New Remote Overlay Malware Highlights Rising Malcode Sophistication in Brazil

securityintelligence.com/client-maximus-new-remote-overlay-malware-highlights-rising-malcode-sophistication-in-brazil/

January 10, 2017

Home&nbsp/ Banking & Finance

Client Maximus: New Remote Overlay Malware Highlights Rising Malcode Sophistication in Brazil

Banking & Finance January 10, 2017

By Or Safran co-authored by Lior Keshet , Limor Kessem 7 min read

The Brazilian malware landscape is notorious for its plethora of Delphi-based code and overall lack of sophistication. But much like the Russian-speaking malware scene, Brazilian cybergangs have been using better malware, such as the recently discovered Client Maximus, in their attacks.

In the summer of 2016, malware like Zeus Panda and Sphinx were spotted in Brazil. Those were followed by intensifying infection campaigns, all targeting Brazilian banks and payment platforms, according to data from IBM X-Force Security Research. Later in the year year, our researchers spotted a new, real-time phishing attack in Brazil, which introduced automation and agility to classic phishing attacks.

It quickly became clear that cybercriminals in Brazil are collaborating with counterparts from other, more sophisticated threat landscapes and importing code and expertise to launch attacks in their own country.

Enter Client Maximus

We recently encountered a case in Brazil that reflects this ongoing trend. Client Maximus is a new malware code that appears to have been written specifically for attacks on Brazilian banks. The malware was recently analyzed alongside other components related to it, leading to further findings and a greater overall understanding of the growing sophistication of cybercrime tools in Brazil.

The purpose of the Client Maximus malware is financial fraud. As such, its code aspires to create the capabilities that most banking Trojans have, which allow attackers to monitor victims' web navigation and interrupt online banking session at will. After taking over a victim's banking session, an attacker operating this malware can initiate a fraudulent transaction from the account and use social engineering screens to manipulate the unwitting victim into authorizing it.

Read the white paper: How to outsmart Fraudsters with Cognitive Fraud Detection

Initial Infection Routine

The initial infection begins with a malicious, obfuscated WScript. Upon deobfuscating the script, we discovered that its goal is to download and execute JavaScript on the target endpoint.

Figure 1: Initial script run by new remote overlay malware

The JavaScript fetched by the malicious code here is a larger snippet that runs several antivirus checks and eventually downloads and executes the actual malware. The malware aims to:

1. Monitor which bank is accessed in the user's open browser windows;
2. Remotely control the user's endpoint at will; and
3. Overlay fake persistent windows on top of the user's web browser application.

The attacker sets up the malware to commit financial fraud.

Malware Deployment

Client Maximus is deployed on the endpoint in two different ways. The first method calls for it to be deployed as a custom NSIS Installer. The malware binary is a custom-compiled version of Nullsoft Scriptable Install System (NSIS), which extracts three files into %TEMP% folder. The first file is system.dll, the second file is a malicious dynamic link library (DLL) and the third is the encrypted payload.

System.dll is part of a plug-in for the NSIS installer. It's actually a legitimate DLL the malware leverages to call its own malicious DLL. After the malware loads the malicious DLL into its own process memory space and calls it using the aforementioned plug-in, the malicious DLL is used to decrypt the payload and load it into the process memory. The activity is subsequently performed by the now-decrypted file.

The Client Maximus malware binary is usually deployed to the current user's AppData folder with a unique name that depends on the dropper. Unlike most financial malware, the malware itself creates its persistence from the location it runs from. It neither generates a new name nor copies itself to a new location.

In most cases the malware’s persistence is set under the endpoint’s run key (HKCU hive), pointing to the malware’s NSIS binary. In some cases, persistence is achieved by setting a link file to the malware binary’s path under the user’s startup folder.

Client Maximus can also be deployed via DLL hijacking and new technology file system (NTFS) alternate data streams (ADS). In this second deployment style, the malware uses a legitimate binary file, signed by a leading technology vendor, with a malicious DLL in the same folder.

DLL Hijacking Dresses Malware Up in Sheep’s Clothing

While there are many forms of DLL hijacking, the author of Client Maximus used the search path method to achieve it. This method relies on the binary to load DLLs from the application’s current working directory before searching in other directories. It then replaces the original DLL the application tries to load with another DLL, which causes the application to load a different DLL from the original, legitimate program intended to load.

In our case, when Client Maximus is deployed on the infected system, it also creates a malicious DLL, the name of which is determined by the internal name of the aforementioned legitimate library. That causes the legitimate executable to load the malicious DLL and execute it while creating the illusion that the original, harmless file is performing normal activity.

Below are some examples of legitimate binaries and the malicious DLL they hijack:

- **agestore.exe** (signed by Microsoft) with malicious DLL name dbghelp.dll
  Malicious DLL MD5: 31e2cd4728be52c50b8501e91d062cfc
- **AcroTextExtractorexe** (signed by Adobe) with malicious DLL name query.dll
  Malicious DLL MD5: 94a72a13eec55a8a7db64e468b17875e
- **cermgr.exe** (signed by Microsoft) with malicious DLL name cryptui.dll
  Malicious DLL MD5: 3dbe5192c62fad86033d596296865221
- **vprintproxy.exe** (signed by VMware) with malicious DLL name vmwarebase.dll
  Malicious DLL MD5: 35e1218a4a2c45005e73ccc0e511691c

The malicious DLL inside the legitimate process tries to load the encrypted payload from an NTFS ADS. We assume the malware author used NTFS ADS to remain hidden from the user or antivirus tools, although this method is not new.

We can use the Sysinternals Process Monitor to see when it actually loads the NTFS ADS while Windows continues to show this file as empty. In the image below, e2iU0L appears to be of size 0 KB, although it does have contents in reality. We can also see that a file named dbghelp.dll is being used to perform the DLL-hijacking.

Figure 2: Filename dbghelp.dll being used to perform the DLL-hijacking

In the next image, we can see that the malware (s9TLNV.exe) reads the encrypted payload using an NTFS-stream:

Figure 3: Malware file (s9TLNV.exe) reading the encrypted payload using an NTFS-stream

Client Maximus: The Fraud M.O.

After the malware is fully installed on the target endpoint, it is designed to allow a remote attacker to take control of the endpoint based on a triggered action from the victim. Much like other financial malware, the trigger is navigation to a targeted bank's website.

Monitoring Window Titles to Zero In on Relevant Banks

To determine which sites the victim is accessing, the malware constantly monitors the titles of windows opened in the browser, hashes the titles and compares the hashes against a predefined list of hashes. Once a target match is detected, the malware connects to the attacker's remote server to notify the fraudster, who can then initiate a remote access session using the malware's remote-access Trojan (RAT) component.

Figure 4: The malware prints the victim's windows and sends the data to the fraudster

As a first step in the process described above, the malware monitors the victim's activity, waiting for a targeted banking session to take place in the browser. To enable the monitoring, the malware takes the following actions:

First, it registers to windows events that indicate the creation of new windows or window title change.

Figure 5: Hooking windows events to monitor window title changes

Upon each of these events, the malware checks the new window title, using the SendMessage application program interface (API), with the "WM_GETTEXT" window message, which allows the malware to obtain each window's title.

Figure 6: Monitoring window titles using the SendMessage API

Finally, the malware hashes the obtained window title and compares it to a list of prehashed targets. If the current hash matches one of the entries on the list, the malware initiates a network communication with the fraudster's control server. The author chose to work with hashes in this case to hide the real target list from researchers.

The Logical Flow of the Fraudulent Transaction

The logical flow of the fraudulent transaction is facilitated by overlay screens the fraudster uses to trick the victim into providing authentication elements.

The flow of events for this new malware is no different than previous malicious code categorized under the remote overlay family:

1. The malware monitors browser windows and waits for a target match.
2. The victim accesses a targeted bank, at which point the malware notifies the attacker.
3. The attacker initiates a remote access session to the infected machine.
4. The victim performs the initial login to the bank account.
5. The attacker takes the session over by blocking the victim's access to the browser's screen, then presents the victim with social engineering screens that cannot be minimized and request more information.
6. The victim unwittingly provides the attacker with authentication elements.
7. The attacker finalizes a fraudulent transaction.

Fraud Continues to Evolve

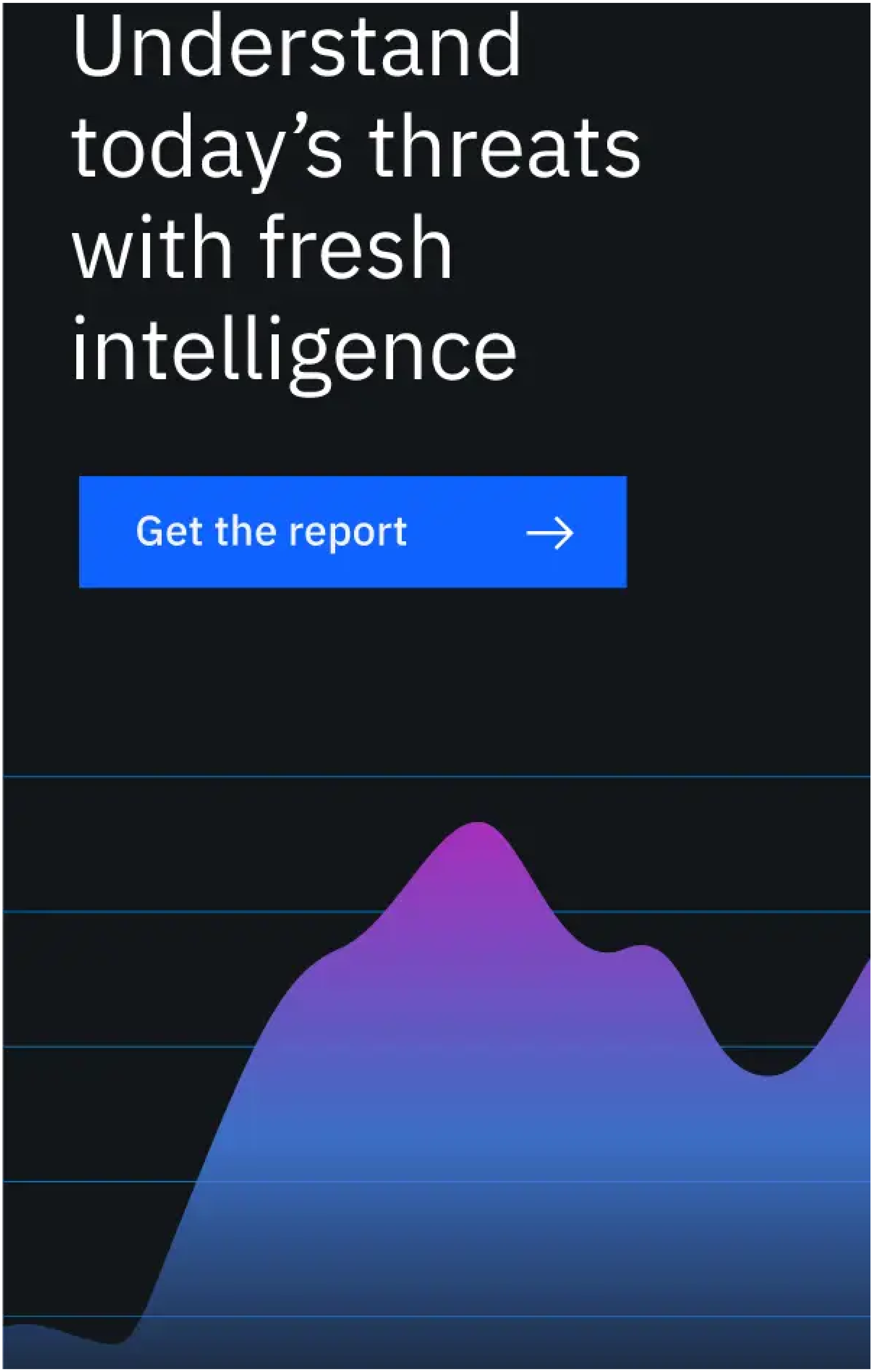
In the past year, we have borne witness to an escalating threat landscape in Brazil, introducing new coding styles to malware projects and revamping popular tools such as the remote overlay Trojan. As we move into 2017, Brazilian banks and payment providers should prepare for continued threat evolution in the country and plan ahead to detect and counter financial threats in new and improved forms.

Banks wishing to protect their customers from evolving threats are invited to learn more about the IBM Security Trusteer Fraud Protection Suite and to read our white paper, "How to Outsmart Fraudsters with Cognitive Fraud Detection."

Individuals looking for tips on protecting themselves from remote overlay malware and other banking Trojans are invited to read our malware mitigation article.

Or Safran
Malware Researcher, IBM Trusteer

Or Safran is a contributor for SecurityIntelligence.



IBM Security