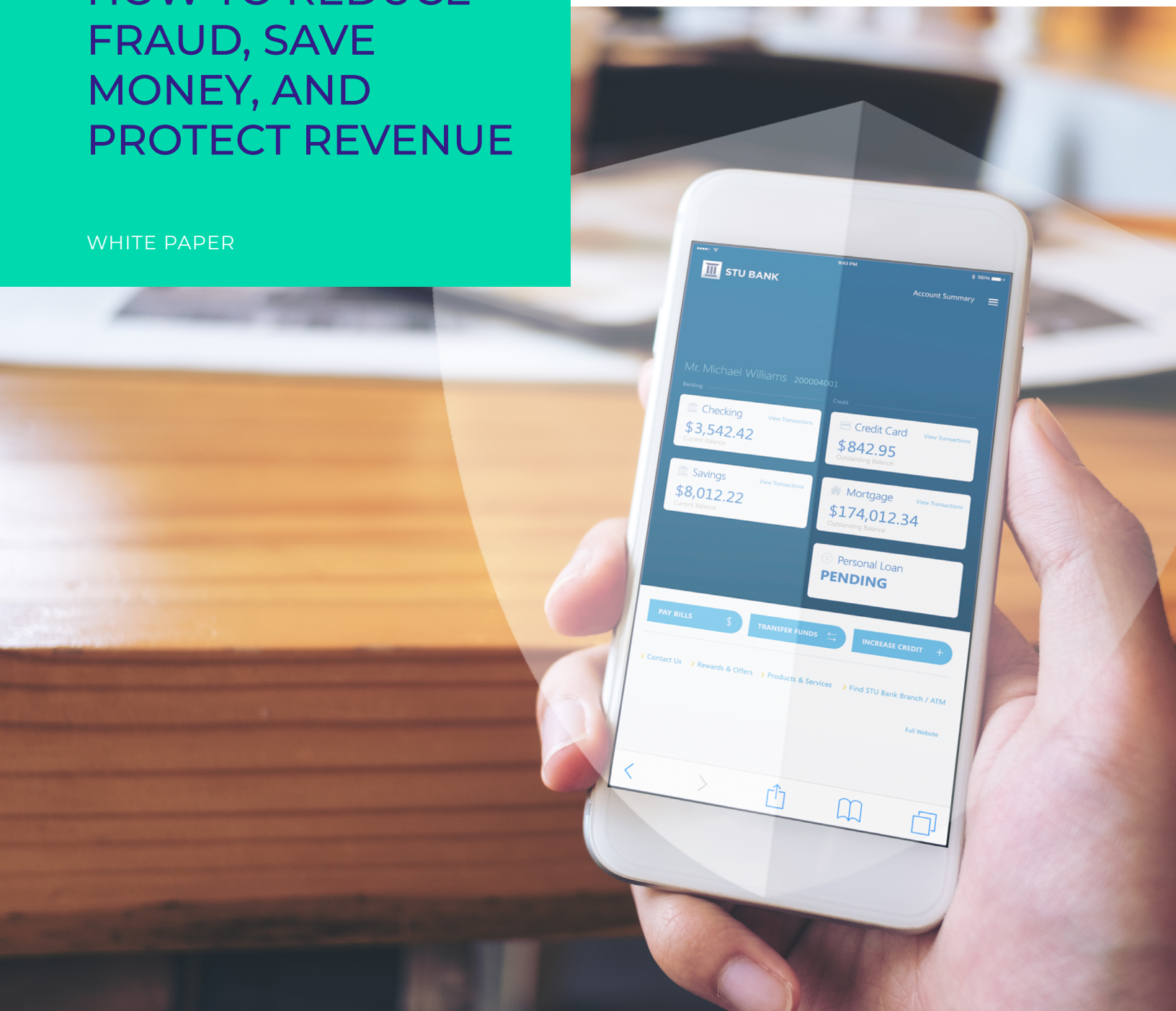


TABLE OF CONTENTS

EXECUTIVE SUMMARY

Fraudsters are targeting the mobile channel more aggressively than ever before. According to Kaspersky Lab, the number of banking Trojans attacking users of mobile devices doubled in 2018. Meanwhile, Kaspersky also noted that some banking Trojans have stopped attacking users of online banks on PCs.[1] These findings, combined with a sharp increase in mobile device adoption and use, point to a clear shift in the focus of fraudsters and fraud schemes. While the online channel is still under threat, malicious actors are clearly investing more time and money than ever in attacking the mobile channel.

Mobile application developers must take a layered approach to security to combat this new aggression. In addition to implementing multi-factor authentication, the app itself must be secured. Doing so requires a complete mobile application security program, which historically consisted of building security into design requirements, providing secure code training and resources to developers, performing regular automated security testing throughout the development lifecycle, and periodic penetration testing. But now, with a surge in attacks on mobile devices, applying proactive, client-side security measures such as mobile app shielding is becoming a necessity. With these security measures and multi-factor authentication in place, financial institutions (FIs) can not only defend the app against attacks but also simplify the user experience. Financial institutions can provide the most convenient authentication methods and keep the advanced mobile app security protections invisible to the user.

Fraud in any channel has wide reaching effects on a financial institution, and the mobile channel is no different. It costs money and time, inconveniences customers, and hurts the FI's reputation in the market. Existing customers are more likely to churn and potential customers will be more reluctant to commit to an organization perceived to be lax on security. In this white paper, we'll explore how app shielding with runtime protection is an essential tool in combatting attacks on the mobile channel. As the most advanced security that developers can leverage for their mobile applications, app shielding wraps around the application code to protect against malicious activity and safeguard sensitive information from cybercriminals – protecting both customers and the financial institution. Moreover, it can accelerate digital transformation initiatives, reduce operational costs, and even open up new opportunities for growth.

Current forecasts estimate that as many as 2 billion people – 50% of the entire global banking population – use their mobile device for banking services.

Exploiting the Growth in Mobile

Wherever there is a vulnerability, fraudsters and attackers will find a way to exploit it for personal gain. Today, that opportunity presents itself in the mobile channel. Users are continuing to flock to mobile devices at a rapid rate. In 2018, the number of mobile users grew by an estimated 200 million,[2] and worldwide consumer spend in app stores reached $101 Billion in 2018.[3]

Those mobile users also increasingly rely on their devices to satisfy their banking needs. Current forecasts estimate that as many as 2 billion people – 50% of the entire global banking population – use their mobile device for banking services.[4]

Clearly, this massive congregation of users on the mobile channel presents an opportunity for fraudsters to capitalize on any lapse in security they can find, and they have taken notice. As explained by Julie Conroy, research director at Aite Group, "The digital channels are without question favored by fraudsters, especially as the attack surface continues to expand. As more transaction value and volume passes through the mobile channel, we see organized crime rings intensifying their attacks as well."

Further, in an Aite survey of large U.S. financial institutions, 74% of respondents said that their online and mobile losses had increased.[5] This survey response is also supported by the September 2018 McAfee Labs Threats Report.[6] which notes that mobile malware increased 40% from Q2 2017 to Q2 2018.

Assumptions Making Mobile Even More Vulnerable

The reality of the mobile application market is that developers cannot know the environment in which their application will be used. There is always a chance that the app may be downloaded onto a jailbroken or rooted device, meaning that the operating system's default security safeguards have been removed. Therefore, there is always a risk that the banking application will be targeted by malware lurking on that device and result in fraud losses for the organization.

However, that reality is not well understood, and the risk is compounded by two incorrect assumptions commonly made by users and developers.

Assumption 1: The Apple App Store and Google Play Store Only Offer Legitimate, Secure Applications

Many assume that if an app is on the Google Play Store, Apple App Store, or other official application marketplaces, it has been scrutinized by the marketplace's approval processes and is safe for use. The belief is that the app stores function as secure gatekeepers that filter out any rogue, malicious, or unsafe applications. Therefore, so long as users stick to downloading applications found on the sanctioned app marketplaces, their device will be safe and secure. This is incorrect, and it leads to otherwise security-conscious users unwittingly compromising their own mobile devices.

Though the app stores do filter out a large percentage of malware, they are not perfect and never will be. Malicious applications exist on the app stores waiting to be downloaded so they can steal personal information, inject malicious code into the mobile device or another app, or otherwise take advantage of an unsuspecting user. For example, in October 2018, the Google Play Store removed 29 banking Trojans masquerading as simple utility apps,[7] such as device boosters, cleaners, battery managers, and horoscope applications. It was discovered that, once installed, these applications could impersonate other applications installed on the user's device and target them with personalized phishing attacks.

This issue isn't a matter of effort or competence. The Google Play Store and Apple App Store make significant investments to keep their marketplace as clean as they can. The issue is scale. There are millions of apps available for download on both the Google Play Store and Apple App Store. With so many applications on the market, it simply isn't feasible to catch all malicious applications all of the time.

Assumption 2: Android and iOS Operating Systems Provide Adequate Security

The second misconception is that the iOS and Android operating systems provide adequate security for the mobile device and, by extension, adequate security for their mobile apps. There is a kernel of truth here. The Android and iOS development teams devote substantial amounts of time to patching their operating systems to remove any vulnerability they may find. But, they are not perfect, and they cannot account for user negligence. Further, there is always a gap of time between a vulnerability being identified and a patch being released. Then, patches must first be sent to device manufacturers and mobile carriers who then release the update according to their own development schedule. Those patches are neither available immediately nor downloaded right away by users, and they may be running long outdated versions of the operating system riddled with opportunities for attackers and malicious code.

In light of these realities, the mobile devices on which an application will be running is an unknown and potentially hostile environment. The application developer must ensure the security of their own mobile app, because if they do not, there are no ironclad external safeguards to protect it. Otherwise, it will be vulnerable to exploitation.

What Does "Secure" Even Mean?

Accepting that developers need to build-in application security is one thing, but security is not a straightforward process. A thorough mobile application security program typically includes specific design requirements, regular automated security testing, and periodic penetration testing. These procedures are used to identify vulnerabilities in the application design, so the development team can patch these holes and prevent fraudsters from exploiting weaknesses.

However, one can no longer simply find and fix vulnerabilities in an application and call it secure. It is impossible to locate every vulnerability, and there would never be enough time to fix them all. Even if it were possible, finding and fixing vulnerabilities will not protect against the most advanced threats that go beyond exploiting common security gaps. Security must go further and empower the app to actively protect itself against attacks in the wild. The application must monitor its environment and take action on malicious behaviors. It can't be helpless. It can't be passive. It must be equipped to defend itself.

Developers can attempt to create this level of security on their own. One example would be an in-house development team leveraging existing, rudimentary jailbreak/root detection tools for their app that would ostensibly provide the security necessary.

However, attackers are targeting the broader mobile channel. They have a deep knowledge of mobile application security and are also aware of these rudimentary tools. Attackers have or will develop strategies to circumvent these basic technologies.

Detection systems must be constantly maintained and updated to keep current with the rapid pace of new attacks and vulnerabilities.

With that in mind, what does "secure" even mean?

OWASP Standards

The Open Web Application Security Project (OWASP), an international community of technologists, data security experts, and developers, drafted an independent baseline for mobile application security called the Mobile App Security Verification Standard (MASVS). It provides unbiased guidance on recommended security capabilities for a mobile application, depending on the function of that application. As stated in the standard, "The MASVS is a community effort to establish a framework of security requirements needed to design, develop, and test secure mobile apps on iOS and Android."

MASVS identifies four different levels of security necessary for mobile applications, because not every application requires the most stringent security measures.

For the purposes of this white paper, we are mostly concerned with MASVS L2+R, because all banking applications fall under this standard. Our larger conversation about app shielding and mobile application security will use these standards as our baseline for the security required to gain the benefits outlined in Part 2 of this paper.

MASVS SECURITY LEVELS

MASVS-L1

The minimum threshold for all mobile applications, it includes security best practices that have a "reasonable" impact on development costs and user experience.

Typical Applications Include:

- Fitness or meditation apps
- Simple tool applications like a flashlight

MASVS-L2

For applications with access to information or capabilities that can be leveraged by attackers for fraud, such as personally identifiable information, credit card numbers, or the ability to move funds.

Typical Applications Include:

- Healthcare applications
- Apps with subscription services

MASVS L1+R

The "R" in the MASVS levels refers to protections against client-side attacks or reverse engineering. Therefore, this level applies to applications that do not have access to sensitive user information, but still must be resilient against tampering and reverse engineering.

Typical Applications Include:

- Competitive gaming apps (to prevent cheating)

MASVS L2+R

The strictest security standard is reserved for apps that require access to sensitive user data, but also must be resilient against reverse engineering and made available to a large consumer base using a wide range of operating systems and devices.

Typical Applications Include:

- Banking and financial services applications

Prevailing Mobile Attack Strategies

The MASVS security standards are designed to protect an application against a wide range of threats. Understanding the manner in which these attacks take place is essential to securing a mobile application.

Below, we've included a brief summary of the leading attack strategies and the means by which these attacks circumvent traditional security measures.

Mobile Threats Explained:

A L2+R mobile application must be fortified against:

Reverse Engineering

An attacker may reverse engineer an app as the first step in a number of attack strategies. Consider it reconnaissance before the targeted strike. Adversaries may reverse engineer an app to analyze its source code and component parts to gather information that can be used to develop malware that exploits the app's operation, or to tamper with the app.

For example, attackers might deploy their own malicious app designed to exploit vulnerabilities discovered by reverse engineering the banking app. If a user has both applications on their device, the malicious app can redirect banking deposits to the attacker's account without the user realizing there was ever a breach.

Repackaging

Repackaging attacks start with an attacker reverse engineering an application, tampering with the source code, and republishing the tampered app back onto unofficial marketplaces. To a user, it will appear as though they downloaded the correct application. In all respects, the app will appear to the user as the legitimate application, but behind the scenes, there is code stealing personal information, redirecting funds, or performing other malicious activity.

Overlay Attacks

Overlay attacks consist of an attacker-generated screen opening on top of the legitimate application UI. To the user, it will appear as a normal experience within the app, but in reality, they will be entering sensitive information, such as usernames, passwords, credit card numbers, or other personally identifiable information, into a form controlled by the attacker. This overlay window is then instructed to deliver whatever information is entered into it directly to the attacker. Unbeknownst to the user, they have just handed over their information.

In addition to hijacking data entry, overlay attacks are used to trick (or socially engineer) users into installing other malware or performing insecure tasks on their mobile devices, like granting a malware app full control of the user's phone.

Rogue Keyboards

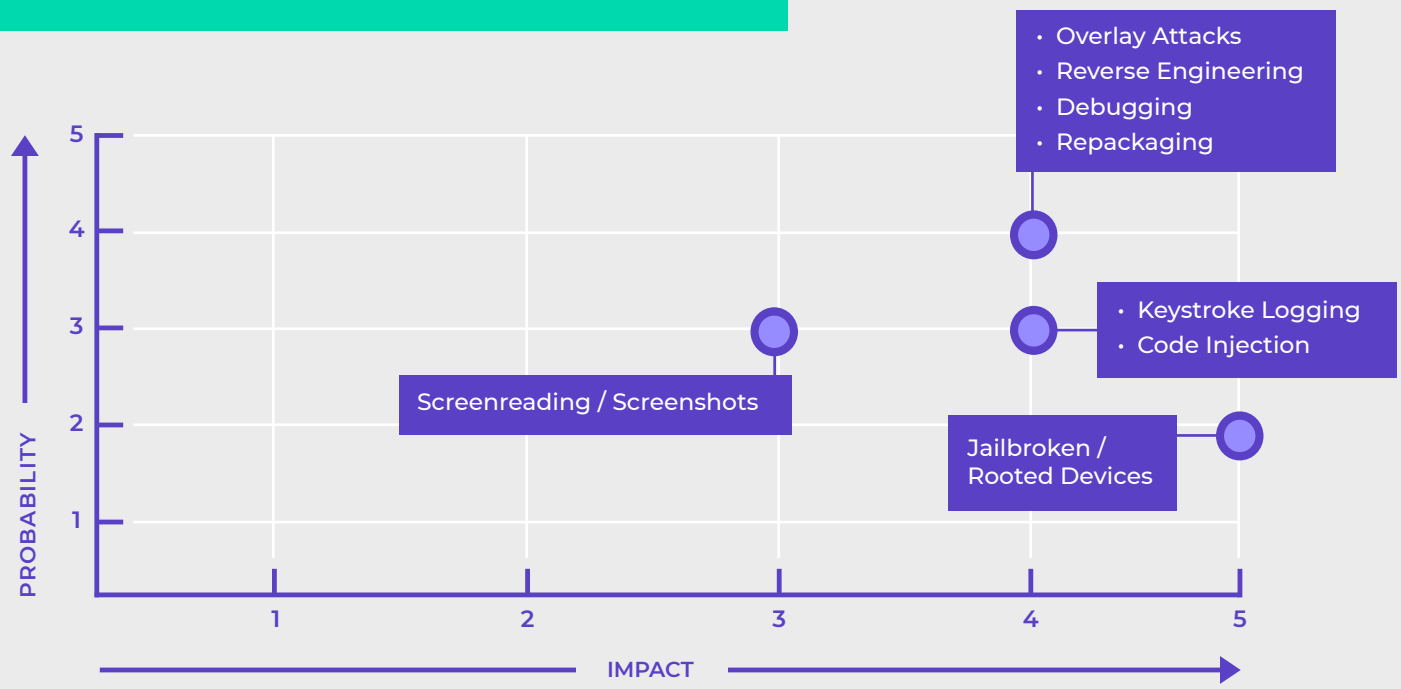
The app marketplace is full of alternative keyboard applications to replace the native keyboards installed on mobile devices. Typically, users download these applications in an innocent attempt to personalize their device.

They may like the color of the new keyboard. They may prefer its functionality. In any case, many of these keyboard apps are completely benign, but some are known as rogue keyboards. These apps have code operating in the background to steal personal information or carry out other malicious activity.

Threat Probabilities

How a financial institution ranks the following threats or attacks will depend on their unique risk profile or threat model, but here are industry estimates regarding the probability and impact of these various risks. The estimations were determined based on analyst and customer consultations, as well as industry media coverage.

PROBABILITY AND IMPACT OF TOP CYBER THREATS



Security and the Development Schedule

If developers had endless time, budget, and a specialized security team in place, every mobile application on the Apple App Store or Google Play Store would be as impervious as possible to known attack strategies. Unfortunately, applications are not developed in a vacuum. There are development schedules to keep, budget limitations, and the pressures of competition to take into account. From multiple fronts, organizations face tremendous pressure to release their application as soon as possible before a competitor seizes on the opportunity first.

To put this in perspective, consider the number of apps on the two major marketplaces. In January 2019, the Google Play Store and Apple App Store had 2.88 million and 1.95 million applications respectively available for download. Further, there is an average of approximately four thousand apps published on the Google Play Store per day.[8]

This level of pressure does not align well with multiple rounds of security testing and fine-tuning necessary to ensure a stable, secure mobile application.

On the other hand, financial institutions are, in some instances, too wary of the mobile channel. They recognize the danger of offering an insecure application or the negative publicity that can come from a high-profile breach. As a result, many choose not to pursue certain functionality in their mobile app. Though this approach exposes the company to less risk, they essentially concede the fight to the competition and risk losing customers as a result. In today's market, financial institutions must make bold moves to attract and retain customers.

In this environment, mobile application developers are backed into a corner. The market will not allow them to delay and provide adequate security. Meanwhile, moving forward without doing so is a risk many financial institutions are unwilling to take. This dynamic traps developers in a difficult situation.

PART 2: ENTER APP SHIELDING WITH RUNTIME PROTECTION

App shielding with runtime protection offers proactive defense against zero-day and other targeted attacks. This allows mobile financial service applications to run securely, block foreign code from working, or shut down the application if a threat to data exists. Integrating app shielding with runtime protection ensures the complete integrity of the app and fully protects sensitive information from cybercriminals – even on untrusted mobile devices.

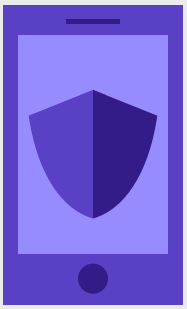
App shielding with runtime protection serves as an excellent counterweight to the challenges and forces impacting the mobile channel. It's fast, invisible to users, and quickly integrates into the application. It is the most advanced security developers can leverage for their mobile apps, and they don't need the in-house expertise to develop, deploy, and maintain it. Even a non-technical professional, assuming they understand which security policies are important, could layer app shielding into the mobile application.

App shielding wraps around the application code to protect against malicious activity. Even if a device becomes infected with malware (including system components such as a screen-reader or key logging on Android), app shielding will detect and prevent that code from running.

Most importantly, app shielding can accomplish these feats with minimal impact on the development schedule.

Mobile App Shielding Capabilities

INTRUSION	ANDROID	iOS
App Impersonation	• **Detect** repackaging	• **Detect** repackaging
Code Modification	• **Detect** Java hooking framework and native code hooking	• **Prevent** runtime library injection and execution
Data Leakage	• **Prevent** user and system screenshots • **Detect** untrusted keyboard	• **Prevent** system screenshots • **Detect** user screenshot • **Prevent** keyboard cache
Debugging	• **Prevent** Java and native debugger	• **Prevent** debugger
Emulator	• **Detect** emulator	
Privilege Escalation	• **Detect** root	• **Detect** jailbreak
Reverse Engineering	• **Prevent** with obfuscation, whitebox cryptography, and more	• **Prevent** with obfuscation, whitebox cryptography, and more
UI Overlay (overlay attacks)	• **Prevent** foreground override	



Mobile App Shielding vs Runtime Application Self-Protection (RASP)

Some vendors use the terms “RASP” and app shielding interchangeably, but there is a difference. RASP refers specifically to server-side technology that allows a server-side application to protect itself against runtime attacks.

App shielding (sometimes also paired with "app hardening") refers to a collection of technologies that ensures the integrity of a client-side app and makes the mobile app more resilient against run-time attacks, reverse engineering, and more, in potentially hostile environments (such as unmanaged mobile devices). Because this white paper explains the business case for fortifying mobile apps against client-side attacks, we use the term app shielding.

How It Works

App shielding with runtime protection ensures the integrity of mobile apps in three ways:

- **Protect**

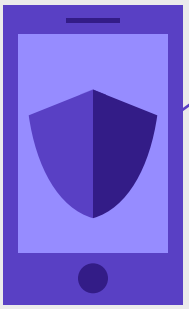
  App shielding protects the mobile application by preventing reverse engineering techniques via code obfuscation and whitebox cryptography.

- **Detect**

  App shielding actively detects malicious key logging, screen readers, repackaged applications, debuggers and emulators, and jailbroken or rooted devices.

- **React**

  Once an attack is detected, app shielding can then react to prevent screenshots, thwart malware, block screen duplication, or enable customized actions based on business policy (i.e., application shutdown).



1

Protect

Reverse Engineering Protection

- Code Obfuscation
- Asset Encryption

2

Detect

- Screen Monitoring
- Code Injection
- Emulators
- Screen Readers
- Jailbreak/Rooted Devices
- Overlay Attack
- Repackaging
- Screenshots
- Key Logging
- Debuggers

3

React

- Alert and Block

What Is Whitebox Cryptography?

Because mobile apps operate in untrusted, potentially hostile environments (a mobile device under an attacker's control, for example), it is important to protect the encryption keys within the app. If an attacker were to extract an encryption key, much of the app's security collapses.

Whitebox cryptography is used in app shielding solutions to prevent an attacker from uncovering the encryption keys, using a combination of encryption and obfuscation.

The Business Case for App Shielding with Runtime Protection

Financial institutions have three metrics that impact their decision when investing in the security of their applications: time to market, budget, and available resources. App shielding easily answers the time question. The wraparound security of app shielding with runtime protection saves developers weeks or months of coding time trying to build advanced protections from the ground up. In addition, developers not specialized in application security cannot keep up with the evolving landscape of security vulnerabilities and attacks.

To make the business case for app shielding (with runtime protection in particular), the technology should be evaluated across four key areas:

- Revenue growth
- Revenue retention
- Cost reduction
- Cost avoidance

When analyzed through this lens, the business case for app shielding with runtime protection becomes clear.

Revenue Growth

Technologies and solutions have the potential to open up new revenue streams for the business or financial institution. In the case of a security solution, such as app shielding with runtime protection, it is more about the revenue streams that the security solution can facilitate rather than directly create. Without adequate security, there are functionalities in the mobile channel that are deemed simply too risky to pursue. They are fraught with opportunities for fraud and abuse, so it makes financial sense in those instances to avoid offering the service.

Peer-to-peer payments, for example, are an extremely attractive functionality for the modern banking customer. More than attractive, users expect their banking application to be able to support peer-to-peer payments. The advanced security capabilities found in app shielding with runtime protection can securely enable peer-to-peer payments in the app. Through this dynamic, app shielding is indirectly, but indisputably, contributing to the revenue growth that would result from enabling that feature.

It is also worth noting that an application secured by app shielding with runtime protection will enable a development team to innovate. The knowledge that the application is protected from known threats (and that the vendor's team is constantly working to strengthen that security) allows a development team to experiment safely with new services and functionality, thereby opening up new revenue streams for the organization.

In addition, security plays a role when consumers choose which financial institutions to bank with. Even a single breach can persuade current customers to switch and dissuade prospective customers from choosing the breached institution in the first place.

Key Considerations for the Reader:

- On average, how many new accounts per month are opened with your institution?
- Are there features you are not adding to your application, because they have been deemed too risky?
- Do you deny mobile services to users with jailbroken or rooted phones?

Revenue Retention

Reducing friction in the customer experience is one strategy in improving revenue retention. A superior customer experience leads to greater customer satisfaction, loyalty, and retention. A key area to remove friction is at the authentication step. For decades, the market has relied on authentication methods such as hardware authentication devices. While these provide a reliable method of proving a user's identity, they also introduce friction by requiring mobile users to carry an additional device everywhere they go.

Today, however, many organizations choose to transition to software for multi-factor authentication, whether as a standalone authentication app or natively integrated into their banking app. We are seeing this transition across the industry for a variety of reasons. Most importantly, it offers the convenience and ease users want and expect from their mobile banking experience. At the same time, software authentication cuts deployment and support costs.

These benefits are enough for many financial institutions to make the switch, but security becomes all the more important once the authenticator is integrated with the mobile device. App shielding with runtime protection enables a financial institution to transition to software authentication, improve the customer experience, and do so while managing the risk such a change incurs.

In terms of reputation, customers who feel that their mobile banking application is secure and reliable will be more likely to use the mobile app to conduct larger transactions and more frequently. Banking customers carry their mobile devices, and therefore their banking applications, on them at all hours of the day. A strong reputation for security paired with the convenience of the digital age will result in mobile customers who regularly use their mobile banking apps for their daily transactions.

Key Considerations for the Reader:

- How many accounts does your organization serve? What is the average revenue per account?
- How often do you lose a customer who falls victim to fraud?
- If you knew your app was more secure, what steps could you take to increase customer engagement and drive more and higher value services utilization?

Cost Reduction

The goal of cost reduction is for the organization to use the resources they already have, but reduce their operating cost while maximizing their impact. App shielding with runtime protection helps deliver this value. To illustrate, consider the dynamics at play in a typical release schedule when the mobile app developer brings their application security in-house.

Mobile application developers only have so many hours in the day and so many days in the development schedule. Their production power is a finite resource that should be spent making the application the best it can be. They should be improving customer experience, expanding the app's functionality, and ensuring that the product remains competitive in the market. This is their area of expertise.

At the same time, the application has to be protected against the multitude of attack strategies laid out in Part 1 of this paper. Who will take on this role?

The development team is a logical first choice, but unfortunately, the organization will need to divert the time and efforts of the development team towards security. This is problematic for two reasons:

- First, security is not the core competency of the development team. Even if they can add some security to the application, security is not their specialty. Their role is to make the application as efficient and effective as possible. Security falls outside that area of expertise, and as such, the app could still be at risk.
- Second, security is not a switch to flip on and forget about. The threat landscape is constantly changing, and the security solutions we depend on need to adapt along with it or become obsolete. Relying on the development team for in-house security will permanently saddle them with a task outside their core competency.

If the development team is not the answer, an alternative would be to recruit an in-house team of security experts to build and maintain the security of the mobile application. Some enterprise organizations and financial institutions take this route, but they have the infrastructure to support a substantial investment in building an effective team of experts. Even still, this approach comes with challenges of its own.

Hiring in the security space is difficult. The demand far exceeds the supply, and the gap is growing larger. According to the 2017 Global Information Security Workforce Study, "we are on pace to reach a cybersecurity workforce gap of 1.8 million by 2022, a 20% increase over the forecast made in the 2015."[9] Moreover, iOS and Android operating systems have different approaches to security. The organization would need to find and employ security experts in both operating systems in order to secure the application. Furthermore, the organization runs the risk of the hiring, onboarding, and training process slowing down the development schedule as a whole.

Because of the security workforce gap, the few security experts on staff are often stretched thin. Development teams already outnumber the security team at a rate of 100:1.[10] If an in-house security team is going to keep pace, they will need a solution, such as app shielding, to apply advanced protections to the application.

Finally, security teams and policies are becoming more tightly woven into DevOps practices (i.e., DevSecOps). Developers will appreciate that some app shielding solutions will not slow them down or frustrate them. OneSpan App Shielding, for example, easily plugs into DevOps pipelines as another automated process that accelerates app delivery through integrations with Jenkins and other popular tools that automate and accelerate app building, testing, and deployment.

Key Considerations for the Reader:

- How do you stay up-to-date on the latest mobile app threats and implement protections against them? Do you have that expertise on staff? Could your time be better spent elsewhere?
- How much do base salaries, benefits, bonuses, and salary increases for your development team cost each year?
- How much of developers' time can you sacrifice to security vs. adding new features?

Cost Avoidance

While cost reduction is about reducing existing expenses, cost avoidance is about preventing a predictable but unknowable future cost.

A perfect example of this kind of cost is fraud. It is certain to occur at some level every year, but the form it takes and its impact are variables a financial institution can influence. Increasing the security effectiveness of a mobile banking application is a sure way to reduce not only fraud in the mobile channel, but overall fraud as well.

While every financial institution experiences fraud, some claim they are not experiencing a significant amount in the mobile channel. That may be possible, but it is more likely that the organization is just unable to track it appropriately.

For example, just because banking credentials are stolen in one channel doesn't mean they will be used for fraud in that same channel. A mobile overlay attack is designed to capture a user's login credentials as they naturally input it into the phone. The attacker could, of course, then use that information to commit fraud directly through the same mobile application. However, they could just as easily record the banking credentials and infiltrate the online channel. To the financial institution, this appears to be a clear-cut case of online banking fraud, but it actually originated in the mobile channel through a vulnerable app.

Much like the other dimensions in our analysis, reputation plays a role in cost avoidance as well. Adding app shielding with runtime protection to the mobile app reduces the risk of reputational damage in the future, thereby avoiding the costs associated with that reputation damage. Reputational costs can take many forms. At one end of the spectrum, a grand-scale, high-profile breach could dramatically affect revenue, cause customer churn, and require significant efforts to repair.

But, reputational hits are not always so dramatic. They happen on an individual level as well. Each time a customer experiences any sort of security incident, the associated financial institution's reputation suffers in the eyes of that customer.

Key Considerations for the Reader:

- What are your total annual fraud losses? In your estimation, what percentage are directly attributable to the mobile channel?
- How much might a publicized security incident cost your organization?
- What regulatory requirements apply to your mobile app? Is there a risk of non-compliance fines or penalties?

Conclusion

Tackling the multi-faceted challenge of developing a successful mobile banking application is no easy feat, and development teams must contend with pressures from every direction. It is imperative to get an application built, tested, and published as quickly as possible. However, in the rush to market, security cannot be overlooked. When releasing a mobile app, one cannot be sure who will download it or the conditions of the device on which the app will be used. If a device is compromised, the application is at risk.

App shielding with runtime protection mitigates risk with proven, reliable security that can fit into already tight budgets and production schedules.

OneSpan, for example, employs teams of security experts in both iOS and Android security to constantly monitor the mobile threat landscape. We remain competitive in the market through our ability to identify new attack strategies and update our security solutions to prevent them. It is our commitment to deliver a product that combines effective, state-of-the-art security while ensuring the best possible customer experience, so you can focus on your business goals.

Works Cited

[1] https://securelist.com/ksb-cyberthreats-to-financial-institutions-2019-overview-and-predictions/88944/

[2] App Annie, The State of Mobile 2019, January 2019

[3] https://www.appannie.com/en/insights/market-data/app-annie-2017-retrospective/#download

[4] Juniper Research, Futureproofing Digital Banking 2018, March 2018

[5] Aite Group, Digital Channel Fraud Mitigation: Evolving to Mobile First, November 2017

[7] https://www.appannie.com/en/insights/market-data/app-annie-2017-retrospective/#download

[6] https://www.mcafee.com/enterprise/en-us/assets/reports/rp-quarterly-threats-sep-2018.pdf

[8] https://42matters.com/stats

[9] https://iamcybersafe.org/wp-content/uploads/2017/06/Europe-GISWS-Report.pdf

[10] https://www.sonatype.com/2018survey

OneSpan

OneSpan enables financial institutions and other organizations to succeed by making bold advances in their digital transformation. We do this by establishing trust in people's identities, the devices they use, and the transactions that shape their lives. We believe that this is the foundation of enhanced business enablement and growth. More than 10,000 customers, including over half of the top 100 global banks, rely on OneSpan solutions to protect their most important relationships and business processes. From digital onboarding to fraud mitigation to workflow management, OneSpan's unified, open platform reduces costs, accelerates customer acquisition, and increases customer satisfaction.

CONTACT US
For more information:
info@OneSpan.com
OneSpan.com