Tamil CTF 2021 - Analyst

TITLE - Analyst

DESCRIPTION

A developer from Tamil CTF organization is tired of coding, so he tries to smuggle our secet flags as an insider threat to a hacker. Retrieve the flag for us

Here is the Sample

AUTHOR - aidenpearce369

Lets download the file

```
aidenpearce369@amun:~/analyst$ file analyst.zip
analyst.zip: Zip archive data, at least v2.0 to extract
aidenpearce369@amun:~/analyst$ unzip analyst.zip
Archive:  analyst.zip
   creating: analyst/
  inflating: analyst/NTUSER.DAT
  inflating: analyst/software
  inflating: analyst/system
aidenpearce369@amun:~/analyst$ ls
analyst  analyst.zip
aidenpearce369@amun:~/analyst$ cd analyst/
aidenpearce369@amun:~/analyst/analyst$ ls -la
total 83720
drwxrwxr-x 2 aidenpearce369 aidenpearce369     4096 Sep 23 19:56 .
drwxrwxr-x 3 aidenpearce369 aidenpearce369     4096 Sep 28 18:23 ..
-rw-r--r-- 1 aidenpearce369 aidenpearce369  1048576 Sep 23 16:46 NTUSER.DAT
-rw-r--r-- 1 aidenpearce369 aidenpearce369 72876032 Sep 23 16:46 software
-rw-r--r-- 1 aidenpearce369 aidenpearce369 11796480 Sep 23 16:46 system
```

Analysing its file type

```
aidenpearce369@amun:~/analyst/analyst$ file NTUSER.DAT
NTUSER.DAT: MS Windows registry file, NT/2000 or above
aidenpearce369@amun:~/analyst/analyst$ file software
software: MS Windows registry file, NT/2000 or above
aidenpearce369@amun:~/analyst/analyst$ file system
system: MS Windows registry file, NT/2000 or above
```

So, its all registry files

Install `regrip.py` using `pip3 install regrippy`

Analysing NTUSER.DAT,

```
aidenpearce369@amun:~/analyst/analyst$ regrip.py --ntuser NTUSER.DAT
usage: regrip.py [-h] [--system SYSTEM] [--software SOFTWARE] [--sam SAM] [--ntuser NTUSER]
                 [--usrclass USRCLASS] [--root ROOT] [--all-user-hives] [--verbose] [--pipe]
                 [--list]
                 plugin_name
regrip.py: error: the following arguments are required: plugin_name
aidenpearce369@amun:~/analyst/analyst$ regrip.py --ntuser NTUSER.DAT typedurls
http://transfer.sh/4LkNJ4/msg.png
http://tamilctf.com/
http://ctf.tamilctf.com/
http://go.microsoft.com/fwlink/p/?LinkId=255141
```

There is a file transfer link http://transfer.sh/4LkNJ4/msg.png,

Lets see whats in it,

```
aidenpearce369@amun:~/analyst/analyst$ curl http://transfer.sh/4LkNJ4/msg.png -o msg.png
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  286k  100  286k    0     0   175k      0  0:00:01  0:00:01 --:--:--  175k
aidenpearce369@amun:~/analyst/analyst$ file msg.png
msg.png: PNG image data, 1280 x 720, 8-bit/color RGBA, non-interlaced
```

Its an image file, viewing it doesn't give any useful info

Lets find any hidden files in it, using binwalk,

```
aidenpearce369@amun:~/analyst/analyst$ binwalk msg.png

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
0             0x0             PNG image, 1280 x 720, 8-bit/color RGBA, non-interlaced
1221          0x4C5           Zlib compressed data, default compression
246016        0x3C100         Zip archive data, encrypted at least v2.0 to extract, compressed size:
46831, uncompressed size: 46804, name: secret335.zip
292949        0x47855         End of Zip archive, footer length: 22
```

There is a zip file, lets extract it

```
aidenpearce369@amun:~/analyst/analyst$ binwalk -e msg.png

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
0             0x0             PNG image, 1280 x 720, 8-bit/color RGBA, non-interlaced
```

```
1221          0x4C5           Zlib compressed data, default compression
246016        0x3C100         Zip archive data, encrypted at least v2.0 to extract, compressed size:
46831, uncompressed size: 46804, name: secret335.zip
292949        0x47855         End of Zip archive, footer length: 22

aidenpearce369@amun:~/analyst/analyst$ ls
msg.png  _msg.png.extracted  NTUSER.DAT  software  system
aidenpearce369@amun:~/analyst/analyst$ cd _msg.png.extracted/
aidenpearce369@amun:~/analyst/analyst/_msg.png.extracted$ ls
3C100.zip  4C5  4C5.zlib  secret335.zip
```

Lets start with `secret335.zip`,

```
aidenpearce369@amun:~/analyst/analyst/_msg.png.extracted$ file secret335.zip
secret335.zip: empty
aidenpearce369@amun:~/analyst/analyst/_msg.png.extracted$ unzip secret335.zip
Archive:  secret335.zip
  End-of-central-directory signature not found.  Either this file is not
  a zipfile, or it constitutes one disk of a multi-part archive.  In the
  latter case the central directory and zipfile comment will be found on
  the last disk(s) of this archive.
unzip:  cannot find zipfile directory in one of secret335.zip or
        secret335.zip.zip, and cannot find secret335.zip.ZIP, period.
```

Trying other zip,

```
aidenpearce369@amun:~/analyst/analyst/_msg.png.extracted$ unzip 3C100.zip
Archive:  3C100.zip
[3C100.zip] secret335.zip password:
```

Its asking for passwd,

It a series of combination in 0 and 1,

```
aidenpearce369@amun:~/analyst/analyst/_msg.png.extracted$ unzip 3C100.zip
Archive:  3C100.zip
[3C100.zip] secret335.zip password:
  inflating: secret335.zip
```

Passwd : 0

```
aidenpearce369@amun:~/analyst/analyst/_msg.png.extracted$ unzip secret335.zip
Archive:  secret335.zip
[secret335.zip] secret334.zip password:
  inflating: secret334.zip
```

Passwd : 1

So lets write a script and run it,

```
aidenpearce369@amun:~/analyst/analyst/_msg.png.extracted/extract$ cat crack.py
from zipfile import ZipFile
key = []
for x in range(335,0,-1):
    zip_file = 'secret%s.zip'%(x)
    try:
        password = '1'
        with ZipFile(zip_file) as zf:
            zf.extractall(pwd=bytes(password,'utf-8'))
        key.append(password)
    except:
        password = '0'
        with ZipFile(zip_file) as zf:
            zf.extractall(pwd=bytes(password,'utf-8'))
        key.append(password)
data = ("".join(x for x in key))
print(data[::-1],len(data))
aidenpearce369@amun:~/analyst/analyst/_msg.png.extracted/extract$ python3 crack.py
101111101100110010110010100011000010111011111010001001100111001010000110111110101010110011000110100011000
010111010011010001101101000011001110010100001101111101011001110110011000100111010001100101010101000111010
100010010011101111101001001010100100100110001000100010110111100110001000101010110000100011011010010110101
10110100001100010101 335
```

```
aidenpearce369@amun:~/analyst/analyst/_msg.png.extracted/extract$ cat flag.txt
TamilCTF{3veRyth1ng_U_s3E_1s_n0t_TRUE}
```

This is a rabit hole,

The actual flag in in password for the `zip`

Don't forget to add the 0 for first password

The final flag in binary is,

```
101111101100110010110010100011000010111011111010001001100111001010000110111110101010110011000110100011000
010111010011010001101101000011001110010100001101111101011001110110011000100111010001100101010101000111010
100010010011101111101001001010100100100110001000100010110111100110001000101010110000100011011010010110101
101101000011000101010  <-- add here
```

Lets try to decode the binary numbers,

Reversing the order,

```
aidenpearce369@amun:~/analyst/analyst/_msg.png.extracted/extract$ echo
"1011111101100110010110010100011000010111011111101000100110011100101000011011111010101011001100011010001100
0010111010011010001101101000011001110010100001101111101011001110110011000100111010001100101010101000111101
0100010010011101111110100100101010010010011000100010001011011111001100010001010101100001000110110100101101 0
1101101000011000101010" | rev
0101010001100001011011010110100101101100010000110101010001000110011110110100010001000110010010010101001 00
1011111011100100100010101110001010101010011000101110010001100110111001101011111011000010100111001100000101
1011000101100101110100001100010110001100110101010111110110000101001110011001000101111101110100001100010 10
0110100110011011111 01
```

Lets run the solve script,

```
aidenpearce369@amun:~/analyst/analyst$ cat solve.py
binary_int =
int("0101010001100001011011010110100101101100010000110101010001000110011110110100010001000110010010010101
0010010111110111001001000101011100010101010100110001011100100011001101110011010111110110000101001110011 00
0010110110001011001011101000011000101100011001101010101111101100001010011100110010001011111011101000011 00
01010011010011001101111101", 2)
byte_number = binary_int.bit_length() + 7 // 8
binary_array = binary_int.to_bytes(byte_number, "big")
ascii_text = binary_array.decode()
print(ascii_text)
aidenpearce369@amun:~/analyst/analyst$ python3 solve.py
TamilCTF{DFIR_rEqU1r3s_aNalYt1c5_aNd_t1M3}
```

The flag is `TamilCTF{DFIR_rEqU1r3s_aNalYt1c5_aNd_t1M3}`