The Internet Banking [in]Security Spiral

Past, Present, and Future of Online Banking Protection Mechanisms based on a Brazilian case study

Marcus Botacin[1], Anatoli Kalysch[2], André Grégio[1]
[1]Federal University of Paraná (UFPR-BR)
[2]Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU-GER)
{mfbotacin,gregio}@inf.ufpr.br,anatoli.kalysch@fau.de

ABSTRACT

Internet Banking have become the primary way of accessing banking services for most customers, but its security is still a constant concern, since million dollars are still lost every year due to frauds. Over time, banks and customers overcome the initial technology distrust and learned how to secure their operations. However, there are still many lessons to learn, mainly when looking to the upcoming technological developments. To understand the lessons learned over time and also to help shedding light on possible future developments, we review the past and the present of internet banking implementations in Brazil, a country widely adopting this type of service and an early adopter of new banking technologies, thus targeted by many threats. We show how Internet banking evolved from desktop software to mobile apps and how attackers also evolved from phishing mails to complete phishing applications to target Brazilian users. We also performed a detailed security analysis of Brazilian banking apps available in the Android app store and identified that developers still fail to follow secure development practices, thus causing banking apps to leak user's sensitive data. Moreover, we also looked to the future to present new attacks which can threat users in a short-term. In particular, we demonstrate an attack against a Whatsapp-based transaction mechanism implemented by some Brazilian banks.

CCS CONCEPTS

• **Security and privacy → Mobile platform security**; **Penetration testing**;

ACM Reference Format:
Marcus Botacin[1], Anatoli Kalysch[2], André Grégio[1]. 2019. The Internet Banking [in]Security Spiral: Past, Present, and Future of Online Banking Protection Mechanisms based on a Brazilian case study. In *Archived version of the paper included in the Proceedings of the 14th International Conference on Availability, Reliability and Security (ARES 2019) (ARES '19), August 26–29, 2019, Canterbury, United Kingdom.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3339252.3340103

Archived version of ARES '19, August 26–29, 2019, Canterbury, United Kingdom

ACM ISBN 978-1-4503-7164-3/19/08…$15.00
https://doi.org/10.1145/3339252.3340103

1 INTRODUCTION

Internet Banking is currently the primary way of accessing banking services for many customers—more than 51% of all US bank customers use online banks since 2013 [6]. A recent report shows that banking apps are the most downloaded ones of all available categories in mobile phones' app stores [43]. Despite the wide adoption of Internet Banking, its security is a constant concern, since millions of dollars are lost every year due to attacks and frauds (e.g., in the first half of 2018, UK banks reported 500 million pounds in losses due to scams [27]). However, banks and their customers have overcome the initial technology distrust [26] and started doing reasonably secure operations. On the one hand, a lesson was learned: security-awareness campaigns leveraged by many banks are already a reality [17, 48]. On the other hand, there still are plenty of unlearned lessons, mainly if we look into the upcoming technological development, such as the mobile banking widespread use and social media-based (e.g., Whatsapp) transactions.

It is certain that these new technologies will pose additional challenges on security and usability to both banks and customers, and we need to be prepared to deal with their emerging problems. Thus, to understand the lessons already learned over time, as well as to help shedding light on possible future developments regarding Internet Banking security, we review the history of Internet Banking in Brazil, a country that widely adopts this type of service (≈76% of bank customers [41]) and which is characterized by the early adoption of banking technologies [45], thus presenting many particular threats (e.g., Brazilian-focused boletos' malware [13]). Banking frauds can be analyzed either by the technical aspects that cause them (e.g., captcha resistance [39] and untrusted environments execution [46]), either by their related social, cultural, and contextual aspects, such as the sources (e.g., Brazilian banks computerizing their systems in the early 80's to keep up with a period of high inflation [45]) or consequences (e.g., financial losses caused by malware [3]) of using Internet Banking technology. Previous work investigated each of these aspects in isolation, i.e., the literature did not consider yet both aspects to try to associate the causes and consequences involving the widespread of Internet Banking technologies by a massive number of unskilled users.

In this work, we investigate how the technical and social aspects of Internet Banking act together to drive the technology development. Our goal here is to discover and analyze correlations between social-cultural aspects (of users) and project decisions adopted in Internet Banking technology (taken by its developers and attackers). For instance, we investigate how the current fin-tech's paperless culture imply in the reduction of mobile app's attack surface by eliminating the need of leveraging native code libraries for documents

digitalizing. We also show how Internet Banking evolved from software completely developed to desktop use, which was popular by the time Brazil started to adopt online banking, to the current mobile apps based on in-browser clients. In addition, we show how attackers responded to that and evolved their *modus operandi* from naive phishing e-mail messages to full phishing-based applications that mimic banks' client apps. We also conducted a detailed security analysis of current Brazilian Android banking apps, since they are the most used method to access Internet Banking services. This analysis allowed us to observe that developers still fail to follow basic secure development guidelines, causing banking apps to leak users' sensitive data: we identified that some apps store credit card information locally instead of in remote servers, which allows for information retrieval through backup procedures; other apps make use of non-pinned certificates, which turn them vulnerable to man-in-the-middle attacks.

Finally, we discuss new attacks that may threat users in the short-term. In particular, we demonstrate the feasibility of attacking a novel chatbot-based banking mechanism that allows users to perform transactions by exchanging Whatsapp messages. To do so, we exploited the fact that the Whatsapp client does not provide the same security level of online banking apps by coding a malicious accessibility service. This service acts like a bot and sends messages to the bank chatbot on behalf of the user, injecting commands that actually commit sensitive operations to the victim's account. In summary, we present the following contributions in this work:

1. We review the Brazilian banking ecosystem, investigating how past historical facts shaped project decisions regarding electronic banking services implementation, and how future applications may also be shaped by current social-cultural trends.

2. We evaluate the implementation of current Brazilian mobile banking apps, focusing on their security. We discuss the vulnerabilities and possible forthcoming threats that may arise due to the issues found, and show a proof-of-concept attack to a new Whatsapp-based Internet Banking service provided to mobile users.

3. We discuss the lessons learned based on our observations and provide general guidelines to help the development of more secure Internet Banking apps and services to online banking users.

The remainder of this paper is divided as follows: In Section 2, we revisit the adoption of Internet Banking in Brazil to discuss past advances and current challenges; in Section 3, we present our methodology for analyzing Android banking apps; in Section 4, we present our results on Brazilian Android apps security; in Section 5, we propose some guidelines to assess the safe development of an Internet Banking app; in Section 6, we discuss this work's implications; in Section 7, we present related work to better position our contributions; finally, we draw our conclusions in Section 8.

2 MOTIVATION: A BRIEF HISTORY OF INTERNET BANKING EVOLUTION

Due to the massive dependency of people in banks, the latter have been leading the automation of services and procedures. Consequently, banks heavily invest in the adoption of new technologies since their conception. Hence, if we observe the broad type of technologies leveraged by banks along time, we are able to understand how their adoption shaped financial-related social and cultural relationships around the world. The produced knowledge includes the reasons behind some technologies ubiquitous adoption, the kind of threats that exploit their weaknesses, and the possibility to infer future menaces based on cyclical historical movements.

The adoption of new technologies and their associated risks are closely tied to the specific aspects of a country and its culture. In this paper, we delve into the peculiarities of the Brazilian banking system, due to the early adoption of technology, as well as the particular threats that arise in consequence of this fact. Our investigation intends to allow for (i) Brazilian banks to learn how to secure their online banking systems in a better way, with inputs from the overall community, and for (ii) the global community to take advantage of successful cases of technological development/deployment about online banking protection and usability while learning from discovered flaws.

2.1 The Brazilian banking system

Brazilian banking system was computerized in the early 80's due to a period of high inflation [45], which makes Brazil to be the first country in the world to report many types of attacks against online banking technologies, such as the boletos' malware. In addition, given its huge population, Brazil was also the first country to show the large scale "exploitability" of many attack vectors—e.g., credit card cloning [40], which is noticeable to the Brazilian banking system, since credit cards are very popular in the country (used by 77% of Brazilians in most transactions [12]). As credit cards are used many times a day in multiple card reader machines, the attack opportunity window for sensitive data collection by a malicious entity is largely wide. For instance, one of the most ordinary strategy used in Brazil to steal credit card information is to add an extra hardware over the ATM card reader to physically read data from the credit card when it is inserted in the machine. This strategy, dubbed skimming [42], may cause millions of dollars in losses. Given the amount of cloning-based crimes, transactions that relied on a person's signature quickly became outdated, which resulted in the adoption of a chip in the credit card unlocked by a PIN (Personal Identification Number) in Brazil already in the 90's [51], instead of the infamous stripes.

As a consequence of the mandatory use of chip+PIN credit cards, Brazilian attackers shifted their strategy from card cloning ATM hardware to deceive users into providing their credentials (credit card number, expiration date, CVV, and PIN) through social engineering, which can be launched from (and aiming at) many distinct platforms. Therefore, the majority of attacks to banks could be performed remotely, targeting the weakest link of the chain: the user of Internet Banking, which is barely aware of online threats (and/or does not browse in safe manner), not skilled enough to properly set up security mechanisms in his own personal computer, and highly prone to fall in all sort of scams in which an impersonated "bank entity" warns about the risks of not providing his sensitive information by filling a mail messages, Web form or executable GUI fields immediately. Below, we provide this scenario's context.

The first platform to support online banking operations in Brazil were MS Windows desktop computers, whose bank applications used to be mere proxies: they were based in forms to be filled at

the client-side and then submitted to the banks servers. To protect the transaction, the application requested tokens from a One Time Password (OTP) or pattern table. Attackers then started to apply basic social engineering scams by luring users into running malicious attachments included in alarming phishing messages and, consequently, providing all sorts of sensitive information.

Despite being the main method to access a bank through the Internet in the 2000's, desktop-based applications rapidly became outdated with the "Web boom": all banks that operated online moved their electronic facilities to multiplatform, Web-based applications. Online banking services accessed through the bank website were first compromised by keyloggers stealthily installed on the victim's side to collect credentials. To fight keyloggers, Brazilian banks decided to demand the installation of a Web browser plugin for protection, which would ensure that the system remained in a safe state [11]. In practice, the plugin was an antivirus-based solution responsible for detecting suspicious processes' behaviors, such as API hooking and proxy configuration. As a reaction, attackers moved back to social engineering campaigns, but now sending phishing emails with embedded links to cloned Web pages with forms for credentials collection.

Attackers realized that committing only electronic crimes were not enough to explore the potential of a huge, lucrative "market" as Brazil, in which inequalities prevent many people from owning a bank account (or having Internet access). Therefore, attackers found out a way of also targeting offline banking customers by exploiting an exclusive Brazilian payment system called **boleto**. A "boleto" consists of a printed promissory note with a bar code readable by ATMs, as illustrated in Figure 1, and can be used to enable a client to pay for everything, from electric and water bills to appliances of department stores. Since the payment of a boleto is a money transference transaction from one bank account (the debtor) to another (the creditor), it is impossible to undo it.

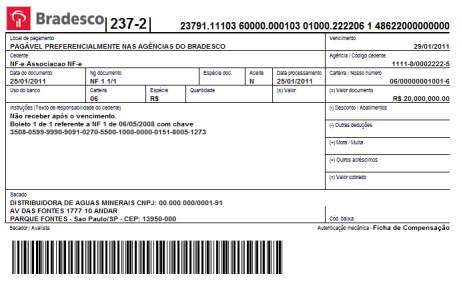

Figure 1: Brazilian Boleto. The barcode stores all payment information and can be scanned by ATMs.

Banks then shifted their attention to smart devices and started to offer services via mobile apps, including physical documents scanning, such as checks and boletos. In consequence, they once again increased users' attack surface, but this time mixing threats both to digital and physical documents. Due to Brazil's latest developments supported by last decade economic increase, now more people own a mobile device able to adopt online banking than the amount of people previously able to afford a desktop computer. These new users skipped steps in the electronic banking technology evolution, being more prone to fall for well-known scams. Currently, 58 million Brazilians access Internet solely through their mobile devices [32], and the effects of such massive users' population are unclear. Due to these facts, and to better understand the opportunities and risks posed by the novel interactions caused by the technology shift, we focus on mobile online banking apps.

In parallel to the aforementioned Internet Banking technological changes, the latest trending entities that support online banking evolution are the "fin-techs", financial institutions structured in the startup model whose goal is to reduce operational costs and customer taxes, such as NuBank in Brazil [19]. Fin-techs typical operation model includes the lack of physical agencies, all-digital communication with the client, and mobile app-based interaction. While the absence of paper records is good for clients privacy, since it cannot be stolen or lost due to mail issues, all-digital records raise additional privacy concerns, which imply in careful considerations on apps' development, data transfer, and information storage. In this work, we study the project decisions behind the most popular Brazilian fin-tech app to shed some light about how the paradigm shift creates new challenges for app's implementation, such as not requiring native code libraries for handling scanned documents, which may reduce attack surfaces. Last, but not least, the Brazilian Internet Banking scenario points toward the adoption of independent, external third-party apps that interacts with the official apps provided by bank institutions. One recent example is the enabling of Whatsapp-based transactions [52] by one of the major Brazilian banks. In this paper, we discuss the security impact of outsourcing banking transactions to a third-party entity.

3 METHODOLOGY

In this section, we present our criteria for apps selection, the analysis approach, and ethical considerations for vulnerability disclosure.

Target Applications Selection. Our investigation of current Internet Banking vulnerabilities only embraces mobile apps, since they are the current interface for interaction between banks and customers. We limited our scope to Android apps, because this platform dominates the mobile platform market-share. In Brazil, 78% of all credit operations are concentrated in only five banks [10]. Therefore, we selected these banks' apps for analysis, as it would allow us to present a representative view of the vulnerabilities that target most of the Brazilian banks customers. If some bank institution provides more than one app, we selected the main app for our analysis procedures (although secondary apps might be referred in this paper to exemplify banks apps diversity). We also analyzed the main app of Nubank, a Brazilian fin-tech company that has been leading this segment in Brazil [18]—its operations increased to the point of being comparable to the major Brazilian banks. The inclusion of a fin-tech app in our evaluation allows us to compare and reason on project decisions supported by two distinct corporate cultures (consolidated banks and fin-tech startups).

Experimental Approach. The nature of our case study is an experimental approach that results in a security assessment of the Android apps. We downloaded all applications from the Google Play Store in the end of November 2018. We relied upon the OWASP mobile

security testing guide [44] as a reference framework for our investigation, since it is mature, broad, and acknowledged in industry and academia. From these guidelines, we selected the most impacting testing categories: (i) `Testing Code Quality and Build Settings`, such as enabling debug flags in a release app version or including unused libraries; (ii) `Testing Secure Data Storage` of at-rest data on the file system or in-transit data available through IPC mechanisms, such as storing data in plain text or allowing critical app data to be included included in backups; (iii) `Testing Network Communication`, such as apps being vulnerable to man-in-the-middle attacks and not implementing certificate pinning; (iv) `Resilience Against Reverse Engineering`, such as not avoiding running on non-trusted environments (e.g., emulated or rooted environments) or allowing application code to be repacked; (v) `Testing Local Authentication`, which includes checking for missing UI and accessibility-based protection [20, 34], such as inadequate checks for active UI overlays, and accessibility related information leakage of personally identifiable information (PII).

We manually checked for misconfigurations and vulnerable constructions via static APK inspection and APK decompilation procedures. We also debugged all applications to identify their anti-analysis protections and insecure storage vulnerabilities. We checked for insecure network communication vulnerabilities by implementing a man-in-the-middle attack against a smartphone and attempting to login into a bank account using random credentials (we do not have bank account in all evaluated banks). We considered the attack as successful when the inputted credentials were present in the network traffic logs. We checked for tampering protection by executing the apps in modified environments (emulated and rooted smartphones). We also manually patched all apps' login views at `smali` code level and checked their execution in multiple (stock, rooted and emulated) environments. We considered the attack as successful when the modified login screen was displayed. To test accessibility safeguards, we collected the accessibility events available during the login and sign-up procedures and vetted them against the inputted credentials. If these events contained the credentials, we deemed them vulnerable. As for UI overlay based attacks, we created overlays over the login fields to simulate an attacker that tries to get a hold on the user credentials. If an application was unable to detect those overlays or was still usable despite an overlay's presence, we identified the app as vulnerable to UI-based attacks.

Toolset. To conduct our experiment, we performed both automated and manual analysis passes. We used the automated vulnerability scanners `QARK v0.9`[1], `dexdump v2007`[2], and `AAPT v2014`[3]. The manual analysis required the disassembly, decompilation, and the recompilation of the APK files, which was performed with `apktool v2.0`[4] and `jadx v0.9`[5] tool. Network communication analysis was performed with the help of an interception proxy (`mitmproxy v2014`[6]) and `wireshark v2.6.6`[7]. UI-related attack surfaces were vetted through custom scripted attacks with full-screen overlays [20], accessibility-based attacks [33, 34, 37], and screen recordings [34]. We focused on attacks that had the potential to leak personally identifiable information (PII), e.g., user credentials, financial information and 2FA tokens.

Vulnerability Disclosure. We communicated any discovered vulnerability to the developers. Since some of these vulnerabilities posed an active threat to the banks customers, we also provided possible mitigation and resolution strategies in our communication to facilitate a fast mending process. The agreed time frame for the implementation of fixes to the vulnerabilities was 30 days, to prevent risks for customers due to their disclosure.

4 BRAZILIAN APPS SECURITY

In this section, we present an overview on Brazilian banking apps implementation. First, we show how historical facts still impact on project decisions. Second, we show existing vulnerabilities in the Brazilian apps already exploited in other banking contexts. Finally, we show threats that may be exploited in the near future.

Apps Overview: History shaped Apps development. Our primary goal here is to understand how history shaped applications development. After summarizing Apps characteristics (Table 1), we discovered that banks migrated to the mobile environment before learning the peculiarities of this new environment. The `Caixa` app, the oldest one among the analyzed apps, illustrates that: it uses a mobile version of the same protection plugin as previously used in the desktop-based apps protection (described in Section 2). As far as we know, embedding a small AV within Internet Banking apps is a feature only seem in the Brazilian scenario. Newer apps, in turn, implement other protection mechanisms, such as native checks.

Table 1: Analyzed Apps Summary. Distinct project decisions between banks and fin-techs apps, and along time.

App	App Version	Android Version	Banking Plugin	Native Lib	Java Lib
BB	6.25.1.1	7.1.1	✗	✓	✗
Bradesco	3.2.28	6.0-27	✗	✓	✗
Caixa	2.0.3	5.0.1	✓	✓	✗
Itau	6.1.4	7.1.1	✗	✓	✗
Santander	6.3.2.7	7.0	✗	✓	✗
Nubank	4.19-0	7.1.1	✗	✗	✓

Our second finding is that all Brazilian banks' apps implement OCR and scanning procedures, incurring in the inclusion of native code libraries as part of app's trusted code base (e.g., `opencv`). It came from the time lapse regarding the adoption of information systems by banks (very early in Brazil) and by the population (more than a decade later). The latter kept using paper-based solutions, such as checks and boletos, which should be supported by the former until today. Finally, we discovered that the current historical moment has been implying in project decisions distinct than the ones taken by the traditional banking apps. For instance, `Nubank` (the major Brazilian fin-tech), which was founded more recently, is operates completely digital and paperless. Due to that, it has neither OCR nor native libraries, reducing the attack surface and dependency on third party's code. However, `Nubank` relies on native Java-based libraries for tasks such as reading the screen to collect customer's signatures in an already digitalized version.

Banking Ecosystem: Heterogeneity increases complexity. The Brazilian ecosystem for mobile banking apps is very diverse, as

[1] https://github.com/linkedin/qark [2] https://github.com/plum-umd/dexdump
[3] https://developer.android.com/studio/command-line/aapt2
[4] https://ibotpeaches.github.io/Apktool/ [5] https://github.com/skylot/jadx
[6] https://mitmproxy.org/ [7] https://www.wireshark.org/

most banks provide specific applications for accessing several services (e.g., trading stocks, enterprise level customization). Itau, for instance, has more than 30 distinct apps (Figure 2).

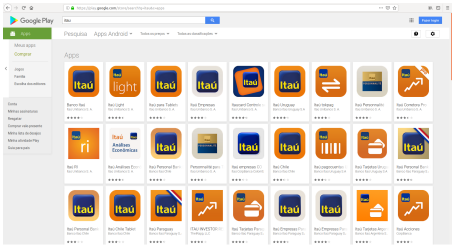

Figure 2: App Heterogeneity. Banks deploy multiple apps for accessing their distinct services.

While multiple apps from each bank share some code portions, mainly in the components related to layout drawing and authentication steps, the apps present significant differences in their set of features and how it is implemented. This fact is made clear when we look to the number of distinct architectures supported by the native libraries embedded in each application, as shown in Table 2. We notice that these complementary apps present a great number of supported architectures, sometimes in a number even greater than in the main apps, thus increasing app's complexity. In fact, even for the main apps, it is not clear app's needs for supporting so many architectures, as customer's devices are concentrated in the standard ARM platform. For the Itau's app, we found native library support for the MIPS architecture, which increases the attack surface while benefiting only a very small amount of devices.

Table 2 also shows differences between the apps deployed in distinct countries by global banks, such as the case of Santander Brazil (BR) and Santander Germany (DE). Whereas the apps share some code portions, their core is tied to local factors to support each country customer's demands, thus being developed by local teams. We identified that each app version is signed with certificates owned by their respective local bank authorities. Whereas allowing for more customer-focused products, the multitude of apps deployed in a non-uniform way makes management harder, as failures identified in a given app version is not guaranteed to be fixed in another one, maintained by a distinct team.

Network Security: History teaches, sometimes we learn. As history shapes overall project decisions, such as including libraries for documents scanning, previous breaches and vulnerabilities should also affect apps development regarding the implementation of protection mechanisms. While some developers have learned how to protect their apps, it is not true for all cases. A major attack vector against banking apps is to tamper with their communication channels so an attacker can spy on and interfere with the transmitted data. A popular implementation of this kind of attack in the past was to force apps to use HTTP connections instead of HTTPS (downgrade attack). If developers learned from this past lesson, current apps should be able to prevent the use of insecure channel such as HTTP for data transmission. To evaluate that in practice, we limited HTTPS connections in our gateway and attempted to login in all bank apps. The obtained results are in Table 3. None of the evaluated apps accepted transferring data using the HTTP, thus showing that developers learned from historical attacks and avoided repeating this vulnerability in new deployments. More than refusing connecting via HTTP, all apps warned the user about their usage in an unsafe environment, as shown in Figure 3.

Since the apps block HTTP connection attempts, the second most popular attack strategy for eavesdropping communication channels is to deploy a rogue secure server and interpose it to apps connections, which is known as a Man-In-The-Middle (MITM) attack. Secure apps are supposed to recognize and communicate only with legitimate servers, identified using certificates, but previous research work have already demonstrated that many developers fail to implement this mechanism [8]. To evaluate that in practice, we deployed a MITM attack via our Internet gateway and installed our rogue certificate in the smartphone prior to attempting to login into the bank accounts using all applications. Table 3 shows that only three apps (BB, Itau, and Santander) are completely secure against MITM attacks, as they implement a technique called certificate-pinning, thus refusing to accept our rogue certificate. More than rejecting the certificate, these apps completely refused to run when connected to the unsafe network, showing that their developers learned from previous attacks. However, the other apps failed to pin their certificates and accepted our rogue certificate, being vulnerable to eavesdropping. Some apps (e.g., Caixa and Bradesco), despite failing to pin their certificates, protect their information with an additional, in-app cryptography layer—they do not not reveal the password, but still leak query data, as shown in Code 1. The Nubank app, in turn, do not implement any additional, in-app cryptography layer: it expects its connection to be protected by a legitimate certificate. However, once a rogue certificate was provided and no certificate pining routine was effectively implemented, it leaked the password in plain text, as shown in Code 2.

```
agn: 1523
ctaDig: 68925
tit: 1
senha: 0000
senhaCrypt: AKyxSjs2L...
```

Code 1: Bradesco cryptopass

```
data{
device_id: 7391db29fe...
login: 95430119822
model: Asus ASUS_Z00VD
password: 12345678
```

Code 2: Nubank plain

Data Storage: Learning that data security goes beyond secure transmissions. Another popular way of extracting sensitive data is to harvest the mobile device for application traces that may reveal some user information. To avoid this possibility, sensitive information should be stored either remotely or encrypted. Some developers, however, still fail to implement secure storage for sensitive data. For instance, the BB app displays notification messages to its users, such as credit cards purchases (Figure 4). Looking into BB's /app/data/ directory, we found the bb_notifications_db, a SQLite database, which is accessible to any user and stores the notification messages in clear (Code 3).

```
sqlite> select * FROM notifications;
43|1|602710506|Compra no valor de R$  88,76, realizada
      em Uber Do Brasil Te 0x00E0s 18:54 do dia 25/10,
      com cart0x00E3o final 1169. Caso n0x00E3o
     reconhe0x00E7a, clique em Bloquear Cart0x00E3o
     .|...|1|13|0|1|1|0|Compra com cart0x00E3o||1|1
```

Code 3: BB's app stores notifications in a plain SQLite DB.

Table 2: Apps Diversity. Same bank, distinct apps and architecture support, increased attack surface/maintenance needs.

Bank	Santander		BB		Bradesco				Caixa			Itaú			
Version	BR	DE	Main	Investing	Main	Cards	Corporate	Trading	Main	Tablet	Cards	Main	Personal	Enterprise	Tablet
Architecture	7	6	2	5	1	6	7	0	3	4	5	7	7	7	3

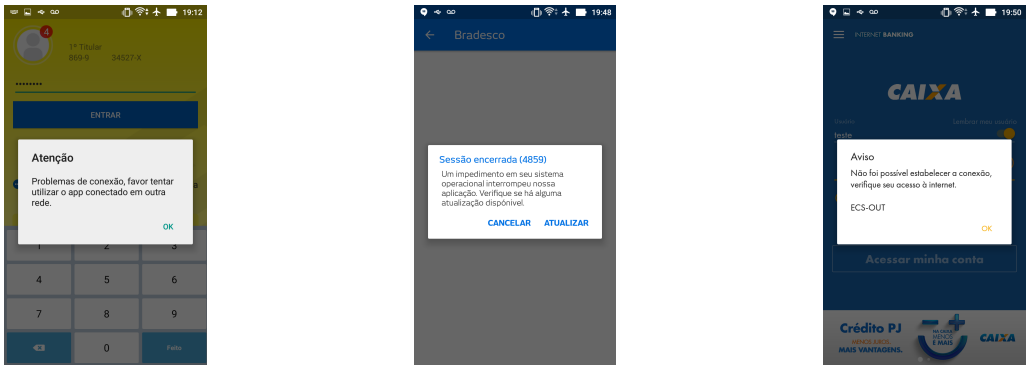

Figure 3: BB, Bradesco, and Caixa banking applications (left to right) blocking HTTP connection attempt.

Table 3: Network Attacks Prevention. All apps are protected against HTTP downgrade; many are vulnerable to MITM.

APP	HTTP Downgrade	Certificate Pinning	Plain Password
BB	✓	✓	–
Bradesco	✓	✗	✓
Caixa	✓	✗	✓
Itau	✓	✓	–
Santander	✓	✓	–
Nubank	✓	✗	✗

Although exhibiting credit card purchases is not a security vulnerability, it is a significant privacy issue: anyone with access to the device may use it to profile the customers' expenses, even with the app logged out. This happens because the app allows backup permission in its manifest, enabling data copying even in non-rooted devices. It seems to be a systemic development mistake, as all evaluated Brazilian bank apps asks for backup permission, thus allowing us to identify other types of locally stored data (e.g., `Itau`'s app developers forgot to remove development configuration files and debugging information, making reverse engineering easy).

Tampering Protection: Trojanized Apps and a lesson still not learned. The history of cyber-frauds in Brazil shows that trojanized and phishing applications have been popular strategies leveraged by attackers since the times of desktop-based banking clients (see Section 2). Therefore, banks learning from this past should care about trojanized apps when migrating their internet banking services to the mobile environment, as attackers often also migrate their attacks and strategies to newer environments.

Unlike desktop binaries and web backends, which naturally protect application source code, mobile apps in the Android platform are distributed as bytecode and thus can be decompiled. Thus, whereas attackers can implement a cloned application from scratch, the most straightforward way for trojanizing an application is repacking the original APK with the addition of a malicious payload. Therefore, banks applications should implement anti-tampering routines to avoid being trojanized.

To prevent reverse engineer, bank apps should, for instance, avoid running in rooted devices and emulators, since these environments provide significantly greater inspection capabilities than stock Android installations. Table 4 shows that, although all apps include some reference in the source code to anti-tampering procedures, such as fingerprints, in practice, these strategies are not enough to prevent their execution in non-standard environments.

Table 4: App Self Protection. Applications fail to prevent their execution in non-standard environments.

App	Source Code	Emulator	Root
BB	✓	✗	✗
Bradesco	✓	✗	✗
Caixa	✓	✗	✗
Itau	✓	✗	✗
Santander	✓	✗	✗
Nubank	✓	✗	✗

The reason for all bank's apps failing to verify rooting and emulation is that they make use of the same routines from the fabric framework[8] for implementing these checks, as shown in Code Snippets 4, 5, and 6. Therefore, if the rooting scripts or the emulator implementation hide these files from the app, they will all fail in the same manner.

```
((!isEmulator && buildTags != null && buildTags.
    contains("test-keys")) || new File("/system/app/
    Superuser.apk").exists()) {
```

Code 4: App Self Protection: Nubank, BB, and Bradesco.

```
Log.w("GoogleSignatureVerifier", "Test-keys␣aren't␣
    accepted␣on␣this␣build.");
```

Code 5: App Self Protection: Nubank, Santander, BB, and Bradesco shared code excerpts.

[8] https://get.fabric.io/

```
1 com.itau/sources/io/fabric/sdk/android/b/a/0x02BD.java
      :        if (!0x02BB && str != null && str.
      contains("test-keys")) {
2 com.santander.app/sources/defpackage/cx.java:
      r4 = "test-keys";
3 br.com.bb.android/sources/br/com/bb/android/api/utils/
      BBRootUtil.java:        return buildTags != null
      && buildTags.contains("test-keys");
4 com.bradesco/sources/br/com/bradesco/integrador/utils/
      RootUtil.java:        return buildTags != null &&
       buildTags.contains("test-keys");
```

Code 6: App Self Protection: Itau, Santander, BB, and Bradesco shared code excerpts.

In addition to run-time checks, repackaging prevention can also be implemented via fingerprinting, signatures and integrity checking at source code level, but none of these characteristics were found in the Brazilian banking apps. Therefore, we were able to develop trojanized version of all apps by patching their `smali` bytecode and repackaging them as a new APK, as exemplified in Figure 5 and 6. The possibility of trojanizing bank apps combined with the diversity of bank apps available in the app stores make harder for users to differentiate between legitimate and malicious apps, thus becoming more vulnerable to rogue app attacks.

Authentication & Usability: Learning from other countries to anticipate new attacks. The findings on the current (in)security of mobile banking apps allow us to envision new attack vectors against Brazilian banks. For instance, European malware authors have been increasingly relying on overlay-based attacks [4, 53] to trick users into either giving up their credentials, or spy on them through accessibility services [24, 36], thus banking apps should adapt security mechanisms against these novel attack vectors. While we are unaware of any real attack exploiting the following presented weaknesses in the Brazilian scenario, we check whether Brazilian banks are also learning from other countries' experiences to try to predict new attack vectors and anticipate incident response. We focused our evaluation on stealth variants of accessibility-based PII leakages, namely accessibility event-based, screenrecording-based, and malicious input method editor (IME) based attacks [34]. Concerning overlay-based attacks, we tested the possibility of overlays over the login fields of the app [9, 20].

Vetting our test set of applications against susceptibility to these attacks painted a grave current condition. We summarize our results in Table 5. None of the tested banks applied any security mechanisms against overlay or accessibility-based attacks. All of them leaked the username and password through a11y events as soon as the user started typing the credentials. Screenrecordings and screenshots were possible at any time, including during the login procedure. As a11y services can install malicious IMEs, e.g. virtual keyboards, we tested if the apps either implemented their own IME for credential input or at least could detect the presence of 3rd party IMEs on the system and warn the user, which was not the case. Lastly, we tested overlays over the login fields of the banking apps and whether the apps tried to detect or even prevent these overlays, however, none of the tested applications applied any safeguards against overlays.

From the past to the future of mobile banking: Whatsapp-based transactions & Vulnerabilities. In our historical approach of banking security, we concluded our experiments trying to understand how the current scenario can be shaping future threats as the past historical facts shaped our present. People in Brazil currently have more smartphones than desktops [50], and most of them only have access to social networks, as their use is not charged in most basic mobile data plans [5]. Among all social networks, Whatsapp is the most popular in Brazil [47]. Therefore, banks adopting Whatsapp support to achieve a broad market is seen as a natural move by most market analysts. In fact, `BB` [52] and `Bradesco` [16] recently deployed support for Whatsapp-powered transaction using chatbots, as shown in Figure 7 and Figure 8, respectively.

Table 5: Apps susceptibility to UI and accessibility-based attacks. Our evaluation shows that no security mechanisms have been implemented to prevent these attacks.

Banking App	A11y Event Sniffing	Screenrecording	Malicious IME	Login Overlay
BB	✗	✗	✗	✗
Bradesco	✗	✗	✗	✗
Caixa	✗	✗	✗	✗
Itau	✗	✗	✗	✗
Santander	✗	✗	✗	✗
Nubank	✗	✗	✗	✗

Supporting transaction via Whatsapp extends bank's market to cover customers with limited internet data plans whereas it poses new security risks, since security is outsourced to the third-party app's developer, without the bank's control. Moreover, attacks to the Whatsapp platform were already seem in the past [38] and nothing prevents them to be ported to newer versions, which also affects banking services security. In face of that, we inspected Whatsapp security features to understand the risk posed by the Whatsapp-based banking mechanism. We discovered that Whatsapp enables accessibility services to interact with the app in a general way, such as taking screenshots and pushing buttons. While it is not a security issue for the Whatsapp operation model, it certainly is for online banking. For instance, screenshots of Whatsapp chats may disclose user's bank account balance, a significant privacy issue.

To demonstrate the feasibility of abusing accessibility services to interfere with Whatsapp-based banking services, we implemented a malicious accessibility service that sends messages to the bank chatbot on behalf of the actual user (victim). In such banks chatbots, once the victim user has already unlocked its phone in bank's systems to perform transactions and first supplied its credentials, no additional authorization is required to perform transactions in future conversations. Therefore, we were able to transfer money from one account (victim's one) to another (attacker's one) just by injecting the destination account number as Whatsapp messages to the chatbot. Even worse, accessibility services can run in background, even when the smartphone screen is locked, thus performing the attack without users noticing.

5 GUIDELINES FOR INTERNET BANKING APPS DEVELOPMENT

In this section, we revisit our findings to pinpoint what banks should follow to assure the safe development of their apps.

Methodologically Assessing Apps Security. As we conducted a structured vulnerability assessment, we were able to identify flaws in apps' implementation and architecture. Therefore, we advocate

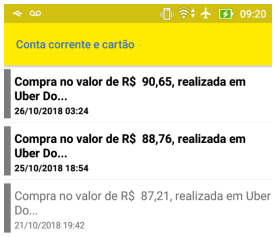

Figure 4: BB's app notification messages displaying credit card purchases.

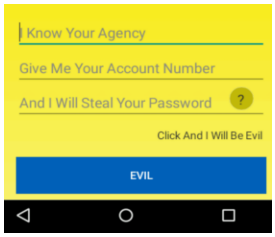

Figure 5: App Repacking. Malicious BB app.

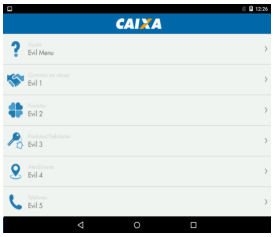

Figure 6: App Repacking. Malicious Caixa app.

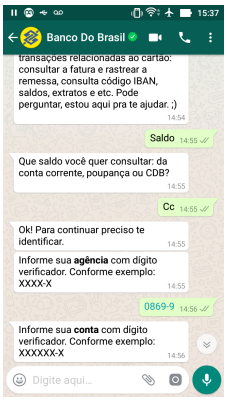

Figure 7: BB's Whatsapp chatbot. Banks are supporting transaction in third-party apps, which increases both usability and risks.

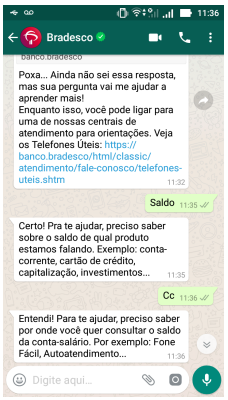

Figure 8: Bradesco's Whatsapp chatbot. Banks are supporting transaction in third-party apps, which increases both usability and risks.

that banks need to perform structured assessments of their apps to increase security and resistance to attacks. In particular, we recommend the OWASP MASVS framework [44], as we did.

Reduce Heterogeneity. The diversity of available apps (for distinct supported architectures) is a feature of the Brazilian mobile banking ecosystem, we believe that reducing the number of apps is a necessary step to decrease the chance of users being deceived into installing trojanized applications instead of the actual bank app. Less embedded libraries in the apps would also reduce the exposed attack surface. Regardless the reason behind the deployment of multiple apps from the same bank, instead of a single one, it increases attack surfaces due to more code pieces to be programmed, verified and released. For instance, since app's functionalities are not unified in a single repository, updates might be independently deployed for each of them, increasing the risk that an application is vulnerable to an attack while another app's branch is already fixed. Therefore, we advocate for unified solutions and joint efforts as the best long term security practice for these apps.

Pin certificates. Eavesdropping into app's connections is a widespread attackers' strategy. While developers learned lessons such as avoiding HTTP downgrades, most apps are still vulnerable to MITM attacks due to not pinning their certificates. Since it is a well-known OWASP top10 vulnerability, we emphasize the importance of developers' continuous training, especially for proper implementation of cryptography routines and protocols.

Do not store sensitive data locally. We discovered that the BB's app stores files locally in plain text (privacy violation). We suggest that apps only store sensitive data in the bank's remote server. In addition, all apps should avoid the backup permission in their manifest, thus preventing a malicious actor to collect app's data.

Do not outsource security to third parties. Supporting transactions within third-party apps is a fact that will define future design decisions, such as the recent BB's Whatsapp-based chatbot. In the past, the same bank integrated Facebook support in its app [21], but this strategy was left behind due to the risks of incorporating third-party code to an online banking app (OWASP `M10. Extraneous Functionality` [44]). However, the bank-Whatsapp integration suggests other moves like that soon. Though we are not aware of an in-the-wild similar attack, our PoC showed the feasibility of a money transfer with a malicious accessibility service targeting the bank-Whatsapp integration. Blocking such attack is out of the bank control, since Whatsapp was misused instead of the bank's app.

Learn from the past. In this work, we revisited the history of the development of Electronic Banking in Brazil, from its early computerization in 80's to the current mobile apps, and the associated shaping of technological development. We observed that the past need of document scanning end up with current apps deploying native libraries for OCR. Paperless fin-techs, in turn, do not require native libraries in their apps. These project decisions are only explained if we take the historical context into account. Thus, we expect that our work incentives other researchers to consider other contexts besides the technical aspects of a security evaluation.

Learn from other countries. Our evaluation showed that Brazilian apps are completely unaware of UI and a11y-based attacks, and information leakage. While we are unaware of such kind of attack being actually exploited in Brazil, we recommend that banks make efforts to learn from other contexts so as to anticipate incident response. Therefore, we suggest the immediate implementation of countermeasures against overlays, as described by Aljarrah et al. [2], and the countermeasures against a11y-based attacks and screen recordings described by Kalysch et al. [34].

Improving Regulation. Banks operation are usually very regulated. Hence, a way to enhance bank's apps security is by adopting stronger, mandatory rules. In this sense, the European Commission has made a huge legislative change in its Revised Payment Service Directive 2015/2366 (PSD2) from November 16th 2015 [14] how

payment services and payment service providers throughout the European Union (EU) and European Economic Area (EEA) should conduct their transactions. This directive includes Regulatory Technical Standards (RTS) [15] which define what security mechanisms must be employed in online banking systems and transactions. The RTS came into effect in March 2018 and are applicable since September of the same year. Among others, the RTS dictate the use of stronger protection mechanisms and stronger authentication mechanisms for online payments which now have to make use of multiple factor authentication mechanisms (MFA), and a single-use token must be dynamically linked to the transaction's beneficiary and amount, with both being displayed to the customer for verification [28, 31]. The rules aim to better protect consumers while paying online, promote the use of novel online and mobile payments, and make cross-border European payment services safer [14]. They also provide a legislative framework which forces banking apps to adapt a certain level of safeguards due to the outlined security goals of secure transactions and strong customer authentication. European researchers have also been looking into their banking systems and the security of mobile banking applications. European banking applications seem to adopt a by far stronger protection strategy, as evident by the measures the authors had to adopt to attempt an analysis [30, 31]. Among others, the use of commercial protectors is employed to impede analysis by reverse engineers and provide a strong measure of integrity protection, as well as enhanced safeguards for data stored on the mobile device and transmitted through the network [30]. Therefore, we advocate for other countries, including Brazil, to adopt similar rules towards establishing a more security environment for mobile banks apps operations.

6 DISCUSSION

In this section, we discuss the limits and implications of our work.

Country-Specific studies are important to shed light on trends of in localized contexts that could be used to predict their evolution in regard of the global scenario. For instance, previous studies have shown that mobile platforms were responsible for integrating banks to the local population culture in India [22] and Uganda [23]. In this same sense, we evaluated the Brazilian banking ecosystem to highlight local trends already present in Brazil that might scale to the world scenario, such as Whatsapp-based transactions.

Fin-Techs are the future of electronic banking, thus causing attacks models to change, as already seen (e.g., in Germany [29]). In this sense, we leveraged the Brazilian scenario to demonstrate how fin-techs strategies, such as complete paper elimination, shaped the development of the new banking apps by causing the elimination of third-party libraries. We expect our work could motivate other researchers to investigate the impact of fin-techs in their countries and also their effects to the security community in a global way.

Limitations. This work presented current vulnerabilities and potential future threats based on current banking apps implementations. Whereas this effort is essential to understand development and programming mistakes and increase banking applications security, this is a non-exhaustive approach, with still multiple open research opportunities. For instance, as we do not have an open account in each bank and evaluated only their login prompts, other execution paths need to be evaluated to identify additional vulnerabilities possibly present in other contexts, such as the web APIs online exposed for mobile apps communication.

This work also evaluated Nubank as the single example of a fin-tech mobile app. As for vulnerabilities, our work cannot be considered an exhaustive analysis of fin-tech apps. Instead, our goal was to present insights on the impact of traditional banks migrating their operations to a fin-tech model.

Future Work. We aim to extend our research effort for other scenarios in which cultural, social and historical aspects may have influenced security in the form of taken project decisions. As an example, we aim to investigate the implementation of e-government apps, given their need of storing citizen's sensitive data.

7 RELATED WORK

The adoption of Internet Banking solutions motivates a myriad of studies, including security evaluations and considerations. In this section, we present the closest related work to better position ours.

Internet banking frauds have been studied since the emergence of online banking services. These studies are mainly distributed in two categories: (i) enhancing banking services protections (e.g., captcha resistance investigation [39], adding untrusted environments execution capabilities [46]); and (ii) investigating the attacks ecosystem (e.g., financial losses caused by malware [3]). To bridge this gap, we presented here a joint analysis of technical aspects of the Brazilian banking services and their underlying cultural/social support, which also determines their impact. In this sense, the work of Gregio et al. [25] is the closest to ours—they presented desktop malware targeting Brazilian banks, and our analyses extended their evaluation to also cover mobile platforms.

Mobile banking frauds were the natural steps for criminals to match the evolution of Internet Banking services. Many mobile banking analyses focused on either presenting a landscape of existing services [1], or digging into implementation details [7]. We presented a combined approach that associates social and technical aspects of Brazilian mobile banking apps. Therefore, we did not limit ourselves to observe technical aspects, such as a previous study on the use of certificates by Brazilian apps [8].

The Brazilian scenario of fully and widespread banking automation offers attackers several opportunities to leverage varied strategies against bank customers. Besides security companies report the occurrence of multiple fraud schemes (e.g., banking malware infections [49] and credit card skimming [35]), we identified the lack of a systematized work on this type of threat. In this work, we offered a scientific view of the Brazilian bank ecosystem's historical security breaches and current implementation flaws, which allowed us to also pinpoint and discuss potential future threats.

8 CONCLUSION

In this work, we reviewed the development of the Internet Banking in Brazil, a country adopting this type of service in a massive and early adopted manner, thus being targeted by specific attacks, such as boleto malware. We showed how Internet Banking implementations are affected by historical (e.g., Brazilian banks computerizing their systems in the 80's to keep up with a period of high inflation)

and social aspects (e.g., attackers targeting printed boletos to reach users without Internet access and bank accounts) and how attackers adapt their approaches to react to bank's new defensive technologies (e.g., attackers moving from fake desktop apps to fake mobile apps). We also showed that despite the early adoption of Internet Banking in Brazil, developers still fail to implement mobile bank apps in a safe manner, thus leaking user's sensitive data by storing data in plaintext and allowing traffic to be hijacked by attackers via man-in-the-middle attacks. Finally, we presented a new attack to a Whatsapp-based chatbot mechanism deployed by some Brazilian banks by injecting Whatsapp messages on behalf of the victim user via a malicious accessibility service, thus showing the risk of banks outsourcing protection to third-party apps.

Acknowledgments. This project was partially financed by the DAAD-ISAP program, the Brazilian National Counsel of Technological and Scientific Development (CNPq, PhD Scholarship, process 164745/2017-3) and the Coordination for the Improvement of Higher Education Personnel (CAPES, Project FORTE, Forensics Sciences Program 24/2014, process 23038.007604/2014-69).

REFERENCES

[1] Mousa Albashrawi and Luvai Motiwalla. 2017. Understanding Mobile Banking Usage: An Integrative Perspective. In *SIGMIS-CPR*. ACM.
[2] Abeer AlJarrah and Mohamed Shehab. 2016. Maintaining user interface integrity on Android. In *IEEE 40th Annual COMPSAC*, Vol. 1. 449–458.
[3] Amin. 2016. A Survey of Financial Losses Due to Malware. In *ICTCS*. ACM.
[4] Yair Amit. 2016. Accessibility Clickjacking – Android Malware Evolution. (2016). https://tinyurl.com/y3vq8fh5, access: 11/Aug./2018.
[5] bnamericas. 2015. Claro Brasil launches plan with 'free' WhatsApp, Facebook and Twitter. https://tinyurl.com/yyl26rtj. (2015).
[6] Pew Research Center. 2013. 51% of U.S. Adults Bank Online. https://tinyurl.com/y6ja3sgn. (2013).
[7] Sen Chen, Ting Su, Lingling Fan, Guozhu Meng, Minhui Xue, Yang Liu, and Lihua Xu. Are Mobile Banking Apps Secure? What Can Be Improved?. In *ACM Joint Meeting on ESE Conf. Symp. Foundations of SE*. 797–802.
[8] Rafael Junio da Cruz e Diego Aranha. 2016. Analise de seguranca em aplicativos bancarios na plataforma Android. In *SBSeg, SBC*.
[9] Bove Davide and Kalysch Anatoli. 2019. *itit.* Chapter In pursuit of a secure UI: The cycle of breaking and fixing Android's UI. https://tinyurl.com/y45wn5s8
[10] Folha de São Paulo. 2018. Quatro maiores bancos concentram 78,5% do crédito, diz BC. https://tinyurl.com/yxz4o6un. (2018).
[11] Banco do Brasil. 2013. Internet Banking - Módulo de Segurança. https://tinyurl.com/y3s5upth. (2013).
[12] Gazeta do Povo. 2013. Pesquisa mostra que 77% dos brasileiros já usam cartão de crédito. https://tinyurl.com/y3fdgobn. (2013).
[13] Stephen Doherty and Nikolaos Tsapakis. 2015. Analysis of malware targeting the Boleto payment system. (2015). ://tinyurl.com/yy2jtedr.
[14] European Commission. 2015. Payment services (PSD2) – Directive (EU) 2015/2366. (2015). https://tinyurl.com/y2wnagmx.
[15] European Commission. 2018. COMMISSION DELEGATED REGULATION (EU) 2018/389. (2018). https://tinyurl.com/y2tlh6cz.
[16] Exame. 2018. Bradesco permite consulta de saldo via WhatsApp. https://tinyurl.com/yxbweypm. (2018).
[17] FDIC. 2016. A Bank Costumer's Guide to CyberSecurity. https://tinyurl.com/y2ozvuma. (2016).
[18] Forbes. 2018. Brazilian Fintech Nubank Launches Debit Card To Reach 120M Clients. https://tinyurl.com/y9vc3ndw. (2018).
[19] Forbes. 2018. Nubank: Driving Financial Services Competition In Brazil. https://tinyurl.com/y5k5c5py. (2018).
[20] Yanick Fratantonio, Chenxiong Qian, Simon P Chung, and Wenke Lee. 2017. Cloak and Dagger: from two permissions to complete control of the UI feedback loop. In *S&P*. IEEE.
[21] G1. 2014. Banco do Brasil desliga integração com Facebook após reclamação. https://tinyurl.com/y46ovj3z. (2014).
[22] Amitava Ghosh, Sourya Joyee De, and Ambuj Mahanti. 2014. A Mobile Banking Model in the Cloud for Financial Inclusion in India. In *ACM Int. Conf. Design of Comm.*
[23] Ishita Ghosh. 2012. The Mobile Phone As a Link to Formal Financial Services: Findings from Uganda. In *Int. Conf. Inf. and Comm. Tech. and Dev.* ACM.
[24] Dan Goodin. 2018. New Android Malware with never before seen spying capabilities. (2018). https://tinyurl.com/y46hezkq, accessed on 17. August 2018.
[25] André Ricardo A. Grégio, Dario Simões Fernandes, Vitor Monte Afonso, Paulo Lício de Geus, Victor Furuse Martins, and Mario Jino. 2013. An Empirical Analysis of Malicious Internet Banking Software Behavior. In *SAC*. ACM.
[26] Guardian. 2015. So you think you're safe doing internet banking? https://tinyurl.com/yxmskmlo. (2015).
[27] Guardian. 2018. UK bank customers lost £500m to scams in first half of 2018. https://tinyurl.com/yyg4vlfh. (2018).
[28] Vincent Haupert and Stephan Gabert. 2019. Short Paper: How to Attack PSD2 Internet Banking. In *23rd Intl. Conf. on Financial Cryptography and Data Security.*
[29] Vincent Haupert, Dominik Maier, and Tilo Müller. 2017. Paying the Price for Disruption: How a FinTech Allowed Account Takeover. In *ROOTS*. ACM.
[30] Vincent Haupert, Dominik Maier, Nicolas Schneider, Julian Kirsch, and Tilo Müller. 2018. Honey, I Shrunk Your App Security: The State of Android App Hardening. In *DIMVA*.
[31] Vincent Haupert and Tilo Müller. 2018. On App-based Matrix Code Authentication in Online Banking.. In *ICISSP*. 149–160.
[32] IDGNow. 2018. Mais de 58 milhões brasileiros acessam Internet apenas pelo celular. https://tinyurl.com/y69qgo3l. (2018).
[33] Yeongjin Jang, Chengyu Song, Simon P Chung, Tielei Wang, and Wenke Lee. 2014. A11y attacks: Exploiting accessibility in operating systems. In *ACM CCS*.
[34] Anatoli Kalysch, Davide Bove, and Tilo Müller. 2018. How Android's UI Security is Undermined by Accessibility. In *ROOTS*. ACM.
[35] Kaspersky. 2018. Cloning chip-and-PIN cards: Brazilian job. https://tinyurl.com/yc6sw5v5. (2018).
[36] Swati Khandelwal. 2017. Ransomware Not Just Encrypts Your Android But Also Changes PIN Lock. (2017). https://tinyurl.com/ydxm3nsm, access: 20/Aug./2018.
[37] Joshua Kraunelis, Yinjie Chen, Zhen Ling, Xinwen Fu, and Wei Zhao. 2013. On malware leveraging the Android accessibility framework. In *International Conference on Mobile and Ubiquitous Systems: Computing, Networking, and Services.*
[38] Andreas Kurtz. 2011. Shooting the Messenger. https://tinyurl.com/yxt79zgq. (2011).
[39] Shujun Li, S. Amier Haider Shah, M. Asad Usman Khan, Syed Ali Khayam, Ahmad-Reza Sadeghi, and Roland Schmitz. 2010. Breaking e-Banking CAPTCHAs. In *Proceedings of the 26th Annual Computer Security Applications Conference*. ACM.
[40] MiamiHerald. 2016. Brazil's hackers win the gold in credit card crime. https://www.miamiherald.com/news/nation-world/world/article93939247.html. (2016).
[41] MundoMarketing. 2016. 76% dos brasileiros usam internet banking, aponta pesquisa do Facebook. https://tinyurl.com/y6ks9z5j. (2016).
[42] Dayton Daily News. 2018. 4 Brazilian men federally indicted for ATM skimmer, fake credit cards. https://tinyurl.com/y35gkup6. (2018).
[43] PR Newswire. 2018. Mobile Banking One of Top Three Most Used Apps by Americans, 2018 Citi Mobile Banking Study Reveals. https://tinyurl.com/y2guuapo. (2018).
[44] OWASP. 2016. Mobile Top 10 2016-Top 10. https://www.owasp.org/index.php/Mobile_Top_10_2016-Top_10. (2016).
[45] E. Pang. 2002. *The International Political Economy of Transformation in Argentina, Brazil and Chile Since 1960.* palgrave macmillan.
[46] Yanlin Peng, Wenji Chen, J. Morris Chang, and Yong Guan. 2010. Secure Online Banking on Untrusted Computers. In *ACM CCS*.
[47] Harvard Bussines Review. 2016. The Rise of WhatsApp in Brazil Is About More than Just Messaging. https://tinyurl.com/yy8762tk. (2016).
[48] WaterTown Savings. 2018. Security Q&A. https://tinyurl.com/yyunrum4. (2018).
[49] SecurityIntelligence. 2018. CamuBot: New Financial Malware Targets Brazilian Banking Customers. https://tinyurl.com/y8z2apln. (2018).
[50] TecMundo. 2018. Smartphone é mais popular do que notebook ou desktop no Brasil, diz estudo. https://tinyurl.com/y686bcvv. (2018).
[51] Visa. 2014. A História da Visa. https://tinyurl.com/y2hocrwn. (2014).
[52] ZDNet. 2018. Banco do Brasil launches financial transactions via WhatsApp. https://tinyurl.com/yaergk8s. (2018).
[53] Wu Zhou. 2016. https://tinyurl.com/j5qx9wo, access: 09/Jun./2018. (2016).