VB2020 localhost

30 September - 2 October, 2020 / vblocalhost.com

LATAM FINANCIAL CYBERCRIME: COMPETITORS-IN-CRIME SHARING TTPS

Jakub Souček & Martin Jirkal

ESET, Czech Republic

jakub.soucek@eset.cz
jirkal@eset.cz

ABSTRACT

A significant portion of crimeware in Latin America is dominated by banking trojans. Due to many common characteristics, these banking trojans are often treated as one. Our ongoing research clearly shows that that is not the case and that at least 11 distinct malware families reside among them. More importantly, they are constantly evolving and incorporating new tactics, techniques and procedures (TTPs).

Over the course of our research, one thing has become clear: the operators of these banking trojans appear to be in contact with one another. We first spotted this when examining algorithms used for string encryption. Most Latin American banking trojans use very simple, custom encryption schemes that are generally unknown in the broader programming community, and yet we see the same algorithm being used in six different families.

These common features do not end with the binaries' contents. By examining the distribution chains (usually a combination of several stages written in various scripting languages), we find usage of the same obfuscation methods or packers applied to different scripts.

During our research we have encountered some major milestones – changes that affected basically all the families we have identified. We have seen the vast majority of those families transitioning from VMProtect to Themida; both powerful binary obfuscation tools. Similarly, many of them globally switched their initial download method to using *Windows Installer* (MSI) over a period of just a few months.

Finally, some TTPs seem to stay strongly rooted deep inside the region. These include heavy use of ZIP archives and use of DLL side-loading as the favoured execution method.

Even though the sharing of knowledge between cybercriminals is not unusual, the existence of so many examples of it in region-specific malware families with the same focus caught our attention. Our presentation will cover all the common characteristics we have discovered and include a timeline illustrating the evolution of these banking trojans. We will draw conclusions about which families are most closely interlinked and how the modus operandi of Latin American banking trojans is different from banking trojans in the rest of the world.

INTRODUCTION

Dominating crimeware in the region, Latin American banking trojans share so many characteristics that they are conventionally treated as one single malware family. Our ongoing research clearly shows otherwise, identifying at least 11 distinct and concurrently active families: Amavaldo [1], Casbaneiro [2], Grandoreiro [3], Guildma [4], Mispadu [5], Mekotio [6], Zumanek [7], Krachulka, Lokorrito, Numando and Vadokrist. IoCs of all these families can be found on *ESET*'s malware IoC *GitHub* repository [8], and detailed descriptions, including MITRE ATT&CK tables, of several are available in the blog posts referenced above.

Given that we consider these to be different malware families, it may seem surprising that they have so much in common. We believe the reason is that the authors of these banking trojans are in touch with each other, sharing TTPs. In this paper – which would not have been possible without the invaluable contributions of our colleagues in the ESET Prague team, in particular Juraj Horňák and Roman Šíma – we will dissect the most notable similarities that led us to this conclusion.

IMPLEMENTATION

The first area we will focus on is the implementation details of these families. Besides the most notable similarity – they are all written in Delphi – the binaries are so similar in their core functionality that it almost seems like they were built from a single set of blueprints.

Core of a typical Latin American banking trojan's implementation

The typical Latin American banking trojan first collects information about the victim's machine. This usually consists of the computer name, username, some unique identifier and sometimes indicators of whether security or banking protection software is installed. The malware then sends this information to a URL distinct from the C&C server (based on debug information we were able to gather from some binaries, the authors refer to this step as '*Registro*', which translates to Registration).

Once the Registration phase is complete, the banking trojan periodically checks the titles of active windows. If a title matches any of the names hard coded in the binary, the trojan launches its attack. The attack consists of displaying a fake pop-up window crafted specifically for that targeted institution. This window is controlled by an underlying Delphi form and typically tries to persuade the victim to divulge sensitive information.

Additionally, the malware usually tries to make it as hard as possible for the potential victim to get rid of the window by:

- blocking input anywhere else
- keeping the window always on top
- disabling hotkeys

- disabling Task Manager
- blocking mouse manipulation.

The whole process is illustrated by the flowchart in Figure 1.

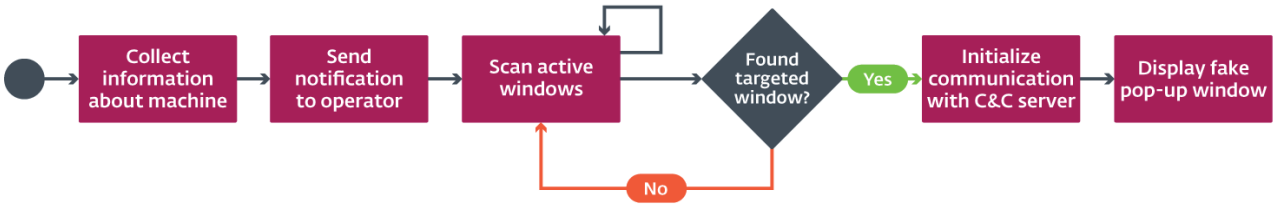


Figure 1: Flowchart showing the core functionality of a typical Latin American banking trojan.

Each of the 11 distinct families we have identified follows this implementation blueprint. We have also seen all of these families being active simultaneously and, while they follow the same logic, we are certain that they are implemented independently.

The implication here is obvious – the authors of these families cooperate. We believe it to be nearly impossible for 11 malware authors to have such specific common ideas without communicating between themselves. And we also don't believe there is one group of malware authors willingly maintaining 11 different pieces of malware with exactly the same logic and goal.

Implementation detail similarities

Even though following the same blueprint is the most significant similarity, it is not the only one. Besides that, Latin American banking trojans share several implementation techniques as well. For example, Amavaldo, Casbaneiro, Mekotio, Mispadu and Vadokrist all base their communication protocols on the custom, third-party remote-control component Delphi Remote Access PC [9]. Casbaneiro and Vadokrist contain identical pieces of code for creating and managing a string table. The vast majority of the malware families rely on the `Magnification.dll` when taking screenshots – a DLL implementing the Windows Magnification API and rarely seen used in other malware.

Most of the families also enable the Desktop Window Manager [10] and disable *Google Chrome* hardware acceleration by changing its `%LOCALAPPDATA%\Google\Chrome\User Data\Local State` configuration file. We strongly believe that this is an attempt to avoid graphics issues when displaying the fake pop-up windows.

String encryption and obfuscation

Most of the Latin American banking trojans use custom encryption schemes, which could suggest that the authors come up with the algorithms by themselves. However, we do not believe that is the case, as one such encryption scheme [11] is used in six distinct families (Casbaneiro, Grandoreiro, Guildma, Numando, Mekotio and Zumanek) and to the best of our knowledge has not been seen used in other malware. Other encryption schemes are shared as well, although not so significantly.

Besides encryption, the only two commonly seen string obfuscation techniques are using a string table or splitting the string into multiple parts and then using string concatenation to join them when needed (see Figure 2). The latter method usually protects the string decryption key.

```
lea     edx, [ebp+var_28]
mov     eax, 248h
call    System_Sysutils_IntToStr
push    [ebp+var_28]
push    offset aHg        ; "HG"
lea     edx, [ebp+var_34]
mov     eax, offset asc_6175B0 ; "   "
call    System_Sysutils_Trim
mov     ecx, [ebp+var_34]
lea     eax, [ebp+var_30]
mov     edx, offset asc_6175C4 ; "  "
call    System_UStrCat3
mov     eax, [ebp+var_30]
lea     edx, [ebp+var_2C]
call    System_Sysutils_Trim
push    [ebp+var_2C]
lea     edx, [ebp+var_38]
mov     eax, 1E4h
call    System_Sysutils_IntToStr
```

```
mov     eax, 0Dh
call    GetStringByIdx  ; C113DB75EC29D558DD3A9B2CEF052B698FC40A
mov     edx, [ebp+var_4C]
mov     ecx, offset aCas ; "CAS"
xor     eax, eax
call    sub_239CCC
mov     edx, [ebp+var_48]
mov     eax, offset dword_672D30
call    System_UStrAsg
lea     eax, [ebp+var_50]
push    eax
lea     ecx, [ebp+var_54]
xor     edx, edx
mov     eax, 0Eh
call    GetStringByIdx  ; 02066EAEED6F
```

Figure 2: Methods of string manipulation used in Latin American banking trojans – string concatenation (left) and string table (right).

Common enemy: protection software

In Latin America, there are two common security products related to banking institutions. The first is *Trusteer*, developed by *IBM*, and it provides authentication and protection against fraud. The second is called *Warsaw*, or *GBPlugin*, and is developed by *GAS Tecnologia*[1]. Quite a few Latin American banks [12] require the latter product to be installed on their users' devices, to provide secure access to online banking services.

Naturally, Latin American banking trojans have to deal with these products in some way. Some just check whether they are installed and report that information during the Registration phase or quit. Some try to protect themselves, mainly by hooking *Windows* APIs to prevent those products from being injected. The rest go even further and try to kill those products. We have seen this done by:

- renaming file system paths
- blocking the products at the firewall level
- tampering with the files' ACLs to prevent them from running
- using a dedicated driver to remove crucial files.

Binary obfuscation

Authors of these banking trojans are fond of using VMProtect, a powerful binary obfuscation tool. In 2017, many of the banking trojans we saw relied on this tool. However, its popularity started to decline in 2018 and today we rarely see it, although some families still use it.

Those that do not use VMProtect seem to have replaced it with one of its competitors – Themida. The popularity of this tool seems to be increasing and more families are experimenting with it.

DISTRIBUTION

Even though there are quite a few similarities in the implementation details, it doesn't end there. In this section, we will focus on similarities in distribution chains.

Typical Latin American banking trojan distribution chains

The initial attack vector is typically a spammed link or attachment or malvertising (as in the case of Mispadu, which we describe in detail in our blog post [5]). The attack starts with one malicious file that is a downloader written either in Delphi or in a scripting language, or occasionally an *Office* document with an embedded malicious macro. For the attack to be successful, the potential victim must download and execute the attachment or file (often inside a ZIP archive).

When executed, this file can lead to subsequent stages that are typically designed only to download the next stage until the final stage is reached. Delphi downloaders usually consist of a single stage, while script downloaders tend to use multiple stages written in various scripting languages.

In the vast majority of these families, the logic in each variant's final stage is almost identical. It typically checks for a *marker* first. A marker is a unique object, usually a file in a specific directory or a Registry key or value, created only by that stage to see whether the malware has already compromised the machine. If not found, it continues by downloading a ZIP archive.

The ZIP archive is something very typical for the distribution chains of Latin American banking trojans. We have observed only a negligible number of chains that did not utilize one. An interesting, atypical example is the Mispadu family's final stage, which, even though it downloads the components independently, wraps each one in a separate ZIP archive.

When the final stage downloads the archive, it follows by:

- extracting its contents
- installing the malware to the specified location
- executing it
- sometimes also setting up persistence (either by using a Run key or LNK file).

The whole process is illustrated in Figure 3.

[1] GAS stands for Global Antifraud Solution.

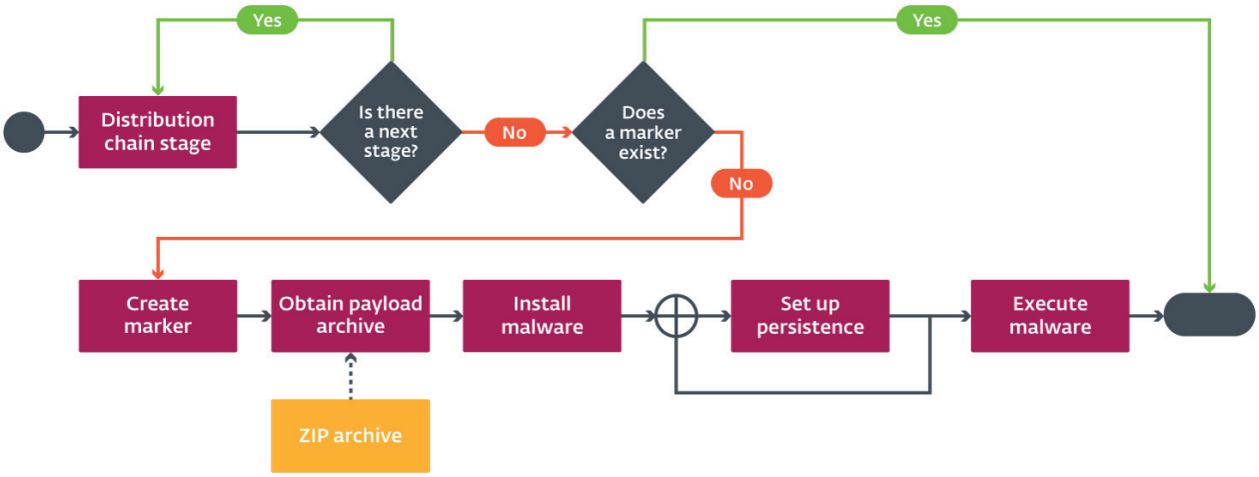


Figure 3: Flowchart of a typical distribution chain used by Latin American banking trojans.

Every Latin American banking trojan's distribution chain more or less follows this logic. We have seen components of this chain and its stages implemented in Delphi, VBScript, JavaScript, PowerShell, AutoIt and batch script. Despite the number of different implementations, the logic remains the same for the majority of the families we have analysed.

It might appear as if some threat actor is implementing these chains and providing distribution for the banking trojan operators. However, if that were the case, this actor would have to implement several distinct chains for each family. Similar as they might be, each family has its own set of distribution chains that it tends to use. Additionally, the chain is very tightly connected to how the banking trojan is executed. We have never observed any of these chains distribute anything other than the Latin American banking trojans we have analysed. That is why we believe the authors of the families write the chains themselves and share information with each other, similar to the way in which they share implementation details.

Sharing the chains

However, there is an even more interesting hint of cooperation. Sometimes, we observe a distribution chain that we know to be used by one Latin American banking trojan but that ends up downloading a different trojan. We have encountered this too many times for it to be a coincidence. To be specific, in our telemetry data we have seen:

1. The same PowerShell script download Casbaneiro, Mekotio and Vadokrist
2. The same JavaScript code download Mekotio and Vadokrist
3. The same chain of *four* consecutive stages being used by Mekotio and Vadokrist
4. The same Delphi downloader downloading Mekotio and Grandoreiro
5. The same Delphi downloader downloading Mekotio and Casbaneiro
6. The same Delphi downloader downloading Grandoreiro and Vadokrist.

We strongly believe that not only do these authors share knowledge, but when it comes to distribution chains, they share the downloaders as well.

The first link in the chain

When we started our research in 2017, LNK files were the favoured initial malicious files. This changed during 2019 with the coming of a new preferred method – using *Windows Installer* (MSI) files. Almost all of the 11 families we have analysed switched to using MSI during 2019 and it remains the most commonly used method at the time of writing. Therefore, we believe it deserves a little deeper explanation.

In 2000, *Microsoft* devised MSI to organize the installation, uninstallation and update of applications running on *Microsoft Windows* operating systems. The format allows for a customized execution, defined by an XML file, during compilation.

Authors of the Latin American banking trojans seem to use *Advanced Installer*, a commercial authoring tool designed for easy creation of MSI files. There are three main ways these malware authors utilize MSI:

1. Embedding a Delphi DLL that the MSI will execute

2. Directly instructing the MSI to download a file from a supplied URL and execute the response
3. Embedding a script (JavaScript and VBScript are the most commonly used) that the MSI will execute.

Script obfuscation

Some authors sometimes obfuscate the scripts used as distribution chain stages. They use a very small pool of obfuscators for this purpose. Unsurprisingly, the authors share some of these between themselves. To be specific, we have seen:

1. The same PowerShell obfuscator used on four different types of scripts for Amavaldo and Casbaneiro
2. The same JavaScript obfuscator used on three different types of scripts for Casbaneiro, Mekotio and Vadokrist
3. The same VBScript obfuscator used on two different types of scripts for Casbaneiro and Lokorrito.

Targeted countries

The name 'Latin American banking trojan' may soon become redundant, as these banking trojans have started expanding beyond Latin American borders – to Europe. We have observed increasing activity of these families in Spain and Portugal. The obvious reason is the language similarity between Spanish and Portuguese. Many of the banking institutions based in Latin America also have offices in those countries.

The families that have started to expand in this way have done so almost all at the same time. Grandoreiro started this expansion (first attempts in July 2019 and bigger campaigns since October 2019 according to *ESET* telemetry), followed by Casbaneiro (February 2020), Mispadu (February 2020) and Mekotio (March 2020). That leads us to the unverified suspicion that this also could be a coordinated move.

We have also noticed a surprising behaviour lately – the trojans look for window titles related to other language variants of the banking applications, such as French or German. However, we have observed no activity of these families in other European countries. We believe the goal of the authors is to determine how popular those language variants are in the countries they already target, as the fake pop-up windows are still all in Portuguese or Spanish.

EXECUTION

We have already mentioned that the final stage downloads a banking trojan inside a ZIP archive, but we have not described how the payload ends up being executed. Commonly, the banking trojan is not the only entry in the archive. During our research, we came across 14 different methods of execution.

As you surely suspect by now, we have observed some of these methods in multiple families. In this section, we will focus on those.

Method 1: Direct execution

Unsurprisingly, the easiest method of execution is simply to execute the banking trojan directly (see Figure 4). Using this method there is no need for additional components, yet in several cases the archive contained legitimate support DLLs too. We have observed this method used by Casbaneiro, Mekotio and Zumanek in the past, but we rarely see it today. Although the direct execution method is simple, other methods are more popular (surprising though that may be).

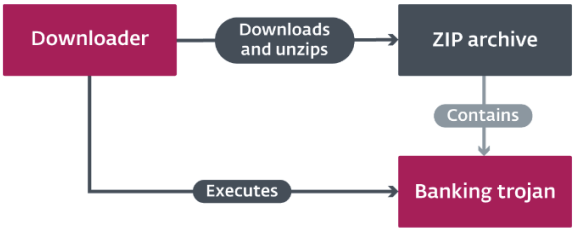


Figure 4: Direct execution method.

Method 2: Using the AutoIt interpreter

In this case, the ZIP archive contains three files: a legitimate AutoIt interpreter, an AutoIt injector or loader script, and the banking trojan. The final stage of the distribution chain executes the AutoIt interpreter and passes the injector or loader script to it as an argument. The script then executes the banking trojan (see Figure 5). This method has been used by Casbaneiro, Vadokrist and Mekotio, and is still in use today.

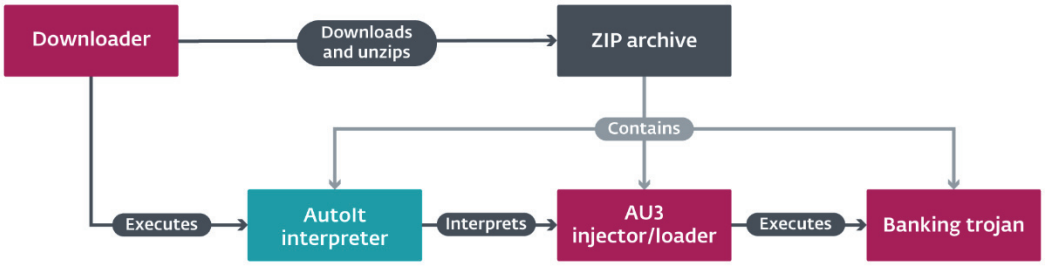


Figure 5: Execution method using a legitimate AutoIt interpreter and an AutoIt injector or loader script.

Method 3: DLL side-loading

This is by far the most popular execution method. The ZIP contains a legitimate application and a banking trojan DLL. The final stage places both files in the same folder and executes the legitimate application which, unknowingly, executes the banking trojan via DLL side-loading (see Figure 6).

At least six families have used this method: Casbaneiro, Krachulka, Lokorrito, Mekotio, Numando and Vadokrist.

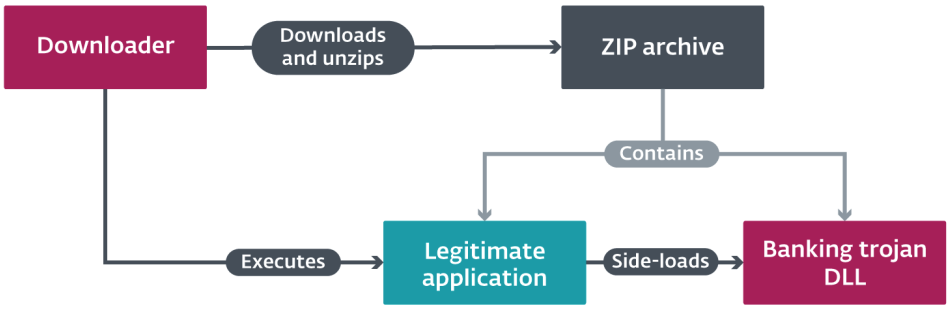


Figure 6: Execution method using the DLL side-loading technique.

Method 4: DLL side-loading combined with injector

In this modified version of the previous execution method, the ZIP archive contains an additional entry – an injector DLL. As before, the final stage executes the legitimate application. However, it does not side-load the banking trojan, but rather the injector. The injector may sometimes need to decrypt the banking trojan before ultimately injecting it into some process (see Figure 7).

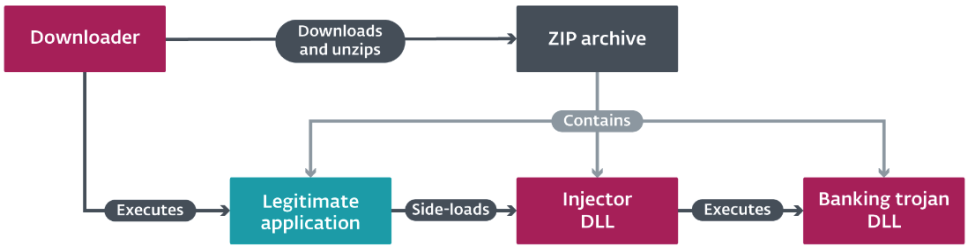


Figure 7: Execution method using DLL side-loading to execute an injector responsible for running the banking trojan.

At least four families use this method consistently (Amavaldo, Casbaneiro, Mekotio and Vadokrist). An interesting aspect of these families is that they occasionally (but not exclusively) use exactly the same type of injector.

Legitimate applications being abused

Since DLL side-loading is so heavily used as the favoured execution method, it is worth looking at what applications are being abused for this purpose. During our research, we have observed overall 22 different applications belonging to *Microsoft*, *Oracle*, several security companies, *NVIDIA*, *VMware* and others. They are listed in Table 1.

Product	Filename	DLL name
Microsoft Corporation CTF Loader	ctfmon.exe	MsCtfMonitor.dll
		AppGetLoader.dll
		CryptUI.dll
Microsoft Corporation OLE/COM Object Viewer	OLEView.exe	IViewers.dll
Microsoft ECM Certificate Manager	CertMgr.exe	CryptUI.dll
Microsoft Office Picture Manager	Ois.exe	MSOCF.dll
Java(TM) Platform SE 8 (cmd-line launcher)	jjs.exe	jli.dll
Java(TM) Platform SE 8 (Remote Method Invocation)	java-rmi.exe	jli.dll
Java(TM) Platform SE 8 (Kerberos)	kinit.exe	jli.dll
Avira	Avira.SysTrayStartTrigger.exe	Avira.OE.NativeCode.dll
Avast Dump Process	avDump32.exe	Dbghelp.dll
AVG Dump Process	avDump32.exe	Dbghelp.dll
G DATA Personal Firewall	GDFwAdmin.exe	GDFwAdmin.dll
G DATA Security Software	AVK.exe	Avk.dll
COMODO Internet Security	CisTray.exe	Cmdres.dll
NVIDIA 3D Vision Test Application	Nvsttest.exe	D3d8.dll
NVIDIA Smart Maximise Helper Host	NvSmartMaxApp.exe	NvSmartMax.dll
VirtualBox Guest Additions Tray Application	VBoxTray.exe	Mpr.dll
VMware NAT Service	Vmnat.exe	Shfolder.dll
WinGup for Notepad++	Gup.exe	Libcurl.dll
Disc Soft Bus Service Pro (DAEMON Tools Pro)	DiscSoftBusService.exe	Imgengine.dll
Bartels Media GmbH Macro Recorder	MacroRecorder.exe	Mrkey.dll
Stonesoft VPN Client Service	Sgvpn.exe	Wtsapi32.dll
OOO Lightshot Starter Module	Lightshot.exe	Lightshot.dll

Table 1: List of legitimate applications abused for DLL side-loading by Latin American banking trojans.

As we have mentioned, these applications are distributed together with the banking trojan. Therefore, it does not need to rely on them being present on the target machine. Interestingly, we have seen many of the applications mentioned in Table 1 (sometimes even the same hash) being abused by multiple families.

FAKE POP-UP WINDOWS

Given so many common features, one might be inclined to think that the authors of these banking trojans share the fake pop-up windows too, since they are designed to attack customers of the same banks. In fact, the opposite seems to be the case. This is likely the one thing they do by themselves. We have analysed around 600 of the most recent of these fake windows and it seems they are unique to each family.

Several of the authors seem to have a sort of graphic template that remains the same and the content is different for each targeted bank. For an example of such a template, refer to Appendix B. They are also inspired by the same sources – official websites, *YouTube* videos and probably even access to actual banking applications themselves (see Appendix C).

CONCLUSION

In this paper, we have discussed Latin American banking trojans. We have shown that the implementation of these malware families looks suspiciously alike. Very specific parts of code – such as disabling hardware acceleration in *Google Chrome*,

enabling Desktop Window Manager or using `Magnification.dll` to take screenshots – are almost identical across multiple families.

Distribution chains of these malware families also look alike. Occasionally, we have observed one family borrowing a distribution chain from a different one. In the case of scripts, different authors tend to use the same obfuscators.

Finally, the banking trojans are even executed similarly. Besides sharing some unusual execution methods, they abuse the same legitimate applications for DLL side-loading.

You may have noticed that we have mentioned some families significantly more than others. Indeed, Casbaneiro, Mekotio and Vadokrist seem to be the most interlinked families. Krachulka seems to be the family that shares the least with the rest of the families.

Since we believe it is impossible for 11 different authors to have come up with so many common ideas and we don't believe that one group is deliberately maintaining 11 different families at the same time, we conclude that the authors of these banking trojans communicate with each other. This cooperation is extensive and it affects the vast majority of the families we have analysed. Such tight collaboration between malware families that share the same goal, are region-specific and are, in fact, expected to be competitors, is something we have never encountered before.

REFERENCES

[1] ESET Research. From Carnaval to Cinco de Mayo – The journey of Amavaldo. WeLiveSecurity. 1 August 2019. https://www.welivesecurity.com/2019/08/01/banking-trojans-amavaldo/.

[2] ESET Research. Casbaneiro - Dangerous cooking with a secret ingredient. WeLiveSecurity. 3 October 2019. https://www.welivesecurity.com/2019/10/03/casbaneiro-trojan-dangerous-cooking/.

[3] ESET Research. Grandoreiro: How engorged can an EXE get? WeLiveSecurity. 28 April 2020. https://www.welivesecurity.com/2020/04/28/grandoreiro-how-engorged-can-exe-get/.

[4] ESET Research. Guildma: The Devil drives electric. WeLiveSecurity. 5 March 2020. https://www.welivesecurity.com/2020/03/05/guildma-devil-drives-electric/.

[5] ESET Research. Mispadu: Advertisement for a discounted Unhappy Meal. WeLiveSecurity. 19 November 2019. https://www.welivesecurity.com/2019/11/19/mispadu-advertisement-discounted-unhappy-meal/.

[6] ESET Research. Mekotio: These aren't the security updates you're looking for… WeLiveSecurity. 13 August 2020. https://www.welivesecurity.com/2020/08/13/mekotio-these-arent-the-security-updates-youre-looking-for/.

[7] Puodzius, C. Zumanek: novo malware tenta roubar credenciais de serviços das vítimas. WeLiveSecurity. 17 January 2018. https://www.welivesecurity.com/br/2018/01/17/zumanek-malware-tenta-roubar-credenciais-de-servicos/.

[8] ESET. https://github.com/eset/malware-ioc.

[9] Wagner, M. Delphi_Remote_Access_PC. https://github.com/abalad/Delphi_Remote_Access_PC.

[10] Glenn, W. How-To Geek. What Is Desktop Window Manager (dwm.exe) and Why Is It Running? 4 July 2017. https://www.howtogeek.com/howto/windows-vista/what-is-dwmexe-and-why-is-it-running/.

[11] MASTER_ZION. Mestres da Espionagem Digital. Sao Paulo : Digerati Books, 2008, pp. 8-11.

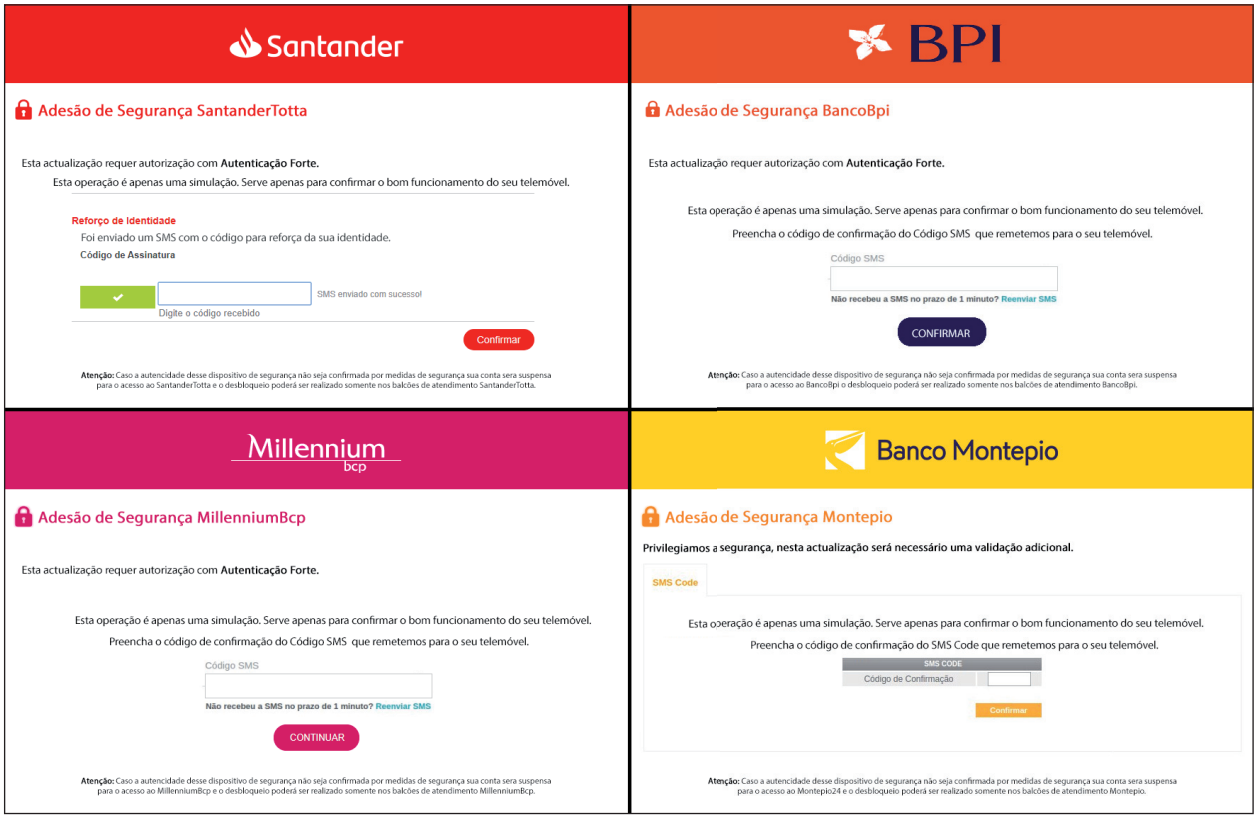
[12] https://diagnostico.gasantifraud.com/.

APPENDIX A

SHA-1	Description
B855D8B1BAD07D578013BDB472122E405D49ACC1	Win32/Spy.Amavaldo.N
9DFFEB147D89ED58C98252B54C07FAE7D5F9FEA7	Win32/Spy.Casbaneiro.AJ
BD88A809B05168D6EFDBA4DC149653B0E1E1E448	Win32/Spy.Grandoreiro.AJ
A7B10B8DE2B0EF898CFF31FA2D9D5CBAAE2E9D0D	Win32/Spy.Guildma.BS
896AB7BF0DAFC7980DB9210E2DFE5FC14BF1344D	Win32/Spy.Krachulka.C
20833DADE1DBA9989DB6B792999FEBAA7FEA866C	Win32/Spy.Lokorrito.L
269D353DFB585DCFFE1F908BD9768E24CC0DAA66	Win32/Spy.Mekotio.BS
A8CD12CC0BBD06F14AA136EA5A9A2E299E450B18	Win32/Spy.Mispadu.C
FC3190CC2EF34F86A594985E7C9BDB781E724CA5	Win32/Spy.Numando.D
E1BA66272CF09F109AC5F8497E1AF85FF2E38C6B	Win32/Spy.Vadokrist.O
AD4ABB8B471139F379A5E6A60A77C4EF5347AAA4	Win32/Spy.Zumanek.CR

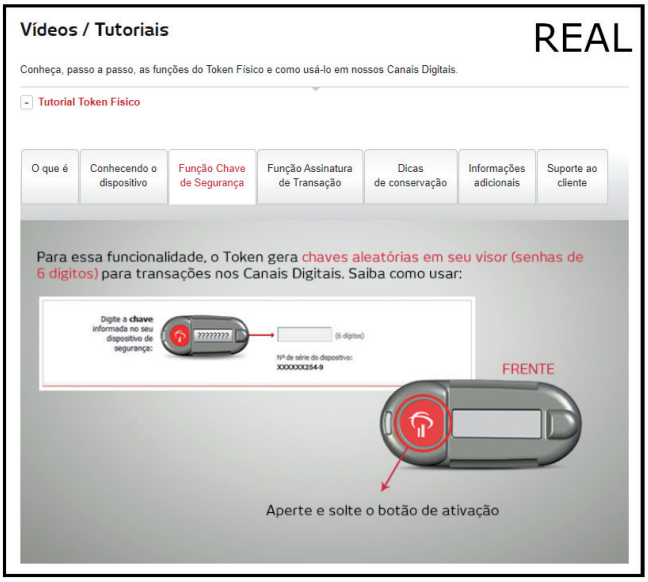
APPENDIX B

Template used by Zumanek

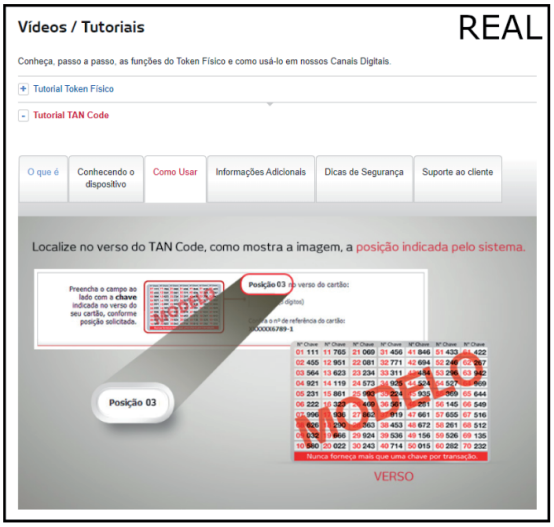


APPENDIX C

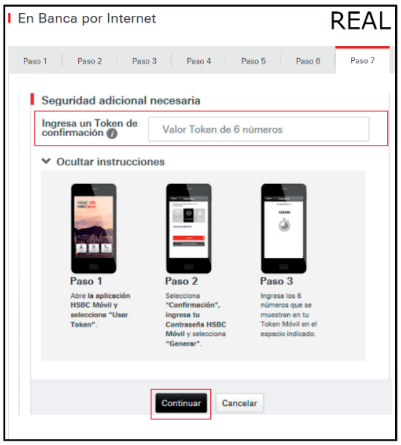
Example 1

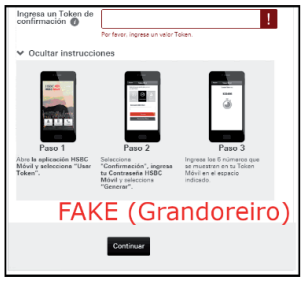


Example 2



Example 3





Example 4