PixPirate: a new Brazilian Banking Trojan

By Francesco Iubatti, Alessandro Strino

Archived: 2026-04-02 10:56:19 UTC

Key points

- Between the end of 2022 and the beginning of 2023, a new Android banking trojan was discovered by the Cleafy TIR team. Since the lack of information and the absence of a proper nomenclature of this malware family, we decided to dub it **PixPirate**, to better track this family inside our internal Threat Intelligence taxonomy.
- PixPirate belongs to the newest generation of Android banking trojan, as it can perform **ATS (Automatic Transfer System)**, enabling attackers to automate the insertion of a malicious money transfer over the Instant Payment platform **Pix**, adopted by multiple Brazilian banks.
- PixPirate appears to have the following features, primarily achieved by abusing Accessibility Services, such as:
  - Ability to intercept valid banking credentials and perform ATS attacks on multiple Brazilian banks via Pix payments
  - Ability to intercept/delete SMS messages
  - Preventing uninstall
  - Malvertising

Introduction

In the first half of 2020, Latin America recorded the **world's highest cyber-attack rate** with 3x more mobile browser attacks than the global average[1]. As per multiple TA observations, phishing attacks have a high success rate and are utilized by financially motivated hackers to steal sensitive info like bank logins. In 2022, several mobile-based banking trojans reappeared after dormant periods, and new ones emerged disguised as legitimate mobile apps (e.g. Vultur, SOVA, TeaBot). Cyber attacks continued to surge in the latter half of 2021 and 2022, not only in quantity but also in impact. Financial groups are expanding their reach, targeting organizations worldwide mainly through **ransomware** but also increasing the range of their activities.

That said, on top of this evolution, one of the most crucial elements which have been disrupting the current state-of-art of anti-fraud departments is Instant Payments. **Instant Payments** areelectronic money transfers that make transferred funds available in real-time from one account to another, bringing speed to money transfers and increasing the underlined risk of unhandled frauds and, consequently, monetary losses. In recent years, the adoption of Instant Payments has been rapidly growing in Europe, America, and, more recently, also in Brazil, with the introduction of Pix, an instant payment platform created and managed by the monetary authority of Brazil, the Central Bank of Brazil (BCB), which enables the quick execution of payments and transfers and now counting over 100 million registered accounts.

One such threat recently discovered in the wild is a **brand-new mobile malware** targeting LATAM countries, specifically Brazil. The primary goal of this malware is to steal sensitive information and perform fraud on users that regularly use Pix platform. This report will provide a detailed analysis of this malware in the following chapters.

[1] https://www.interpol.int/en/News-and-Events/News/2022/INTERPOL-Working-Group-highlights-cyber-threats-across-the-Americas

Technical Analysis: Overview

PixPirate hides its malicious purposes with familiar names and icons, posing as a legitimate application to the victims. At the end of 2022, we intercepted the following decoys, which appear to be pretty consolidated by TAs for delivering their malicious samples:

Figure 1 - Main names/icons used by PixPirate

PixPirate is usually delivered using a dropper application, used to download (or in some cases just to unpack) and install the banking trojan. During its installation, PixPirate immediately tries to enable Accessibility Services that keep being requested persistently with fake pop-ups until the victim accepts.

```
<uses-permission android:name="android.permission.QUERY_ALL_PACKAGES"/>
<uses-permission android:name="android.permission.REQUEST_INSTALL_PACKAGES"/>
<uses-permission android:name="android.permission.INTERNET"/>
<application android:label="Bonus Authenticator" android:icon="res/NR.
```

Figure 2 - Permissions used by the dropper application

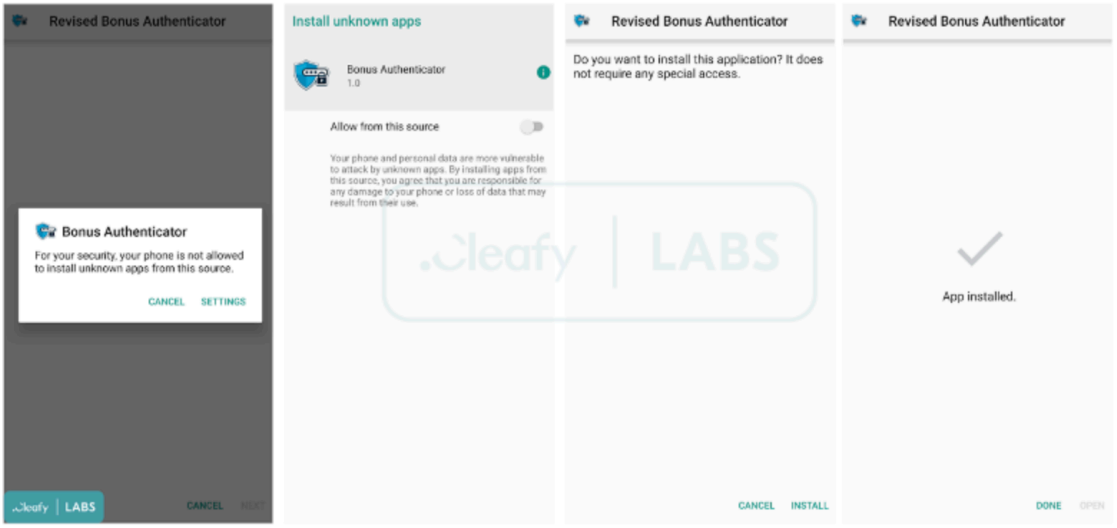

Figure 3 - Installation phases of PixPirate

Abusing the Accessibility Services is a standard routine for banking trojans since they provide features to interact with other apps. After the victim gives the permissions, PixPirate will enable all its malicious functionalities.

After inspecting PixPirate code, we identified a few references related to a framework called **Auto.js**[2]. This is an open-source tool for automating tasks on Android devices using JavaScript. It allows developers to write scripts that can interact with the device's UI and perform actions like finding and interacting with UI elements, entering text, scrolling through lists, simulating touch events, etc. Auto.js also provides a built-in JavaScript interpreter, which allows scripts to run on the device itself without the need for an external runtime. The following list is an example of the built-in functions available on Auto.js:

- **sleep()**: pause the execution
- **toast():** show a toast notification
- **swipe():** execute a swipe gesture
- **click():** clicks on the screen
- **device.width:** get the device screen width
- **device. height:** get the device screen height

Figure 4 - Auto.js home page

Since Auto.js represents a new framework for mobile banking trojan, we wanted to understand the reason behind this choice. By inspecting the framework capabilities, it was possible to identify some features that could speed up the development phase:

- **Non-Android coders** can write automation scripts using JavaScript code and create standalone applications.
- **Web communication management** within the application.
- Built-in **mechanism to encrypt/obfuscate the code**.

TAs were able to adapt this legitimate framework and building-up the entire malicious routines in JavaScript, executed on top of the Auto.js stack. For slowing down analysts, TAs adopt a heavy layer of code obfuscation, including multiple techniques such as string array encoding, control flow flattering, etc.

The following features have been observed:

- Preventing uninstall
- Disabling Google Play Protect
- Intercepting SMS messages
- Intercepting banking credentials
- Monitoring victim’s financial activities
- Malvertising via push notifications
- Perform ATS attacks via PIX payments

Figure 5 - Overview of the malicious JavaScript routines

The following chapters explore and discuss all PixPirate's main features.

[2] https://pro.autojs.org/

Password Stealer

One of the JavaScript modules of PixPirate is used to steal the banking password with the help of the well-known accessibility services of Android. Inside this module, a specific function was created for each targeted bank, since every banking application has a different layout.

In fact, through the Accessibility Services, PixPirate can recognize the different UI elements of the bank's activity [3] and the password element displayed on the screen. If it detects some changes in the password input text, it grabs the password of the user (if it hasn't already been stolen previously).

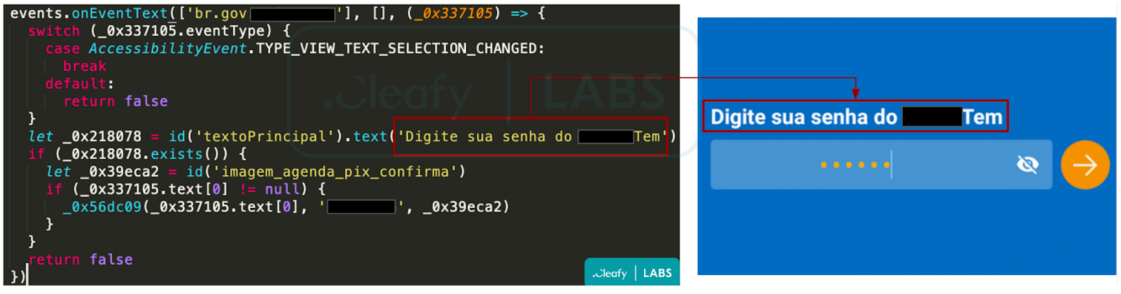

Figure 6 - Example of a portion of code used to steal the password of the targeted bank

[3] An Activity in Android represents a single screen with a user interface and is used to display and interact with content.

ATS landed on Pix System

The Banco Central do Brasil (BCB) created Pix, an instant payment method that enables users like people, companies, and governmental entities, to send or receive payment transfers in a few seconds at any time, including non-business days. Furthermore, Pix transactions can be performed between any payment institutions or financial institutions that comply with this ecosystem.

In recent years, different Android banking trojans have been created to perform fraudulent transactions on this ecosystem, like PixStealer or the most recent BrasDex.

PixPirate's TAs created a specific javascript file for each targeted bank to manage the different phases of the fraudulent transaction, which can be summarized with the following steps:

- **The identification of the UI elements of the targeted app**, like buttons, texts, or inputs, to establish which activity is displayed on the victim device.

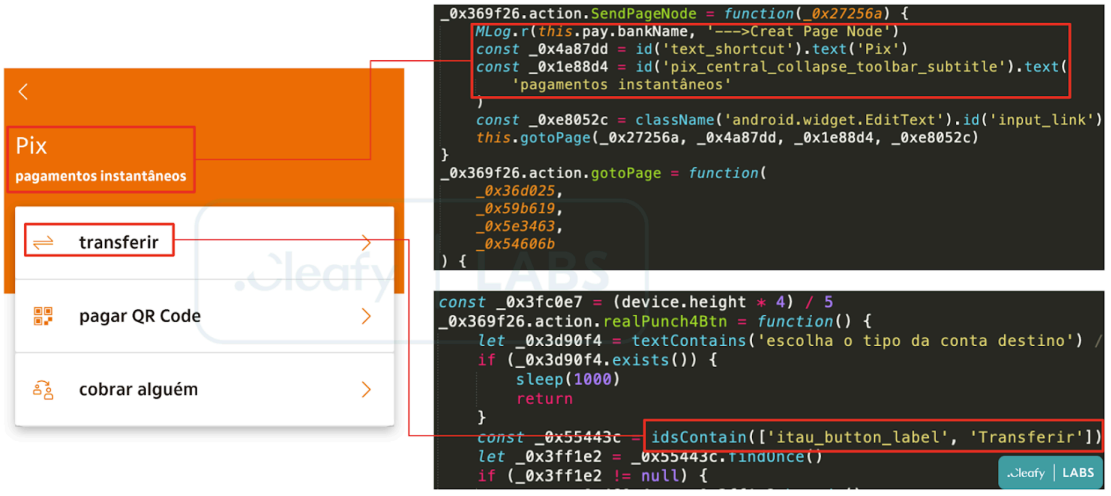

Figure 7 - An example of code used to identify the elements of the targeted bank

- The **balance discovery** of the account, used to set up the amount of money to steal (balance available * 0.95) or discard accounts considered invalid. Followed by the single steps to carry out the transaction.

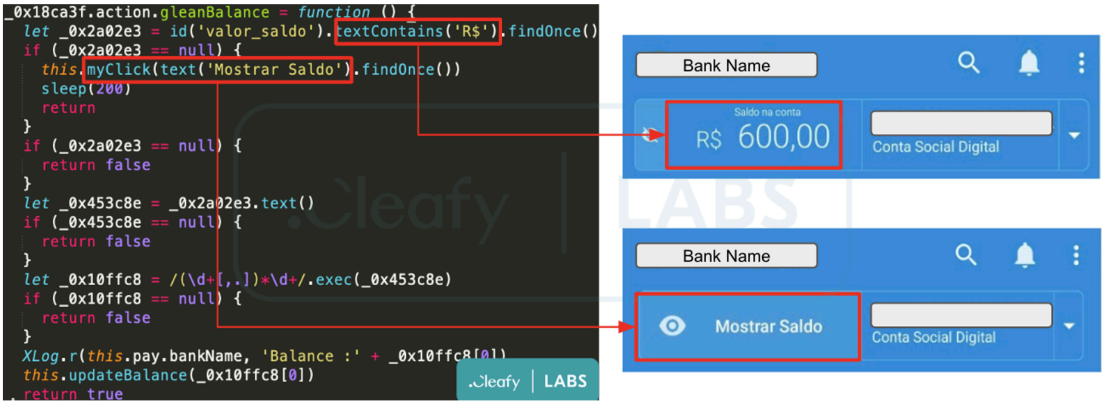

Figure 8 - A portion of code used to identify the balance of the targeted bank

SMS Hijack

PixPirate also contains a script designed to **delete SMS messages** with specific text content. To perform this action, the malware can observe and detect when the default SMS app is in the foreground and perform actions such as long clicking, clicking the delete button, and confirming the deletion. To hide these activities from the user, PixPirate can display a loading window and mutes the device during the deletion of the messages.

This kind of feature is also present in other Android banking trojans and has the dual function of hiding suspicious actions carried out by the malware and removing specific details from analysts carrying out IR operations.

```
_0x5219e3()
events.observeActivity()
smsHijark = {
  isSmsForeground: false,
  loadingWindow: null,
}
const _0x227173 = android.provider.Telephony.Sms.getDefaultSmsPackage(context)
smsHijark.enterListener = events.onActivityEnter(
```

Figure 9 - A portion of code used in the sms hijack method

Code Protection

Among the main countermeasures adopted by PixPirate to slow down the analysis are code obfuscation and encryption, other than classic functionalities that try to avoid application removal at runtime. Speaking about obfuscation, it has been implemented wisely, making the code quite challenging to be analyzed directly. In fact, before proceeding with the analysis, it was required to remove garbage functions and rename variables other than proceeding with multiple steps of deobfuscation. At the end of this process, it was possible to have a clearer understanding of the code.

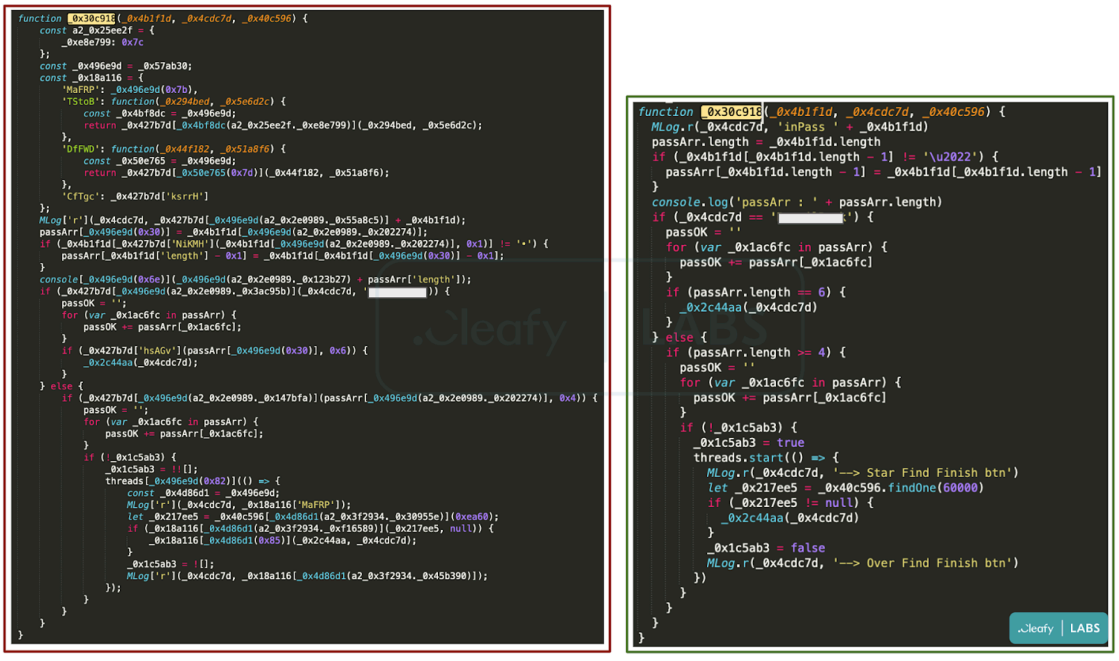

Figure 10 - Comparison of the same function obfuscated (left) and partially deobfuscated (right)

Moreover, TAs have adopted an **encryption** routine provided by **Auto.js** using the Rhino engine and the Common Encryption method. Through this method, most of the strings within the code have been encrypted through a xor operation. An example of the code is given in the following Figure:

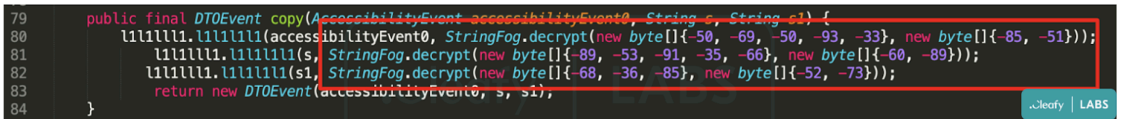

Figure 11 - String encryption with Common Encryption routine

As the reader can infer from the code above, the function **decrypt** takes two-byte arrays as input and performs the xor operation among these values. According to the information retrieved through our analysis, the former parameters of the decrypt function represent the ciphertext; instead, the latter is used as a key. Once the xor operation is completed, it returns a string containing the plaintext value. The code below shows the result of this operation.

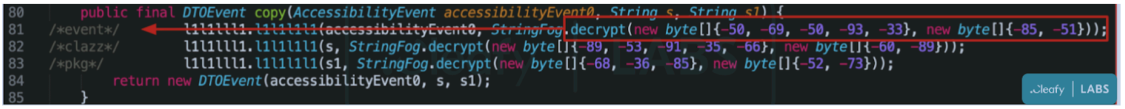

Figure 12 - String decrypted

C2 Infrastructure and Communication

It has been observed that PixPirate and its C2 server use the HTTP protocol for communication, and the data exchanged uses the JSON format. Moreover, TAs adopted certificate pinning, a common technique for preventing man-in-the-middle attacks and securing communications.

We identified two different types of communications, as follows:

- Banking-related communication that embrace the current status of PixPirate (e.g., permissions and also configuration files to instruct it during ATS attacks) as well as targeted bank opened by the victim. An example is given in the Figure:

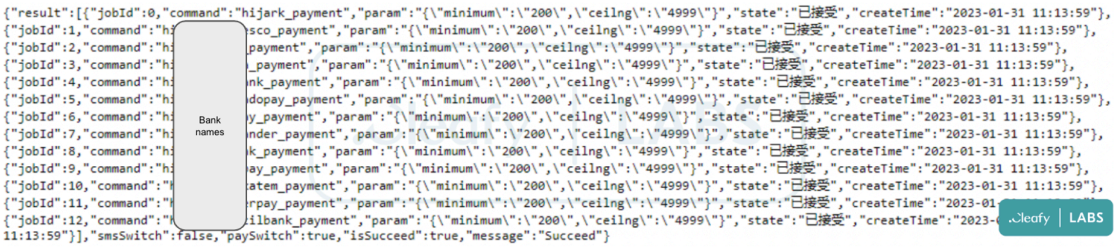

Figure 13 - Initial configuration file

- Debugging communication, including multiple logs of specific errors during the execution, stack traces, etc.

During our analysis, it was possible to find the web panel (shown in the image below) hosted on multiple C2 infrastructures, highly correlated with PixPirate operations.

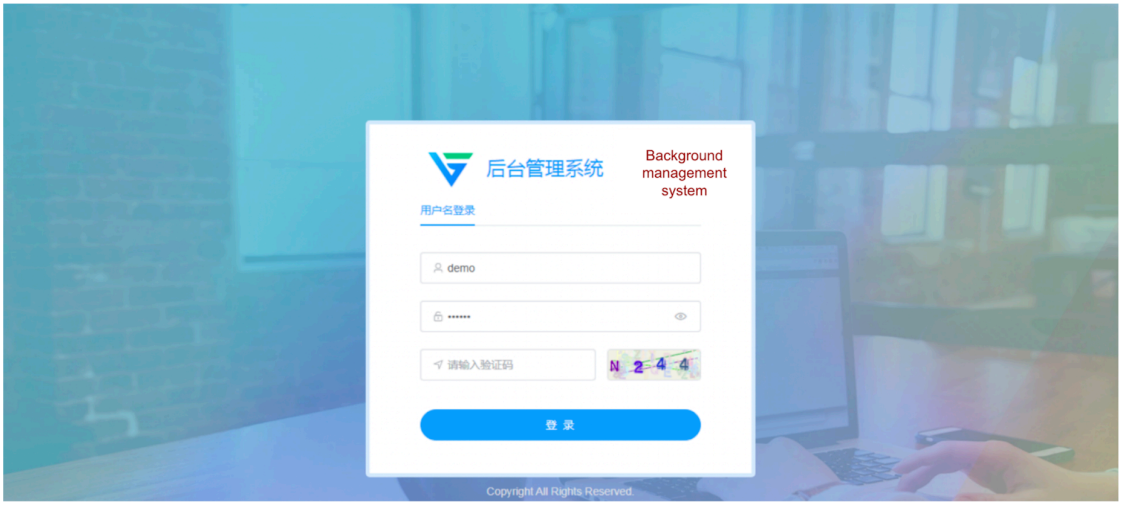

Figure 14 - Login panel

The login page reports the text “Background management system” (written in Chinese), and it appears to be based on an open-source project written in Vue.js, a JavaScript framework for building user interfaces and single-page applications.

Pivoting C2 fingerprints through Internet search engines, such as Shodan, could provide excellent information, and in this case, it confirms that the growing trend began in the second half of 2022:

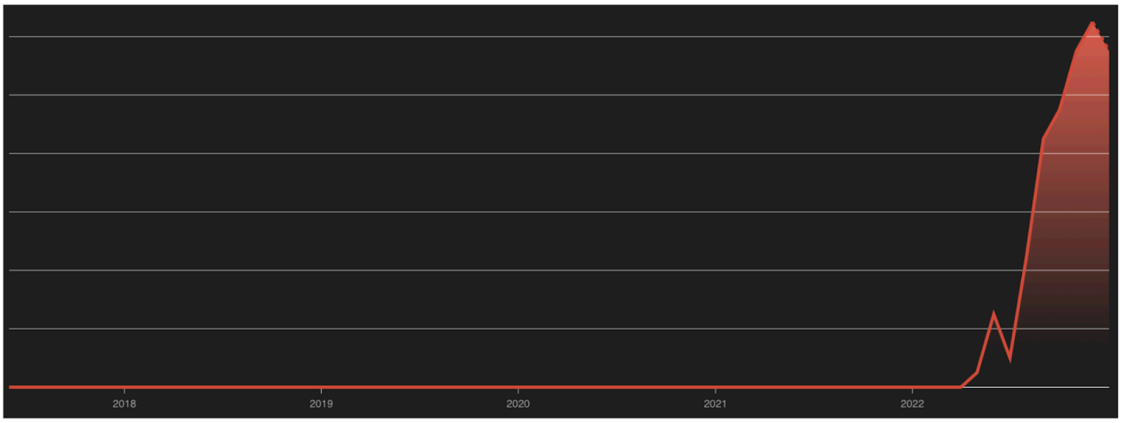

Figure 15 - C2 fingerprinting over time (source: Shodan Trends)

Conclusion

PixPirate represents one of the emerging malware that will try and leverage the double edge blade mechanism related to instant payments.

The introduction of ATS capabilities paired with frameworks that will help the development of mobile applications, using flexible and more widespread languages (lowering the learning curve and development time), could lead to more sophisticated malware that, in the future, could be compared with their workstation counterparts. Additionally, PixPirate has been observed to target the instant payment platform Pix, adopted by multiple Brazilian banks.

Although PixPirate seems to be still in the early stages of development because of the IOCs observed (e.g. logs sent to C2, comments in the code and more variants with very few differences), it's not possible to exclude that in the next future, there will be even more threats that are going to follow the PixPirate example, targeting other LATAM countries or even moving their eyes towards different regions.

Appendix 1: IOCs

IoC	**Description**
cdown883.oss-us-east-1.aliyuncs[.]com	URL used to deliver PixPirate
0b7a66004793b4b976be4e5e21ceeb03	Dropper
ccc18f54f77f5b1295f3b4cc3509cb3b	PixPirate
https[:]//apendgo[.]com/api/	C2
https[:]//applebalanyou.]com/api	C2

Source: https://www.cleafy.com/cleafy-labs/pixpirate-a-new-brazilian-banking-trojan