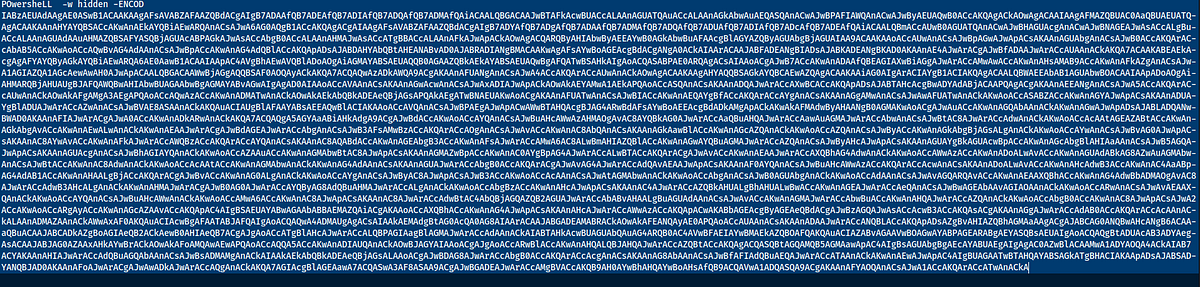
### **Malicious PowerShell Analysis — BTLO**

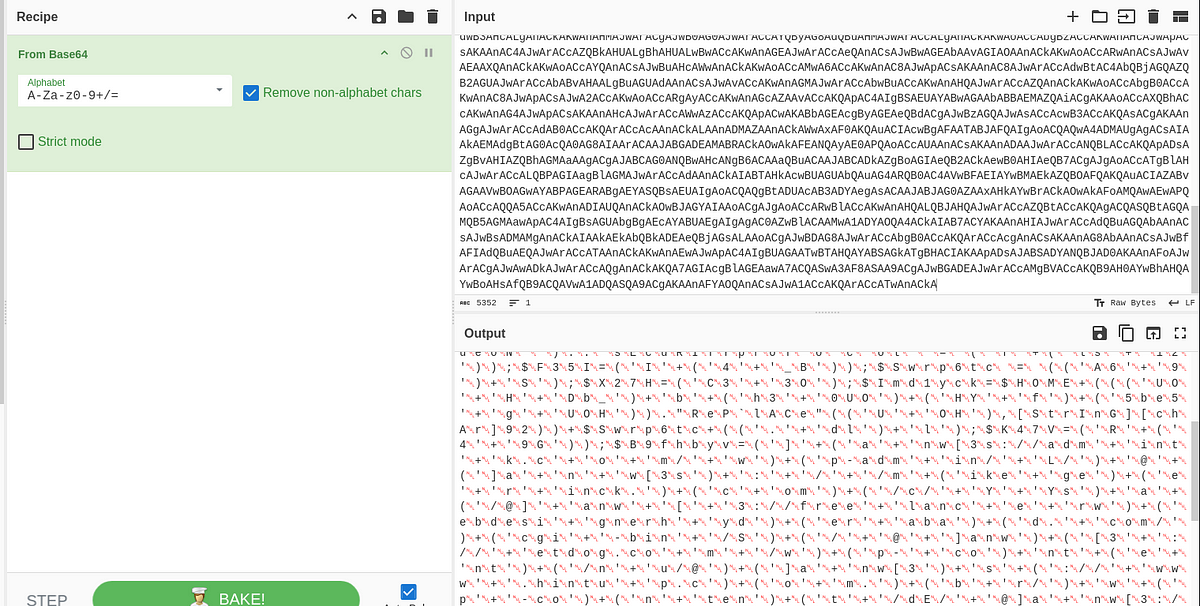
**Scenario:** Recently the networks of a large company named GothamLegend were compromised after an employee opened a phishing email containing malware. The damage caused was critical and resulted in business-wide disruption. GothamLegend had to reach out to a third-party incident response team to assist with the investigation. You are a member of the IR team — all you have is an encoded Powershell script. Can you decode it and identify what malware is responsible for this attack?

File: ps\_script.txt.

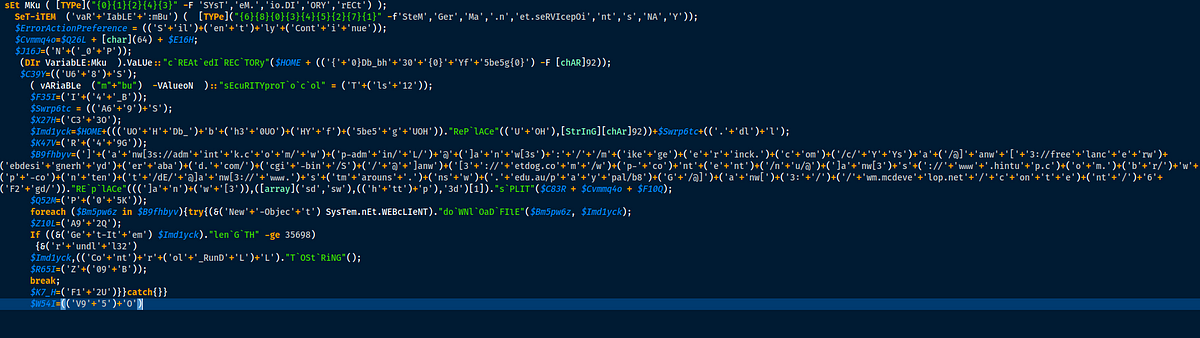
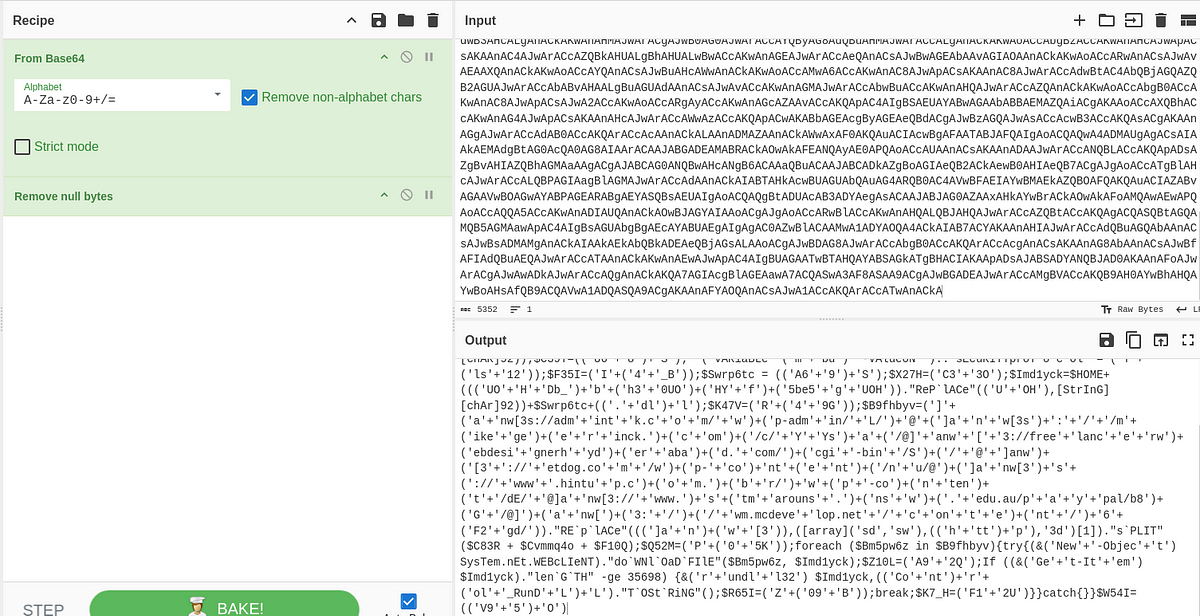
The downloaded challenge file contains encoded text, and a command P0wershell -w hidden, which is a command to run PowerShell in the background.



We should decode the encoded text to find out what exactly does this script do. Since it is not very clear what the encoding is, I had do some trial and error to decode this text. I took the text and checked in CyberChef, it did not immediately recognize the encoding, I tried to decode from base64.



We can see that the decoded text has many null characters, so I used the Remove null characters option on CyberChef.



Although, the above text is still highly obfuscated, we can now make some sense of the text.

Tasks: Now that we are able to make some sense of the data, let us start answering some questions.

1. What security protocol is being used for the communication with a malicious domain?

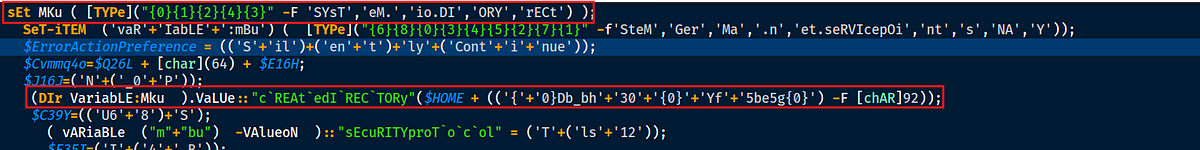
By organizing the code and reading it the security protocol, is clearly evident.



(‘T’+(‘ls’+’12’)) if we combine this we will get tls12 which is a string value of the enum which will translate to Tls1.2

Answer: Tls1.2

2. What directory does the obfuscated PowerShell create? (Starting from \HOME\)

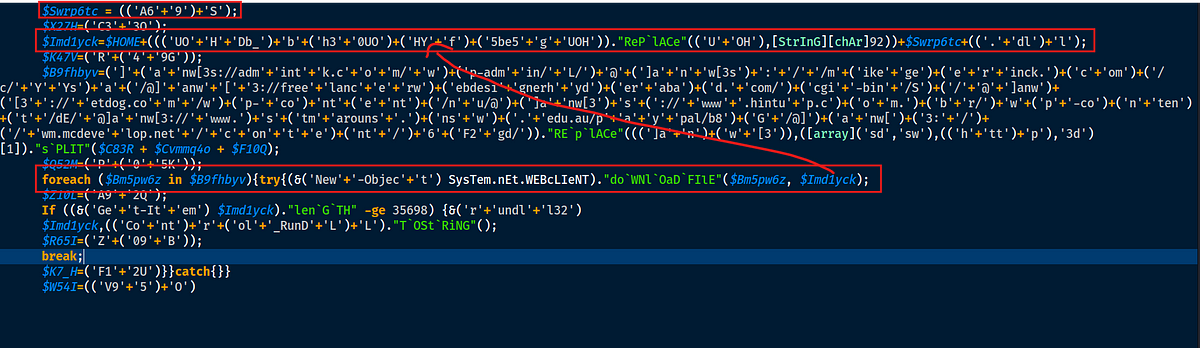


Let us check the first line of the code which sets the variable Mku to System.IO.Directory. Basically the value of sEt MKu ([Type](“{0}{1}{2}{4}{3}” -F ‘SYsT’, ‘eM’, ‘io.DI’, ‘ORY’, ‘eECt’)) translates to MKu = System.IO.Directory. This is called string formatting and the same logic can be applied throughout the code wherever there is a format string.

In the other line marked, it is creating a directory called at \home\Db\_bh30\Yf5be5g.

Answer: \home\Db\_bh30\Yf5be5g

3. What file is being downloaded (full name)?

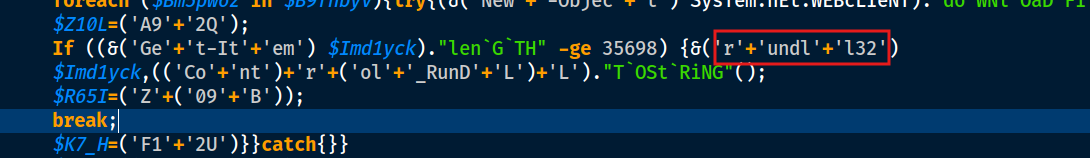


The above code is downloading a file from some kind of path and storing it at the directory in $Imd1yck which is \home\Db\_bh30\Yf5be5g\A69S.dll

Answer: A69S.dll

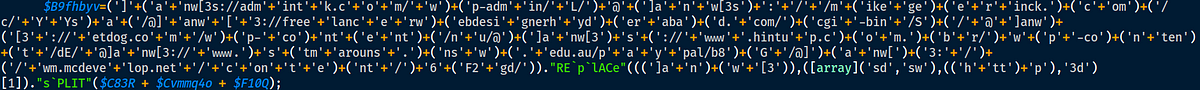
4. What is used to execute the downloaded file?

If we check the code inside the if statement, once the file is successfully downloaded, the system will run a command line utility called rundll32, which is used to run dll files.

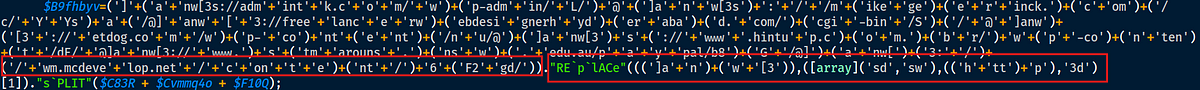


Answer: rundll32

5. What is the domain name of the URI ending in ‘/6F2gd/’



The above code constructs many URLs which are split by an @ symbol. One of those URLs has an ending of /6F2gd/.



Answer: wm.mcdevelop.net

6. Based on the analysis of the obfuscated code, what is the name of the malware?

So let us put it all together

This script, when executed will

1. Run silently in the background
2. creates a directory on the system
3. Enforces a security protocol of Tls1.2
4. Concatenates a string of URL’s
5. Connects to the URL’s and downloads files onto the previously created directory
6. checks if the file is correctly downloaded.
7. Then calls rundll32 to execute the downloaded file.

These characteristics resemble an Emotet malware. More about the malware can be read [here](https://www.cisa.gov/news-events/cybersecurity-advisories/aa20-280a).

Answer: Emotet.

