

FORENSIC

1. Decrypting the ransom : Malicious DOCM Analysis

The screenshot shows a challenge interface with a dark background. At the top, it says 'Challenge' and '26 Solves'. The title 'Decrypting the Ransom: Malicious DOCM Analysis' is prominently displayed, followed by the number '88'. Below the title, a description reads: 'A challenge where the goal was to analyze a malicious DOCM file, extract the encryption key from the ransomware, and decrypt the encrypted data.' There are three buttons: 'File.docm' with a download icon, 'Flag', and 'Submit'.

- Download the file.docm
- Goto linux terminal and type following command.
\$olevba file.docm --decode

```
Type: OpenXML
WARNING For now, VBA stomping cannot be detected for files in memory

VBA MACRO ThisDocument.cls
in file: word/vbaProject.bin - OLE stream: 'VBA/ThisDocument'
-----
(empty macro)

VBA MACRO Module1.bas
in file: word/vbaProject.bin - OLE stream: 'VBA/Module1'
-----
Sub RunPython()
    Dim Ret_Val As Integer
    Dim PythonCommand As String
    Dim CMDCommand As String
    PythonCommand = "python -c ''print('cm9vdE8sb2NhbgHvc3R7bTRjcjBzX3JfZDRuZzNyMHVzfQ==')
    ...
    CMDCommand = "cmd /K " & PythonCommand & " & timeout /T 0.2 & exit"
    Ret_Val = Shell(CMDCommand, vbNormalFocus)
    If Ret_Val = 0 Then
        MsgBox "Couldn't run python script!", vbOKOnly
    End If
End Sub

VBA MACRO UserForm1.frm
in file: word/vbaProject.bin - OLE stream: 'VBA/UserForm1'
-----
(empty macro)

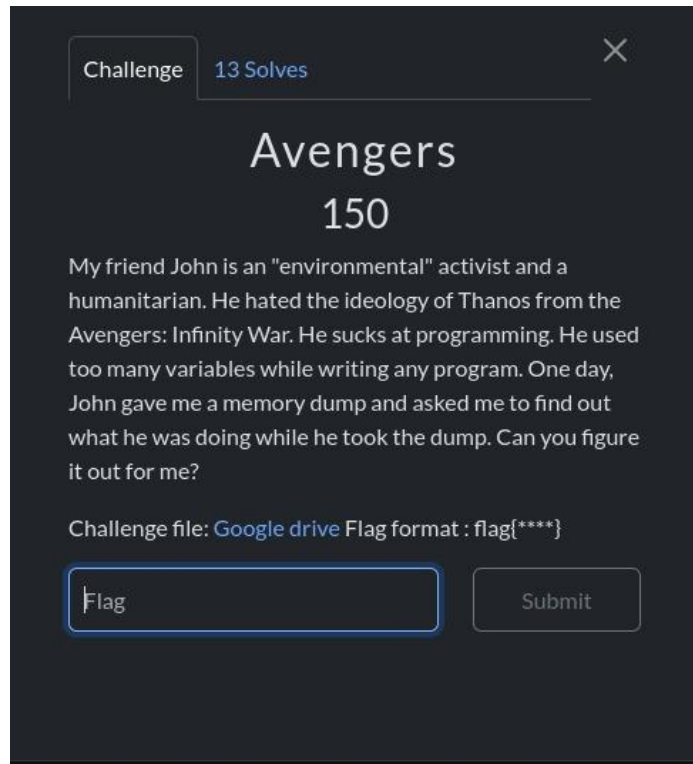
|Type|Keyword|Description|
|-----|-----|-----|
|Suspicious|Shell|May run an executable file or a system command|
|Suspicious|vbNormalFocus|May run an executable file or a system command|
|Suspicious|run|May run an executable file or a system command|
|Suspicious|Hex Strings|Hex-encoded strings were detected, may be used to obfuscate strings (option --decode to see all)|
|Hex String|'\x17\x06'|178FB806|
|Hex String|'\x0a\x07'|CE9F2A92E307|
|Hex String|'\x18.\x10'|182E1085|
|Hex String|'he'|68D8EFC0E345|

HotWater% |
```

Flag is encoded in print statement, decoded using base64.

Flag : root@localhost{m4cr0s_r_d4ng3r0us}

2. Avengers



- Download challenge.raw
- Install volatility package for reading memory dump file.
- Get profile info from volatility command
\$ volatility -f challenge.raw --profile=Win7SP1x86 cmdscan
(it will display command execution demon.py.txt)
- Check console for python output.
- \$ volatility -f challenge.raw --profile=Win7SP1x86 consoles

- Output of demon.py.txt displayed in hexa encoded string

```
$ volatility -f challenge.raw --profile=Win7SP1x86  
Envvars
```

Capture the xor and password.

- Check the output of hashdump

```
$ volatility -f challenge.raw --profile=Win7SP1x86  
Hashdump.
```

We got a hash password, decrypting the other part of the flag

Flag : flag{you_are_good_but1_4m_b3tt3r}