

Inside Astaroth's New Spambot Component

 crowdstrike.com/en-us/blog/inside-astaroths-new-spambot-component

Kevin Ratto



Established Latin American (LATAM) threat actors are continuously adapting their malware capabilities and attack methodologies to circumvent defensive measures and maintain effectiveness against target environments.

Exemplifying these evolving operations, in Q4 2025, operators of the *Astaroth* (aka *Guildma*) botnet introduced a previously unidentified capability: a WhatsApp Web spambot component designed to turn victims into unwitting distributors of the malware by automatically messaging every contact in each victim's WhatsApp contact list.

This blog provides a technical deep dive into the *Astaroth* spambot, examines its overlaps with other recently observed spambots, and explores what this capability expansion signals about the evolving LATAM eCrime ecosystem.

Key Insights

- **New delivery vector:** *Astaroth* operators began distributing a WhatsApp Web spambot component in Q4 2025, marking a significant shift from their traditional email-based spam propagation to a distribution method using trusted social messaging platforms.

- **Invisible browser automation:** By running a browser instance in headless mode and stripping automation indicators from a WebDriver session, the spambot conducts its entire spam campaign without ever displaying a browser window to the victim.
- **Shared codebase:** Identical function names, variable names, file delivery logic, contact-filtering implementations, and configuration field-naming conventions confirm a strong developmental relationship between the *Astaroth* spambot and a spambot dubbed *Vareg* (aka *WATER SACL*, *Eternidade*), which previously distributed other LATAM banking trojans via WhatsApp Web in October 2025 and November 2025 campaigns.
- **Brazil-focused targeting:** Multiple technical indicators — including Brazil-specific phone number filtering, Portuguese-language spam templates, and specific language-oriented HTTP headers — confirm *Astaroth*'s operators are targeting Brazil-based victims.

Background: *Astaroth*

Active since at least 2015, the *Astaroth* banking trojan and information stealer implements a multi-stage infection chain with several loader components and evasion techniques (Figure 1) and now exclusively targets Brazil-based users. *Astaroth* is delivered through a downloader script component — often a Windows shortcut (LNK) file running JScript code — that retrieves an installer component. The installer runs an Autolt-based loader that decodes and executes a Delphi-based loader DLL in memory; this DLL subsequently decrypts and executes *Astaroth*'s core component.

In Q4 2025, the *Astaroth* installer's command-and-control (C2) servers began distributing an additional, previously unidentified component: a WhatsApp Web spambot.

Figure 1. *Astaroth* infection chain

Technical Analysis

Payload Encryption

The *Astaroth* loader components decrypt the spambot component using AES in CTS mode from the open-source Delphi library *DelphiEncryptionCompendium*¹ — the same encryption library *Astaroth*'s developer uses to encrypt the malware's core component.

The encrypted *Astaroth* spambot decrypts to a 32-bit Delphi Portable Executable (PE) (SHA256 hash: d89105c4d567a95f674ed6eac538e32e288b658a4222a3d52e284a77782af4d5; build timestamp: 2026-02-13 04:14:16 UTC).

String Obfuscation

To decode strings, the spambot component uses a complex, six-stage process first observed in *Astaroth*'s core component in early October 2025. This process involves XOR decrypting data, reversing and transforming the decrypted results, and applying custom algorithms to further disguise and then reveal the original data.

Configuration File

The spambot configuration file implements a custom format containing a magic word (EF01), a Base64-encoded and AES-encrypted configuration, and, in the file's final 32 bytes, an HMAC-SHA256 hash value (which serves as an integrity check).

Integrity Check

To verify the configuration file's validity, the spambot removes the file's final 32 bytes and calculates the HMAC-SHA256 hash of the remaining file using the secret key TenHaVunf!@#2026 (observed across all analyzed *Astaroth* spambot samples). If the calculated hash matches the hardcoded value located in the final 32 bytes, the configuration is valid; otherwise, the spambot terminates.

Configuration Decryption

The configuration file is encrypted using AES-CBC with ciphertext stealing (CTS). Once decrypted, the configuration yields a JSON structure containing the fields described in Table 1. The Appendix to this blog post contains a configuration snippet.

Field	Description
ENA	Instructs the spambot to enable or disable itself
FCX	Not used in the analyzed sample
LT	Not used in the analyzed sample
HL	Flag to enable headless mode for browser execution
PFX	Next-stage file name pattern
DEL	Defines the delay time in milliseconds between spam messages
ArrMSG	List of greetings to use in spam messages
ArrMF	List of body text to include in spam messages
ArrURL	List of download URLs highly likely serving <i>Astaroth</i> downloader components

Table 1. Astaroth configuration fields

Spamming Process

WebDriver Download

The spamming process begins with *Astaroth* downloading a browser WebDriver — a legitimate browser automation executable — to handle browser instances and open a WhatsApp Web session. The spambot downloads the driver via a PowerShell (PS) command that retrieves the driver based on the installed browser version.

The *Astaroth* spambot supports Google Chrome and Microsoft Edge browsers.

Browser Profile Directory

The spambot copies the browser profile directory User Data to

C:\Users\Public\Temp\ChromeAuto_<BROWSER_ID><DATE>, where DATE is the current date in yyyyymmdd format and BROWSER_ID is an integer identifying the detected browser. The *Astaroth* spambot uses this folder to spawn a new browser instance with the victim's information, enabling access to sensitive data such as cookies and credentials.

WebDriver Execution

The spambot launches the browser WebDriver as a separate process in a hidden window. The driver is configured to run on a random port between 31175 and 39999, and acts as a local HTTP REST API server that translates WebDriver protocol commands into browser automation actions. The spambot functions as a WebDriver client communicating with this local server to automate WhatsApp Web spam distribution (Figure 2).

Figure 2. The *Astaroth* spambot's internal component communication architecture

Session Creation

After launching the WebDriver executable, the spambot initiates a browser instance via custom command-line browser capabilities sent via an HTTP POST request to the driver's /session endpoint.² The driver replies with a unique session ID used to interact with the browser instance.

The Chromium command line includes several capabilities designed to suppress automation indicators, such as:

- --headless: Runs Chrome without a visible user interface (UI)
- --disable-infobars: Hides the Chrome is being controlled by automated test software banner
- --window-size=1920,1080: Sets browser screen size (even in headless mode), as some websites check this value
- excludeSwitches: ["enable-automation", "test-type"]: Removes flags that identify the browser as an automated instance

The Microsoft Edge command line includes analogous capabilities, with additional flags to suppress sync behavior, logging, and default app creation.

WPPConnect/WA-JS Download

Once the *Astaroth* spambot creates the browser instance, the spambot retrieves a copy of the JavaScript (JS) library WPPConnect/WA-JS — a legitimate library that exports WhatsApp Web functions for customer service automations.³ *Astaroth*'s developer abuses this library to interact with the victim's WhatsApp Web session and gather and spam contacts.

WhatsApp Web Navigation and Verification

After creating the session ID and downloading the JS library, the spambot instructs the automated browser to load WhatsApp Web. Before running its contact grabber functionality, the spambot verifies that one of the following HTML elements exists in the loaded page — highly likely to confirm that the site has fully loaded and that the victim has an active login session:

- `new-chat-outline`: Button to start a new chat
- `settings-refreshed`: Button to access settings
- `chat`: Button to view the victim's WhatsApp Web chats

Contact Grabber

After loading the WPPConnect/WA-JS library, the spambot reads the victim's WhatsApp Web contact list by executing a Base64-decoded script. The script performs the following actions:

1. Retrieves all contacts by invoking `WPP.contact.getAllContacts` (or `WPP.contact.list` if `WPP.contact.getAllContacts` is unavailable)
2. Filters contacts by removing group chats (`@g.us`), broadcast lists (`@broadcast`), linked devices (`@lid`), phone numbers not starting with the Brazil country code prefix 55, non-saved contacts, and the victim's own contact
3. Maps each contact into a list containing the contact's display name (`nome`), phone number (`fone`), and assigned number (`numero`)

Spam Payload Retrieval

Before sending spam messages, the spambot downloads an *Astaroth* payload following this process:

1. Randomly selecting a URL from the configuration field `ArrURL` (Table 1)
2. Replacing a pattern in the selected URL with a random number
3. Setting the user agent to a Chrome browser string
4. Retrieving a Base64-encoded ZIP file via an HTTP GET request

Spamming Contacts

After downloading the payload, the spambot composes and sends spam messages to each collected contact. The process:

1. Parses the victim device's current hour to select an appropriate greeting (e.g., good morning, good evening, or good night in Portuguese)
2. Randomly selects a greeting from `ArrMSG` and a body message from `ArrMF`
3. Composes a JS script including the targeted contact's phone number, name, selected greeting, and body message
4. Prepends the composed JS to a hardcoded Base64-encoded WPP script
5. Executes the script via the WebDriver that sends three messages: a greeting, a malware ZIP file attachment, and a body message

Vareg Spambot Overlap

In October 2025 and November 2025, a threat actor leveraged a spambot dubbed *Vareg* to distribute several LATAM banking trojans — including *Astaroth* — targeting Brazil-based users. However, *Vareg* activity ceased in Q4 2025, closely coinciding with the period when the *Astaroth* botnet began distributing its spambot component.

Over the course of the *Vareg* spambot’s period of activity, its developer rewrote the PS-based spambot in Python while maintaining its functionality, highly likely in an attempt to avoid detections. Technical analysis reveals that *Vareg*’s developer likely used AI to implement the malware, as the scripting codes include patterns commonly associated with AI-assisted coding.

Further analysis of the *Astaroth* spambot and the *Vareg* spambot's Python and PS versions reveals extensive code overlaps, shown in Table 2.

<i>Astaroth</i> 's Field	<i>Vareg</i> 's Field	Translation	Description
FCX	filtro_contatos_excluir	Filter to exclude contacts	Filters variable to exclude contacts
LT	limite_teste	Limit test	Sets a threshold for the maximum number of users to spam
HL	modo_headless	Headless mode	Enables headless browser mode
DEL	delay_entre_mensagens	Delay between messages	Defines the delay in milliseconds between consecutive spam messages
ArrMSG	mensagem_saudacao	Greeting message	Specifies the selected greeting for the spam message
ArrMF	mensagem_final	Final message	Specifies the selected body for the spam message

Table 2. *Astaroth* and *Vareg* configuration overlaps

The overlapping code between *Astaroth* and *Vareg* includes identical message delivery implementations with the same function and variable names, including `enviarParaContato` and `mensagemSaudacao`. Both spambots implement the same contact-grabber logic to filter non-valid contacts, with the *Astaroth* spambot adding an additional check to filter out non-Brazil-based phone numbers. Both spambots also implement identical delay logic — with *Astaroth*’s delay implemented in Delphi and *Vareg*’s delay implemented in Python — with the same minimum (50 milliseconds) and maximum (200 milliseconds) values.

Assessment

Astaroth's new spambot component marks a major capability expansion for the malware's developer — an established LATAM-focused threat actor — by shifting from traditional email-based spam propagation to leveraging trusted social messaging platforms to enhance the spam messages' apparent credibility and reach a broader set of victims.

The spambot seamlessly integrates into the existing *Astaroth* infection chain, and reuses the same encryption, obfuscation, and encoding techniques employed in the core component. This suggests the spambot was highly likely developed by the initial *Astaroth* developer, rather than acquired from or outsourced to another developer. This assessment is made with high confidence based on technical analysis of *Astaroth*'s core component and spambot component.

The code overlap between the *Astaroth* and *Vareg* spambots highly likely indicates that the same threat actor developed both spambots or that the actors operating the two toolsets have a code-sharing arrangement. This assessment is made with moderate confidence based on technical analysis of the two spambots and on analysis of campaigns leveraging them during Q4 2025.

Whether *Astaroth*'s operator conducted the October 2025 and November 2025 *Vareg* campaigns presently remains unknown; however, the cessation of *Vareg* activity in Q4 2025 — closely coinciding with the date the *Astaroth* botnet began distributing its spambot component — likely indicates that the *Vareg* operation was incorporated into *Astaroth* activity. This assessment is made with low confidence based on limited campaign visibility and the circumstantial nature of the available evidence.

Recommendations

The following recommendations can help protect against the activity described in this report:

- Train users to identify common social engineering techniques such as leveraging phishing emails, inauthentic WhatsApp messages, and fake websites
- Train users to avoid executing files or commands from untrusted sources, including unknown contacts and profiles in messaging applications
- Enable download protections in browser settings to warn users about potentially harmful websites or downloads
- Consider blocking WhatsApp Web when not necessary for business purposes
- Monitor WebDriver downloads via PS into unusual folder destinations, which may indicate a spambot is present within the environment
- Monitor for folder creation activity following the pattern ChromeAuto_<BROWSER_ID><DATE> within C:\Users\Public\Temp\
- Block WebDriver from executing when it is not needed for business purposes
- Monitor for headless browser instances, which may indicate suspicious hidden browser activity
- Monitor downloads and executions of the WPPConnect/WA-JS library from GitHub; consider blocking access to the URL for this library when not needed for business purposes

Appendix

YARA Rule

The following YARA rule detects the *Astaroth* spambot component described in this report.

```
rule CrowdStrike_Astaroth_Spambot_01 : astaroth spambot
{
    meta:
        copyright = "(c) 2026 CrowdStrike Inc."
        description = "Common patterns for Astaroth spambot"
        version = "202604011037"
        last_modified = "2026-04-01"
        malware_family = "Astaroth"
    strings:
        $c1 = {0f b7 44 50 fe 33 [2] 89}
        $c2 = {80 ea 0a f6 d2 b9 00 00 00 00}
        $c3 = {66 83 e9 41 66 03 d1 66 [3] 66}
        $s1 = "BaixarWAJS"
        $s2 = "ProcessarEnvio"
        $s3 = "Cleanup"
        $s4 = "LoadConfigs"
        $s5 = "ProcessarContatos"
        $s6 = "WebDriverUpdater"
    condition:
        2 of ($c*)
        and 2 of ($s*)
}
```

Falcon LogScale Query

This Falcon LogScale Query detects the copy of the user data browser directory that Astaroth saves to C:\Users\Public\Temp\ChromeAuto_<BROWSER_ID><DATE>.

```
event_platform=Win |
#event_simpleName = DirectoryCreate |
TargetFileName=~/.*\ChromeAuto_[0-5]{1}202/
```

Configuration Snippet

The following is a snippet of *Astaroth*'s configuration code.

```
{
    "ENA":true,
    "FCX":"","
    "LT":0,
    "HL":true,
    "PFX":"m_[RANDN11111/99779]_NUMERO_",
    "DEL":300,
    "ArrMSG":[
        "{saudacao} {nome}!",
        "{saudacao}",
        "{saudacao} {nome}",
    ],
}
```

```

        " {saudacao} {nome} ",
        "{saudacao} {nome}"
    ],
    "ArrMF":[
        "Segue anexo. Fico à disposição.",
        "Arquivo em anexo. Estou à disposição.",
        ... TRUNCATED ...
    ],
    "ArrURL":[
        ... TRUNCATED ...
    ]
}

```

Indicators of Compromise (IOCs)

Malware	Description	Indicator
<i>Astaroth</i> spambot	SHA256 hash of encrypted spambot	c7c62303ee1a37fd7a6e2db9c590ba75c647bc4d22d7dca50cfa8879222ac9e1
	SHA256 hash of encrypted spambot configuration file	ec43a17685e3a555c2eb5f0a2802e9e45d5a2a5d49a0803155acbd74d9ecdbd7
	SHA256 hash of decrypted spambot	d89105c4d567a95f674ed6eac538e32e288b658a4222a3d52e284a77782af4d5
	Installer component C2 servers	stretar7[.]contabilfacil[.]sbs graconxonjal[.]empresaeficiente[.]sbs plansonval[.]impostosrapido[.]top

Vareg spambot	Python version	6168d63fad22a4e5e45547ca6116ef68bb5173e17e25fd1714f7cc1e4f7b41e1
	PS version	a1aa786e02fb9a37a71e0f76b052ab284ba877f2aaa2fb28f05d60487389976a
	C2 server for Python version	https[[:]//varegjoepaks[.]com/api/
	C2 server for PS version	https[[:]//docsmoonstudioclayworks[.]online/arquivoatualizado/gera.php

Table 3. IOCs for Astaroth and Vareg

¹ <https://www.crowdstrike.com/en-us/blog/latin-america-malware-update/>

² [https://developer\[.\]chrome\[.\]com/docs/chromedriver/capabilities](https://developer[.]chrome[.]com/docs/chromedriver/capabilities)

³ [https://github\[.\]com/wppconnect-team](https://github[.]com/wppconnect-team)



CrowdStrike 2026 Global Threat Report

AI threats have reached a critical turning point. Access the definitive look at the cyber threat landscape.

[Download](#)