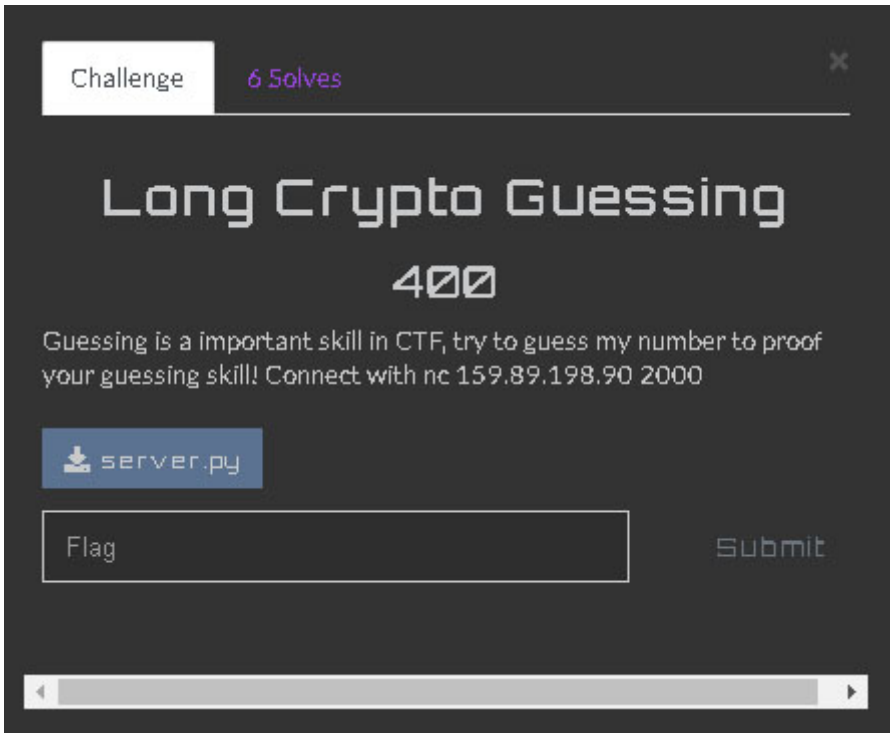
Writeup for Wargames.my 2020

Writeup for Wargames.my 2020

Category: Cryptography - Long Crypto Guessing



In the question, we are given nc `address` and the `python source code` for the server.

```
#!/usr/bin/env python3
from random import getrandbits
import sys
flag = open("flag.txt","r").read()
class PRNG:
 a = getrandbits(64)
 b = getrandbits(64)
 p = 11760071327054544317
def __init__(self, seed):
  self.state = seed
def next(self):
  self.state = (self.a * self.state + self.b) % self.p
  return self.state
print("Guessing is a important skill in CTF, try to guess my number!")
print("I give you first 3 values of my number,")
print("but you need to guess correctly for next 1000 times in a row!!")
print("If you're lucky enough, you can get the flag as reward!\n")
```

```
gen = PRNG(getrandbits(64))
print(f"First 3 values: {gen.next()},{gen.next()},{gen.next()}\n")
for i in range(1000):
 try:
  guess = int(input("Enter a number between 0-9999: "))
 except:
  print("HACKER ALERT! Aborting..")
  sys.exit()
 num = gen.next() % 10000
 if guess == num:
  print("Incredible! Next round!")
 else:
  print("Sorry! Better luck next time..")
  sys.exit()
print(f"Well done!! Good guessing! Flag: {flag}")
```

Reading through the code, the code give **first 3 random number** to the user then the user need to **enter (guess/predict)** the next number(remainder of divide by 10000) that the server generated correctly for **1000 times** consecutively.

In PRNG(pseudo random generator) class it generate a random `a`, `b` and given `p` value, also it has `state` value initialize through object creation. After that function `next` is to create calculate the next number using the number before. Therefore, the function is a type of linear function. This specific generator are called Linear Congruential Generator(LCG)

```
gen = PRNG(getrandbits(64))
print(f"First 3 values: {gen.next()},{gen.next()},{gen.next()}\n")
```

In here, the server will **generate random number** that will become the **seeds** of the PRNG object, then it will generate next **3 number** and print to users Since it is type of linear function.

$$X_{n+1} = (a * X_n + b) \bmod p$$

Therefore **seeds** is `state` or first term of the equation, we are gonna call it T:

$$X_1 = T$$

Then it will generate next 3 term:

$$X_2 = (a * X_1 + b) \bmod p$$

$$X_3 = (a * X_2 + b) \bmod p$$

$$X_4 = (a * X_3 + b) \bmod p$$

In LCG it has 3 integer:

- Multiplier `a`

- Increment `b`
- Modulus `p`

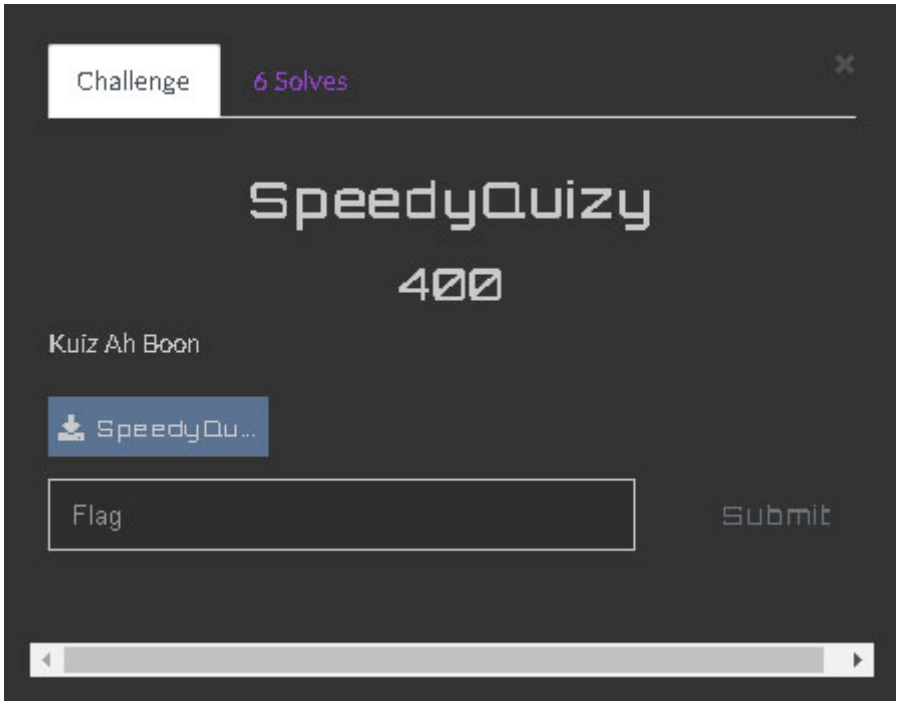
`p` is given in the source code `11760071327054544317`, So we need to find `a` and `b` to predict/solve the next number. Since we have **2 equation** and 2 unknown `a` and `b`, we can solve the equation. Here are detail explaination of finding the missing `Multiplier` and `Increment`, also contains different ways to crack LCG. Once we find out the `a` and `b`, we can find out next number using script below.

```
from pwn import *
import math

def prng(s, a, b, p):
    return (a*s + b) % p

def egcd(a, b):
    """Returns a triple (g, x, y), such that ax + by = g = gcd(a,b).
       Assumes a, b >= 0, and that at least one of them is > 0.
       Bounds on output values: |x|, |y| <= max(a, b)."""
    if a == 0:
        return (b, 0, 1)
    else:
        g, y, x = egcd(b % a, a)
        return (g, x - (b // a) * y, y)

def modinv(a, m):
    g, x, y = egcd(a, m)
    if g != 1:
        return None
    else:
        return x % m

def crack_unknown_increment(states, modulus, multiplier):
    increment = (states[1] - states[0]*multiplier) % modulus
    print("Increment", increment)#
    return modulus, multiplier, increment

def crack_unknown_multiplier(states, modulus):
    mod = modinv(states[1] - states[0], modulus)
    multiplier = (states[2] - states[1]) * mod % modulus
    print("Multiplier",multiplier)#
    return crack_unknown_increment(states, modulus, multiplier)
```

```
r = remote("159.89.198.90", 2000)
p = 11760071327054544317
t = r.recvuntil("values: ")
t = r.recvline().decode('UTF-8').rstrip()
states = list(map(int, t.split(',')))
currentState = states[2]
p, a, b = crack_unknown_multiplier(states, p)
for i in range(1000):
    t = r.recvuntil("0-9999: ")
    currentState = prng(currentState, a, b, p)
    print(currentState)
    r.sendline(str(currentState%10000))
t = r.recvuntil("}").decode('UTF-8').rstrip()
print(t)
```

By implement some of his code to find `a` and `b` then we write the script to answer 1000 consecutive number correctly until it return a flag.

Category: Mobile - SpeedyQuizzy



Given APK file `SpeedyQuizy.apk`

Using [online apk decompiler], we can decompile the file.

Inside source code `StartQuiz.java`, below some snippet of the code

```
    public void onCreate(Bundle bundle) {
        super.onCreate(bundle);
        setContentView((int) C0272R.layout.activity_start_quiz);
        String stringExtra =
getIntent().getStringExtra(MainActivity.EXTRA_MESSAGE);
        this.answerText = (EditText)
findViewById(C0272R.C0274id.answerText);
        this.answerSubmit = (Button)
findViewById(C0272R.C0274id.answerSubmit);
        ((TextView)
findViewById(C0272R.C0274id.textView)).setText(stringExtra);
        this.SERVER_IP = "www2.wargames.my";
        this.SERVER_PORT = "8080";
        Thread thread = new Thread(new Thread1());
        this.Thread1 = thread;
        thread.start();
        this.answerSubmit.setOnClickListener(new View.OnClickListener() {
            public void onClick(View view) {
                String trim =
StartQuiz.this.answerText.getText().toString().trim();
                if (!trim.isEmpty()) {
                    new Thread(new Thread3(trim)).start();
                }
```

```
            }
        });
    }
    class Thread1 implements Runnable {
        Thread1() {
        }

        public void run() {
            try {
                final TextView textView = (TextView)
StartQuiz.this.findViewById(C0272R.C0274id.textView);
                Socket socket = new Socket(StartQuiz.this.SERVER_IP,
Integer.parseInt(StartQuiz.this.SERVER_PORT));
                PrintWriter unused = StartQuiz.this.output = new
PrintWriter(socket.getOutputStream());
                BufferedReader unused2 = StartQuiz.this.input = new
BufferedReader(new InputStreamReader(socket.getInputStream()));
                StartQuiz.this.runOnUiThread(new Runnable() {
                    public void run() {
                        textView.append("");
                    }
                });
                new Thread(new Thread2()).start();
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
    }
```

We know that it used `socket` to connect to the server, therefore we can just use `nc` in terminal to the server `www2.wargames.my` with port `8080`. In terminal we `nc www2.wargames.my 8080`

```
[2020-12-06 11:34:24pm] You are to answer 3 question in 4 seconds.
Any incorrect attempt will require you to start again.
If not sure, just answer in small letter.

Type 'ok' to proceed, or 'quit' to end.
```

Typing `ok` will lead to question

```
[2020-12-06 11:34:55pm] Question No 1
> I am not sure what does PuTTY means. Do you know what is TTY?
```

But the question are **randomly** select from a **set of question**. Therefore we need to **fetch** some of the question so we can create script to answer it.

Using script below we can fetch some of the question. Credits to [H0j3n]

```
from pwn import *

Question = []

for i in range(10):
    r = remote("www2.wargames.my", 8080)
    r.recv("1024")
    r.send("ok\n")
    r.recvuntil(">")
    Question.append(r.recvline().decode('UTF-8').rstrip())
    r.send("1\n")

for i in Question:
    print(i)
```

Below is list of some of the question

```
 DNS zone transfer occurs on port 53. (Of course you know that). But, it
is TCP or UDP?
 DNS zone transfer occurs on port 53. (Of course you know that). But, it
is TCP or UDP?
 I am not sure what does PuTTY means. Do you know what is TTY?
 I am not sure what does PuTTY means. Do you know what is TTY?
 Shifted by 13, and we got this pvephvg
 Shifted by 13, and we got this nnyvz
 Divide 67012 with 14286. Round to the nearest whole number.
 Reverse of retupmoc is ...
 Multiply 55583 and 67056.
 Divide 86517 with 4460. Round to the nearest whole number.
 Given 90707 - 38282 = x and y=2+x. Find y.
 Shifted by 13, and we got this nnyvz
 After applying a monoalphabetic cipher, the string become gvhg
 Reverse of tae is ...
 Shifted by 13, and we got this nnyvz
 Reverse of tae is ...
 Can you add 88279 to 14864?
 Given 50562 - 39612 = x and y=2+x. Find y.
 I am not sure what does PuTTY means. Do you know what is TTY?
 Shifted by 13, and we got this jngre
```

```
 DNS zone transfer occurs on port 53. (Of course you know that). But, it
is TCP or UDP?
 Divide 81017 with 34149. Round to the nearest whole number.
 Multiply 92204 and 78340.
 Reverse of retupmoc is ...
 Can you add 47558 to 86954?
 DNS zone transfer occurs on port 53. (Of course you know that). But, it
is TCP or UDP?
 Can you add 1592 to 61311?
```

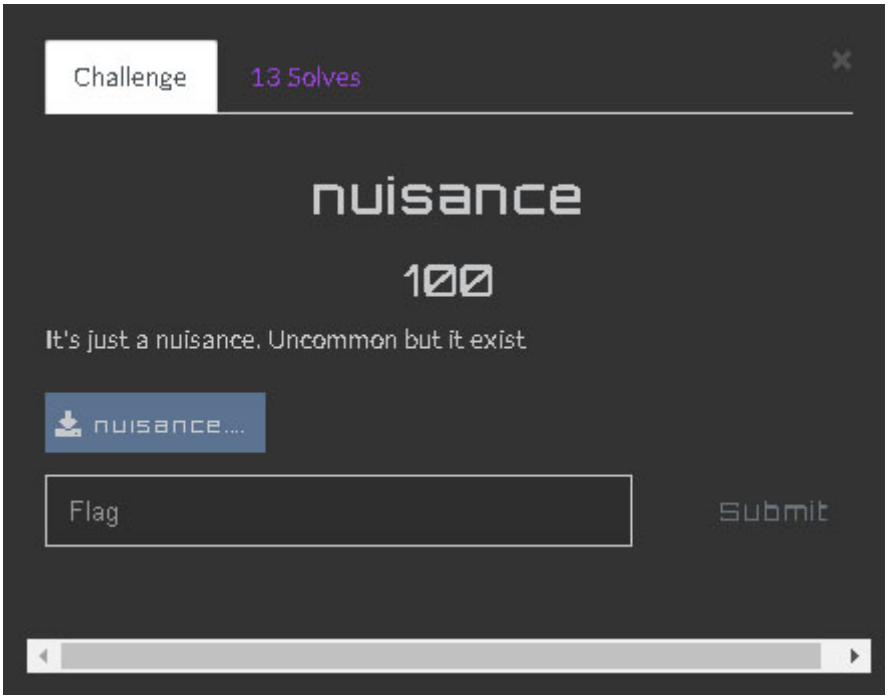
Script below will automatically answer the question:

```
from pwn import *

def rot13(phrase):
   abc = "abcdefghijklmnopqrstuvwxyz"
   out_phrase = ""
   for char in phrase:
       out_phrase += abc[(abc.find(char)+13)%26]
   return out_phrase

def atbash(text):
      N = ord('z') + ord('a')
      ans=''
      return ans.join([chr(N - ord(s)) for s in text])

def question(q):
    ans = ""
    if q.startswith("Reverse"):
        p = q.split()
        ans = p[2][::-1]
    elif q.startswith("DNS"):
        ans = "TCP"
    elif q.startswith("I am not"):
        ans = "teletype"
    elif q.startswith("Shifted"):
        p = q.split()
        ans = rot13(p[7])
    elif q.startswith("Divide"):
        p = q.split()
        ans = str(round(float(p[1])/float(p[3].rstrip('.'))))
    elif q.startswith("Multiply"):
        p = q.split()
```

```
        ans = str(int(p[1])*int(p[3].rstrip('.')))
    elif q.startswith("Given"):
        p = q.split()
        ans = str(2 + int(p[1]) - int(p[3]))
    elif q.startswith("Can"):
        p = q.split()
        ans = str(int(p[3]) + int(p[5].rstrip('?')))
    elif q.startswith("Biggest"):
        ans = "65535"
    elif q.startswith("After"):
        p = q.split()
        ans = atbash(p[8])
    return ans

r = remote("www2.wargames.my", 8080)

t = r.recvuntil("end.")
r.sendline("ok")

for x in range(3):
    t = r.recvuntil("> ")
    print(t)
    t = r.recvline().decode('UTF-8').rstrip()
    print(t)
    ans = question(t)
    print(ans)
    r.sendline(ans)
t = r.recvuntil("}")
print(t)
```

```
kaitorque-kali@kaitorque-kali:~/Desktop/Wargames2020$ python3 speed.py
[+] Opening connection to www2.wargames.my on port 8080: Done
b'\n\n\n[2020-12-05 11:09:26pm] Question No 1\n> '
Given 18985 - 8278 = x and y=2+x. Find y.
10709
b'\n\n[2020-12-05 11:09:27pm] You answered 10709 for question no 1\nCORRECT!\n\n\n[2020-12-05 11:09:27pm] Question
 No 2\n> '
I am not sure what does PuTTY means. Do you know what is TTY?
teletype
b'\n\n[2020-12-05 11:09:28pm] You answered teletype for question no 2\nCORRECT!\n\n\n[2020-12-05 11:09:28pm] Quest
ion No 3\n> '
Shifted by 13, and we got this nnyvz
aalim
b'.\n\n\n[2020-12-05 11:09:29pm] You answered aalim for question no 3\nCORRECT!\n\nGreat! You solved within the ti
me limit. The flag is wgmy{418b3ea849ff3b93def86cfbc90440c1}'
kaitorque-kali@kaitorque-kali:~/Desktop/Wargames2020$
```

Category: Steganography - Nuisance



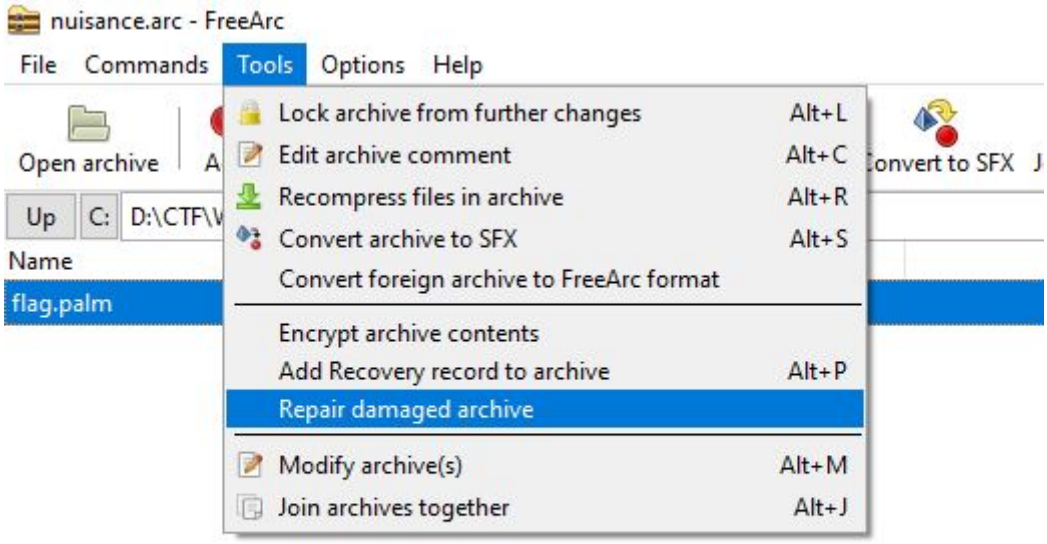
Given file `nuisance.arc`

Open file using HxD to look for signature file, Quick Google of `.arc` file type signature found `41 72 43 01` which is a **FreeArc** compress file.

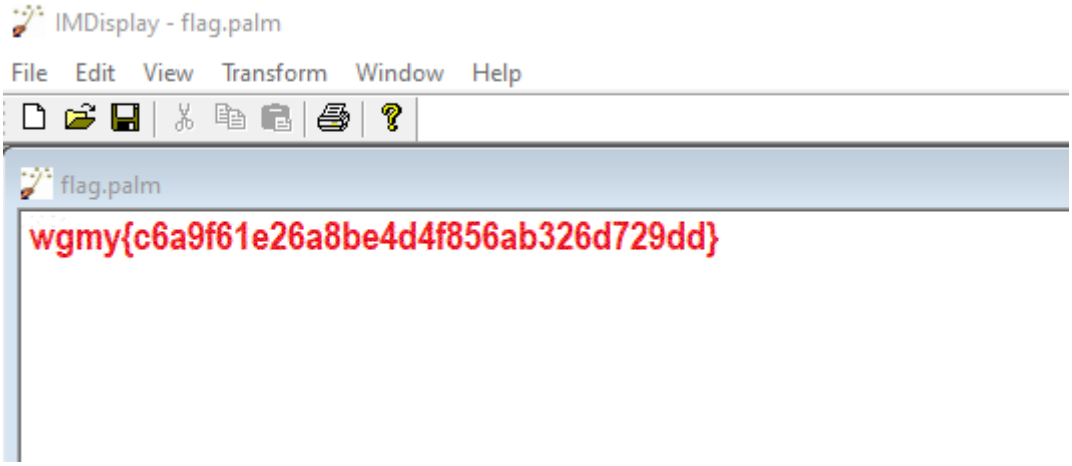
nuisance.arc

Offset (h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Decoded text
00000000	43	41	72	01	00	00	06	07	41	72	43	01	02	73	74	6F	CAr.....ArC..sto
00000010	72	69	6E	67	00	10	10	4F	BF	70	87	67	0B	7A	C5	30	ring...O¿p‡g.zÅ0
00000020	25	00	00	09	0E	20	00	3C	15	00	00	ED	08	00	00	23	%.... .<...í...#
00000030	08	00	00	00	00	00	00	00	00	00	00	00	FF	BC	00	16	..........ÿ¼..
00000040	AB	75	EC	61	BA	9E	46	2B	8D	1A	BE	BD	EC	32	4A	C2	«uìaºžF+..¾½ì2JÂ
00000050	36	62	6A	2F	2E	4F	0D	DA	D5	D1	3D	A2	A6	3A	19	57	6bj/.O.ÚÕÑ=¢¦:.W
00000060	E3	00	05	11	C3	2E	95	2F	7A	A9	A1	BF	81	42	0D	BE	ã...Ã.•/z©¡¿.B.¾
00000070	63	36	21	F3	8D	FB	31	86	63	3E	F7	03	46	CD	52	AF	c6!ó.û1†c>÷.FÍR¯
00000080	39	4D	5D	BD	78	BD	33	9A	7F	E3	06	C9	E2	67	89	E1	9M]½x½3š.ã.Éâg‰á
00000090	58	28	CB	E3	2C	E0	23	DA	A1	E7	91	CB	25	02	84	49	X(Ëã,à#Ú¡ç‘Ë%.„I

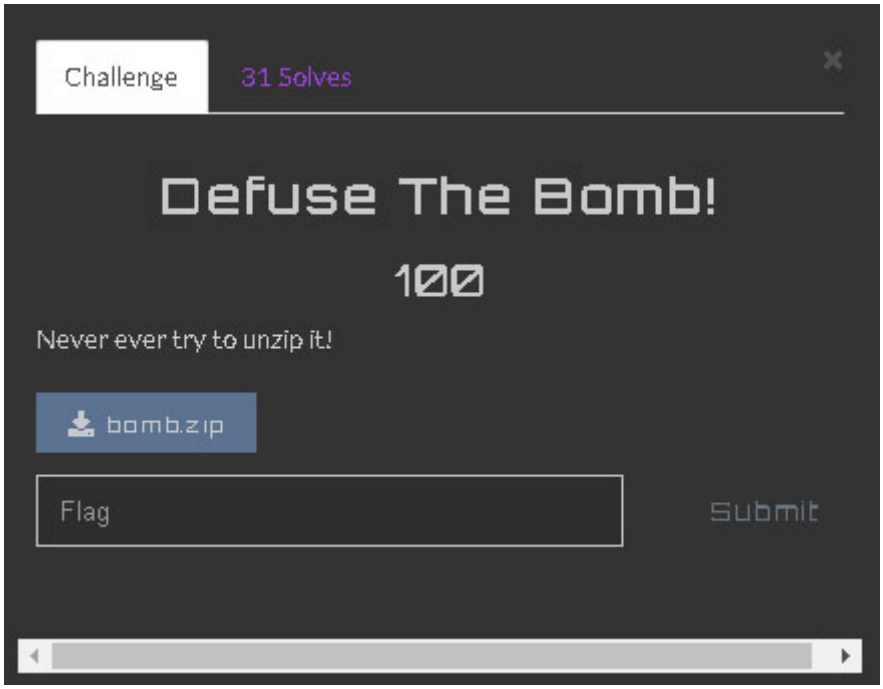
Download FreeArc software, In the software we cannot extract the file due to it is **corrupted**. But the software has the ability to **repair** the damaged achive.



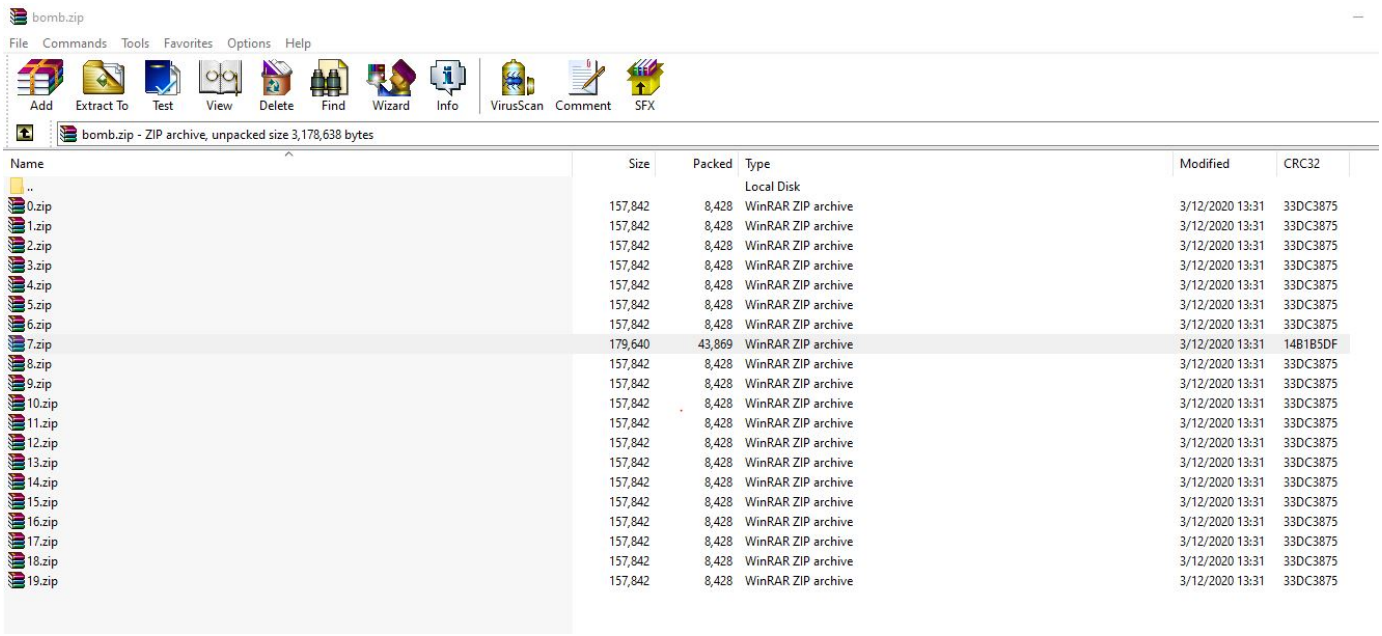
After repairing and extract, we still cannot open `flag.palm` file. So, we did a quick google on `.palm` file and found some information and it says that the file is an **image** file of `Palm OS Bitmap Image` that can be open with application **ImageDisk**. Download ImageDisk and open the file using it:



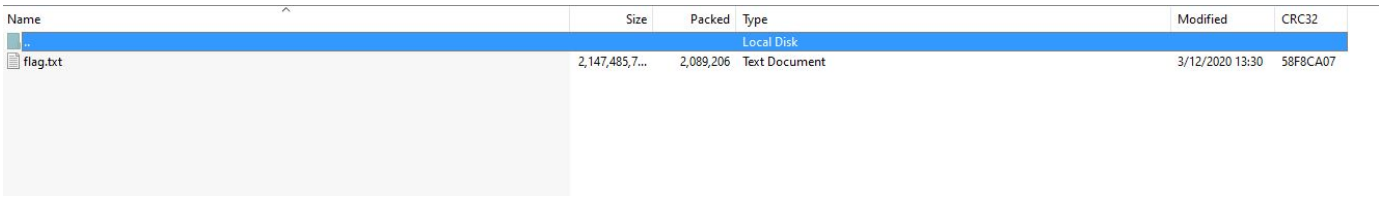
Category: Miscellaneous - Defuse The Bomb!



Given file `bomb.zip`. The file is a Zip bomb which contain file that have multiple time been compress. Open it using `Winrar`



We can see that **one** of the zip is contains file `size` and `CRC32` that is diffrent, so we just click it to reveal another of the same situation. so we just click all zip file that is different until it reach to flag.txt



The we just extract the `flag.txt` file, but the file is `2GB`. So cannot open with any normal text editor, but HxD can open it, then scroll down to find the flag.

```
800007A0  77 77 77 77 77 77 77 77 77 77 77 77 77 77 77 77  wwwwwwwwwwwwwwww
800007B0  77 77 77 77 77 77 77 77 77 77 77 77 77 77 77 77  wwwwwwwwwwwwwwww
800007C0  77 77 77 77 77 77 77 77 77 77 77 77 77 77 77 77  wwwwwwwwwwwwwwww
800007D0  77 77 77 77 77 77 77 77 77 77 77 77 77 77 77 77  wwwwwwwwwwwwwwww
800007E0  77 77 77 77 77 77 77 77 77 77 77 77 77 77 77 77  wwwwwwwwwwwwwwww
800007F0  77 77 77 77 77 77 77 77 77 77 77 77 77 77 77 0A  wwwwwwwwwwwwwww.
80000800  77 67 6D 79 7B 30 34 61 32 37 36 36 65 37 32 66  wgmy{04a2766e72f
80000810  30 65 32 36 37 65 64 35 38 37 39 32 63 63 31 35  0e267ed58792cc15
80000820  37 39 37 39 31 7D                               79791}
```