Cyber Security

HackRushCTF{flag}

FIRST YEAR BTECH- Purple_Plasma Points -170

Points - 170

Arjun Sekar 22110034 (Carbon_Fiber)
Shreyans Jain 22110245(IntenseDrop)

Write up for reverse engineering

The given file is a .ll file.

My approach: I first found out what an .ll file is and found out that its an LLVM IR file. Now I had to read about LLVM just a day prior when looking at gsoc companies and knew that its the bridge between C language and Assembly language. So i thought to convert it into C as Assembly is not really my forte. I started reading about it and found its a herculean task with some possible errors as there are some compressions which can lead to loss of data but it was mentioned in the problem statement that its not reverse engineering really, but what i was doing was proper reverse engineering. So I opened the f.ll file in VS code and found hexadecimal strings. I parsed them using ___ and got the following:

f.ll

C: > Users > asus > Downloads > f.ll

```
; ModuleID = 'f.c'
source_filename = "f.c"
target datalayout = "e-m:e-p270:32:32-p271:32:32-p272:64:64-i64:64-f80:128-n8:16:32:64-S128"
target triple = "x86_64-pc-linux-gnu"

@__const.main.hex_str = private unnamed_addr constant [43 x i8] c"5468697320697320612074657374207374726967\00", align 16
@.str = private unnamed_addr constant [55 x i8] c"4861636b527573684354467b315f6834355f363176336e5f75707d\00", align 1
@.str.1 = private unnamed_addr constant [28 x i8] c"The ciphered string is: %s\0A\00", align 1
@.str.2 = private unnamed_addr constant [5 x i8] c"%02x\00", align 1

```

Fig1 . Shows two strings which are encoded in hex

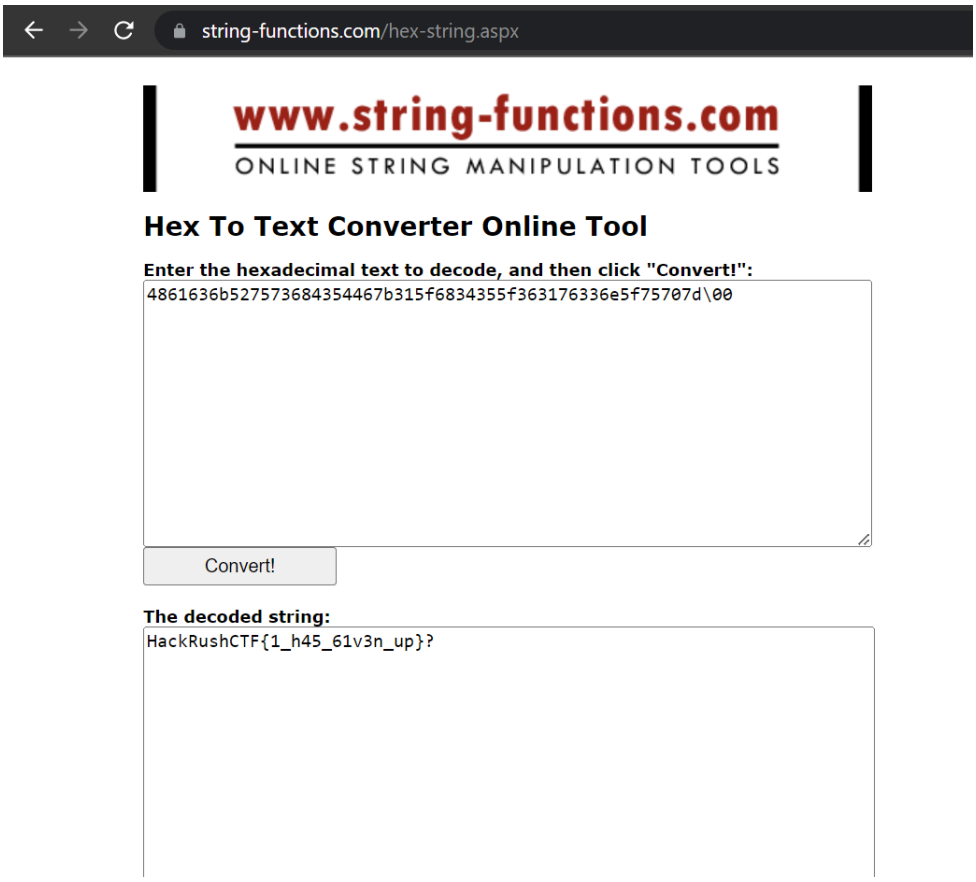
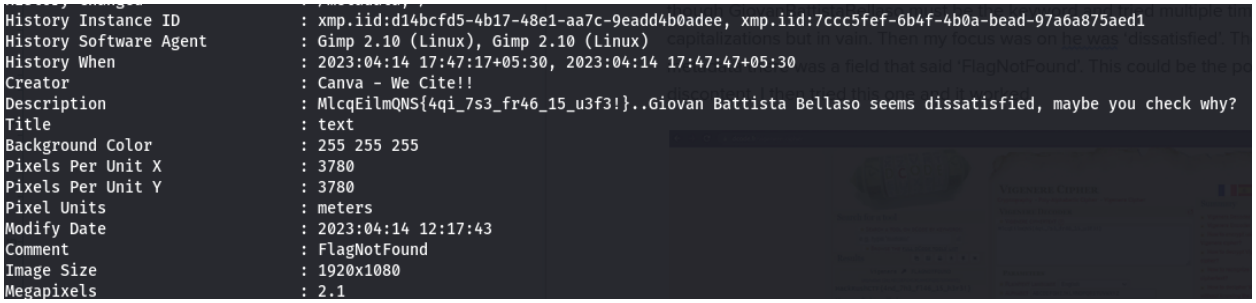


Fig2. Using a tool to convert the hex to a string

Images can be Fun

Here we have been given an image in png format. This is a classic stenography question. I know about these types of questions from the infamous ‘Cicada 3301’. My first thought was to examine the image metadata and I used exiftool for that. This was the metadata present.v



In this we see there is an encoded string which could possibly be the flag.

My next thought was to find out who or what Giovan Battista Bellaso was. I found he was a mathematician who made the first unbreakable cipher which is based on the keyword. I then thought GiovanBattistaBellaso must be the keyword and tried multiple times with various different capitalizations but in vain. Then my focus was on he was ‘dissatisfied’. Then by reading the other metadata there was a field that said ‘FlagNotFound’. This could be the possible reason for his discontent. I then tried this one and it worked.

Plot

1. Beginning: what happened at the beginning?
2. Middle: usually the highest point of action in the story.
3. Ending: how did everything finally work out?

Reading Rainbow Tip: Think about the most important events in the story. Be careful not to re-tell the whole story but give enough detail so that the plot makes sense to someone who hasn't read the book.

EasyAF

72 97 99 107 82 117 115 104 67 84 70 123 49 95 114 52 110 95 48 117 55 95 48 102 95 49 100 51 52 53 95 98 121 95 55 104 49 53 95 112 48 49 110 55 125

Approach:

Simple Decimal to ASCII character conversion

Python code for the same:

```
numbers = [72, 97, 99, 107, 82, 117, 115, 104, 67, 84, 70, 123, 49, 95, 114, 52, 110, 95, 48, 117, 55,
    95, 48, 102, 95, 49, 100, 51, 52, 53, 95, 98, 121, 95, 55, 104, 49, 53, 95, 112, 48, 49, 110, 55,
    125]

flag = ''.join([chr(num) for num in numbers])
print(flag)
```

This program converts ASCII Decimals to Characters

OUTPUT:

```
HackRushCTF{1_r4n_0u7_0f_1d345_by_7h15_p01n7}
>
```

So Many Directories !!!

Problem Statement:

You have been given a folder with multiple sub folders which individually contains numerous folders. Your task is to find the flag which is hidden inside one of the sub folders.

Approach:

I initially thought of checking each folder manually using TAB key which can be used to move to forward directories but then I realized that it is actually a waste of time because this method cannot be used if there are multiple directories inside multiple directories. After manually checking a single directory I found that local directory contains a file **random.txt** , so I wrote a python code to search inside multiple directories and find the local directory which contains something other than **random.txt .**

Python Code:

```
import os

def search_directories(path):
    for root, dirs, files in os.walk(path):
        for file in files:
            if file != "random.txt":
                print(f"Directory with non-random file: {root}")
                break

search_directories(".")
```

OUTPUT:

```
┌──(assassin㉿AssassinKali)-[~/Downloads/SoManyDirectories]
└─$ python check.py
Directory containing non-random file: .
Directory containing non-random file: ./bgivkupt/yksablda/pijwykhk/ybvpimsz/ciwumroq/mqskydfn/lskxdgtw/yu
zkjjqn/tozrdxxp/khgwzknc/vumluirt/synadmxk/khrjbohn/nvwnyzet/gewzqgyo/ixhitseo/tpprhjzh/lasohlqt/lskbyqyf
```

This gave me the the location of local directory which contains a file which is not **random.txt.**

Checking that directory

```
┌──(assassin㉿AssassinKali)-[~/Downloads/SoManyDirectories]
└─$ cd ./bgivkupt/yksablda/pijwykhk/ybvpimsz/ciwumroq/mqskydfn/lskxdgtw/yuzkjjqn/tozrdxxp/khgwzknc/vumlui
rt/synadmxk/khrjbohn/nvwnyzet/gewzqgyo/ixhitseo/tpprhjzh/lasohlqt/lskbyqyf/

┌──(assassin㉿AssassinKali)-[~/…/ixhitseo/tpprhjzh/lasohlqt/lskbyqyf]
└─$ ls
flag.txt
```

Finally I found the local directory that contains **flag.txt**

Flag.txt:

```
┌──(assassin㉿AssassinKali)-[~/…/ixhitseo/tpprhjzh/lasohlqt/lskbyqyf]
└─$ cat flag.txt
HackRushCTF{00p5!y0u_f0und_m3!}
```