RARCVE

Kumpel7

August 2024

1 About The Lab

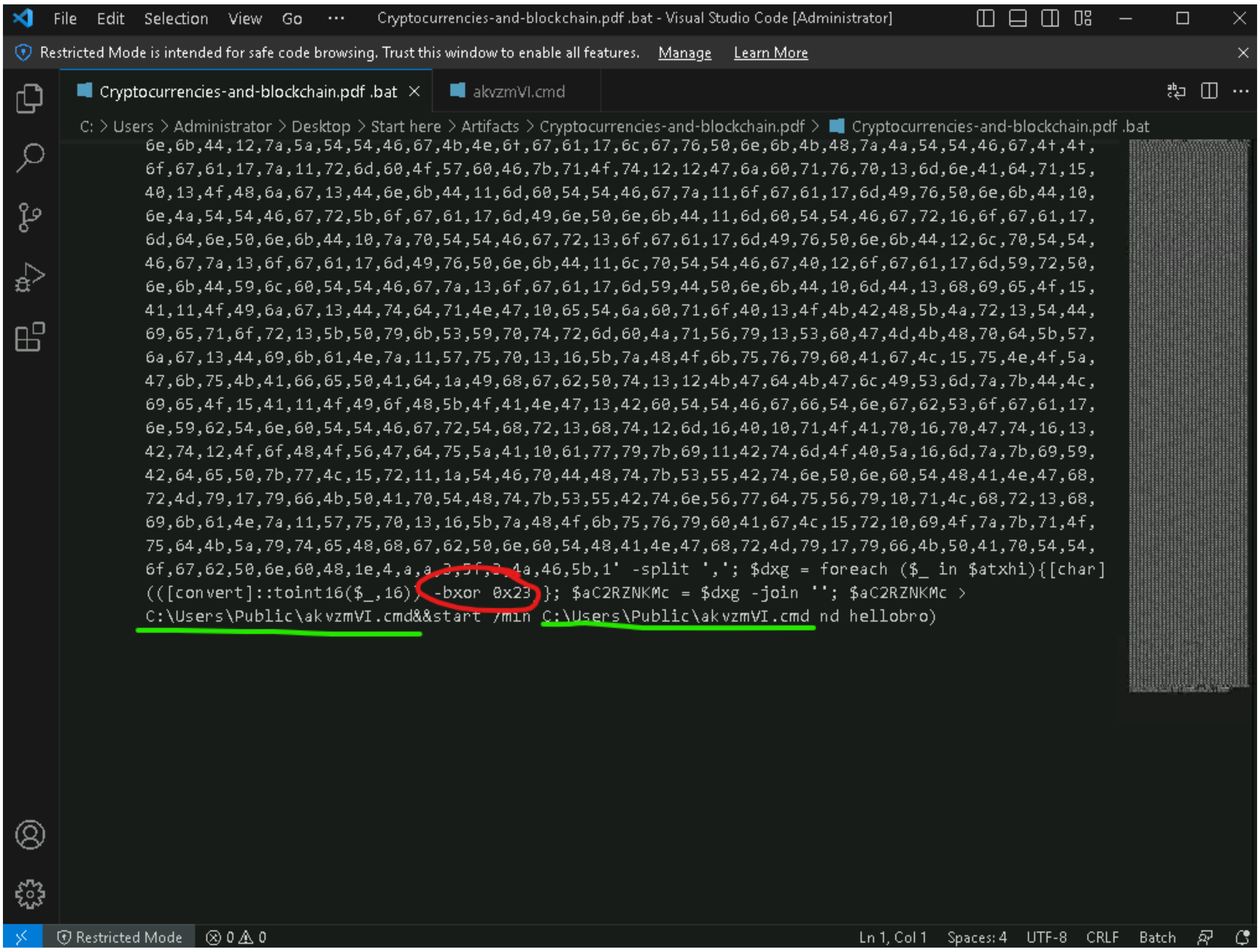
This blueteam CTF challenge from cyberdefenders focuses on malware analysis of zip archive with "pdf" file containing information about cryptocurrencies. However, upon opening the archive, the exploit was lauched. More on that you can read searching for **CVE–2023–38831**.

Screenshot 1: Here are the questions that need to be answered to complete the lab. What I like about the questions from CD is the fact that they help you in real life analysis, because you can ask yourself the same questions :)

To solve those questions we use *cyberchef* for deciphering/decoding and *scdbg* for looking for shellcode.

2 Question 1 and Question 2

To start we go to Artifacts folder and try to unpack the archive. Not all of the contents inside want to be unpacked, but the file inside the folder "Cryptocurrencies-and-blockchain.pdf" is a file "Cryptocurrencies-and-blockchain.pdf.bat". We can open it and analyze what is inside.



Screenshot 2: The contents of *Cryptocurrencies-and-blockchain.pdf.bat*. Most of this code is obfuscated hex text, however we can see the key used to encrypt via XOR (red circle) and an extra file the code is extracted (green line).

3 Question 3, 4 and 5

Next we jump onto the newly created file. It is powershell obfuscated command with base64 encoded "real" command.

Screenshot 3: To decode the real script, we will use cyberchef

Screenshot 4: Here is the decoded script. We can see that the payload is fully obfuscated using RC4 algorithm.

The script goes as follows:

1. The attacker defined deciphering function Xdfwqp
2. Then attacker imports 2 functions from kernel32.dll, VirtualAlloc and CreateThread
3. Next the attacker defines the encrypted payload $KsIajLbCL with key to decipher it $TKCLldzsId
4. Nextly the attacker defines the variable $Yzoic, which is exactly the deciphered payload.
5. Then the payload is executed via imported modules

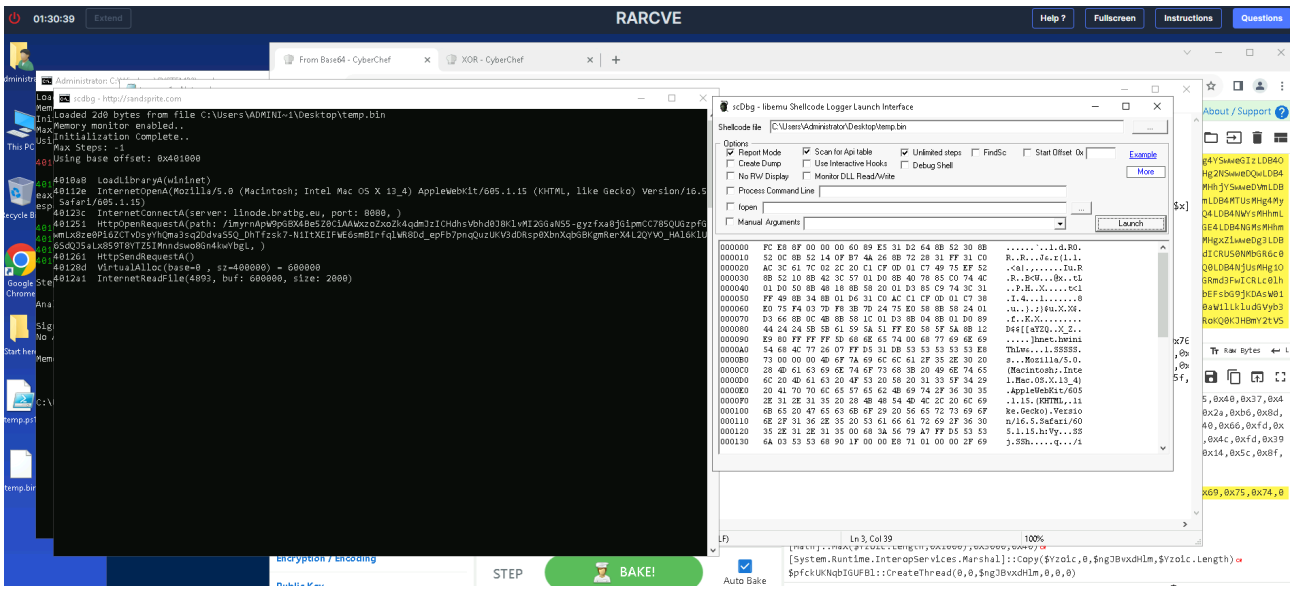
Let's try to analyze what that encrypted payload does.

4 Question 6 and 7

To analyze it we will generate the binary file so we can look for the shell code. After deleting the last 3 lines of the script and copying it to powershell we have in memory defined variable $Yzoic. So we can use the following command to change the output of deciphered payload to binary file:

```
1 [System.IO.File]::WriteAllBytes("C:\Users\Administrator\Desktop\temp.bin",
    $Yzoic)
```

Then we can load this file (in my case it's *temp.bin*) to scDbg. I used the Report Mode, Scan for Api table and Unlimited steps, but this is probably unnecessary. This uncovers for us the answers for the last 2 questions.



Screenshot 5: We see the user agent inside the InternetOpenA function and C2 serwer with port inside InternetConnectA function.

5 Conclusions and thoughts

This lab is labeled as medium and I think it is a fair verdict when it comes to difficulty. It doesn't require a lot of tools to use, but you need to be familiar with simple decoding/deciphering and powershell cmdlets. The hardest part for me was understanding how scDbg works and how to use it on "ps1 script". When it came to me you can generate the bin file, it all started to make sense :)
I enjoyed the lab, and hope you too will.