MemLabs - Lab3

3 minutos de leitura

Descrição do desafio

Um script malicioso criptografou uma informação muito secreta que eu tinha no meu sistema. Você pode recuperar as informações para mim, por favor?

***Obs** : Este desafio é composto por apenas 1 bandeira e dividido em 2 partes.*

***Dica** : você precisará da primeira metade da bandeira para obter a segunda.*

Você precisará desta ferramenta adicional para resolver o desafio,

```
$ sudo apt install steghide
```

O formato do sinalizador para este laboratório é: ***inctf{s0me_l33t_Str1ng}***

***Arquivo de desafio** : MemLabs_Lab3 (https://github.com/joathamp/maratona_forense)*

Primeiro precisamos identificar o sistema operacional da imagem da memória.

```
$ volatility -f MemoryDump_Lab3.raw imageinfo
```

```
night-wolf@ubuntu:~/MemLabs/Lab3$ volatility -f MemoryDump_Lab3.raw imageinfo
Volatility Foundation Volatility Framework 2.6.1
INFO    : volatility.debug    : Determining profile based on KDBG search...
          Suggested Profile(s) : Win7SP1x86_23418, Win7SP0x86, Win7SP1x86_24000, Win7SP1x86
                     AS Layer1 : IA32PagedMemoryPae (Kernel AS)
                     AS Layer2 : FileAddressSpace (/home/night-wolf/MemLabs/Lab3/MemoryDump_Lab3.raw)
                      PAE type : PAE
                           DTB : 0x185000L
                          KDBG : 0x82742c68L
          Number of Processors : 1
     Image Type (Service Pack) : 1
                KPCR for CPU 0 : 0x82743d00L
             KUSER_SHARED_DATA : 0xffdf0000L
           Image date and time : 2018-09-30 09:47:54 UTC+0000
     Image local date and time : 2018-09-30 15:17:54 +0530
```

Em seguida, vamos verificar a linha de comando dos processos em execução.

```
$ volatility -f MemoryDump_Lab3.raw --profile Win7SP1x86_23418 cmdline
Volatility Foundation Volatility Framework 2.6.1
........
notepad.exe pid:   3736
Command line : "C:\Windows\system32\NOTEPAD.EXE" C:\Users\hello\Desktop\evilscript.py
**************************************************************************
notepad.exe pid:   3432
Command line : "C:\Windows\system32\NOTEPAD.EXE" C:\Users\hello\Desktop\vip.txt
```

Interessante, temos dois arquivos. `evilscript.py` que, como o nome indica, é maligno e `vip.txt` parece um arquivo importante.

Vamos procurar esses dois arquivos na memória.

```
$ volatility -f MemoryDump_Lab3.raw --profile Win7SP1x86_23418 filescan | egrep
"evilscript.py|vip.txt"
Volatility Foundation Volatility Framework 2.6.1
0x000000003de1b5f0      8      0 R--rw- \Device\HarddiskVolume2\Users\hello\Desktop\evilscript.py.py
.........
0x000000003e727e50      8      0 -W-rw- \Device\HarddiskVolume2\Users\hello\Desktop\vip.txt
```

Agora que temos os deslocamentos dos dois arquivos, vamos despejá-los.

```
$ volatility -f MemoryDump_Lab3.raw --profile Win7SP1x86_23418 dumpfiles -Q 0x000000003de1b5f0 -D
lab3_output/
$ volatility -f MemoryDump_Lab3.raw --profile Win7SP1x86_23418 dumpfiles -Q 0x000000003de1b5f0 -D
lab3_output
```

Aqui está o arquivo python despejado:

```
import sys
import string

def xor(s):
        a = ''.join(chr(ord(i)^3) for i in s)
        return a

def encoder(x):
        return x.encode("base64")

if __name__ == "__main__":
        f = open("C:\\Users\\hello\\Desktop\\vip.txt", "w")
        arr = sys.argv[1]
        arr = encoder(xor(arr))
        f.write(arr)
        f.close()
```

Este script maligno está fazendo XOR no arquivo `vip.txt` com um único caractere e, em seguida, codificando-o em Base64.

E aqui está o conteúdo do arquivo de texto despejado:

```
am1gd2V4M20wXGs3b2U=
```

Portanto, primeiro precisamos decodificá-lo em Base64 e fazer o XOR novamente com o mesmo caractere para recuperar o texto original.

```
$ python
>>> s = 'am1gd2V4M20wXGs3b2U='
>>> s = s.decode('base64')
>>> ''.join(chr(ord(i)^3) for i in s)
inctf{0n3_h4lf
```

Primeiro tempo: inctf{0n3_h4lf

Agora que temos a primeira metade da bandeira, vamos caçar a outra metade.

Este me levou algum tempo, então eu olhei para a dica e diz algo sobre `steghide`.

Steghide é um programa de esteganografia capaz de ocultar dados em imagens e arquivos de áudio e suporta imagens JPEG e BMP, então decidi procurar imagens JPEG na memória.

```
$ volatility -f MemoryDump_Lab3.raw --profile Win7SP1x86_23418 filescan | grep ".jpeg"
Volatility Foundation Volatility Framework 2.6.1
0x0000000004f34148      2      0 RW---- \Device\HarddiskVolume2\Users\hello\Desktop\suspision1.jpeg
```

Você olharia para isso!!!, apenas uma imagem e parece suspeito :)

Então, vamos descartá-lo.

É apenas uma imagem normal, ou é ???

Aí vem `steghide`, essa imagem deve ter algo escondido.

```
$ steghide extract -sf lab3_output/suspision1.jpeg
Enter passphrase:
```

Está pedindo uma senha, a dica diz claramente que: `You'll need the first half of the flag to get the second`.

Vamos tentar a primeira metade do sinalizador como a senha.

```
$ steghide extract -sf lab3_output/suspision1.jpeg
Enter passphrase:
wrote extracted data to "secret text".
```

Voilá!!! vamos pegar esse texto secreto.

```
$ cat secret\ text
_1s_n0t_3n0ugh}
```

Sinalizador: inctf{0n3_h4lf_1s_n0t_3n0ugh}