WRITEUP FINAL ICYPTION 2020

Semoga Menang
SMKN 2 Surakarta

Category General

1. love on the weekend (50 pts)

Diberikan sebuah file mp3 yang bernama “love on the weekend.mp3” maka kemudian saya cek file tersebut menggunakan command “file” pada linux.

```
root@kali:~/icyption# file love\ on\ the\ weekend.mp3
love on the weekend.mp3: Audio file with ID3 version 2.4.0, contains:MPEG ADTS, layer III, v1, 320 kbps, 48 kHz, JntStereo
```

Lalu saya coba putar file tersebut hanya terdengar lagu dari John Mayer yaitu Love on the Weekend, maka saya mencoba cara lain yaitu dengan melihat metadata dari file tersebut menggunakan exiftool.

```
root@kali:~/icyption# exiftool love\ on\ the\ weekend.mp3
[REDACTED]
Lyrics                                  : It's a Friday, we finally made it.I can't believe I get to see your face.You've been working and I've been waiting.To pick you up and take you from this place.Love on the weekend, love on the weekend.Like only we can, like only we can.Love on the weekend, love on the weekend.I'm coming up and I'm loving every minute of it.You be the DJ, I'll be the driver.You put your feet up in the getaway car.I'm flying fast like a, a wanted man.I want you, baby, like you can't understand.Love on the weekend, love on the weekend.We found a message in a bottle we were drinking.Love on the weekend, love on the weekend.I hate your guts 'cause I'm loving every minute of it.Oh oh oh oh-oh.I gotta leave ya, it's gonna hurt me.My clothes are dirty and my friends are getting worried.Down there below us, under the clouds.Baby, take my hand and pull me down, down, down, down.And I'll be dreamin' of the next time we can go.Into another serotonin overflow.Love on the weekend, love on the weekend.I'm busted up but I'm loving every minute of it.Love on the weekend.Love on the weekend icyption{Do_y0u_l1k3_J0hn_May3r}.I'm looking for a little love I'm looking for a little love, oh yeah.Love on the weekend.Love on the weekend.Love on the weekend.
[REDACTED]
```

Maka akan terlihat metadata lyrics yang berisi flag dari soal ini.

Flag: icyption{Do_y0u_l1k3_J0hn_May3r}

2. bitcoin make you rich (75pts)

Pada deskripsi soal terdapat text yang diencode, dan diketahui menggunakan base58
Maka saya melakukan scripting dengan bash sebagai berikut.

```
flag="J8pRND46rbHKmPuz4zWBNaWYzuo8uP6Kit4eFCnCgjGP7JWe8e9CVaK2
LitS7CmeQdcCueM"

for((;;)); do
	flag=`echo -n $flag | base58 -d`
	echo $flag
done
```

Lalu ketika saya run script diatas maka akan mengeluarkan output seperti berikut.

```
root@kali:~/icyption# bash btc.sh
GbDaDQz5BVRQg6xn6n1VEULC6jtXbdDmcZd2T1khCWS2mn7wFFun
icyption{satoshi_nakamoto_create_this}
```

Flag: icyption{satoshi_nakamoto_create_this}

Category Forensics

1. Something wrong with drive (150pts)

Diberikan sebuah file dengan nama “data.img” dan ketika saya cek menggunakan command “file” pada linux dan memberikan output seperti berikut.

```
root@kali:~/icyption# file data.img
data.img: data
```

Terlihat command file hanya menunjukkan ouput data, kemungkinan besar file .img ini telah corrupt. Maka untuk melakukan fix pada file tersebut kita dapat menggunakan command “e2fsck”.

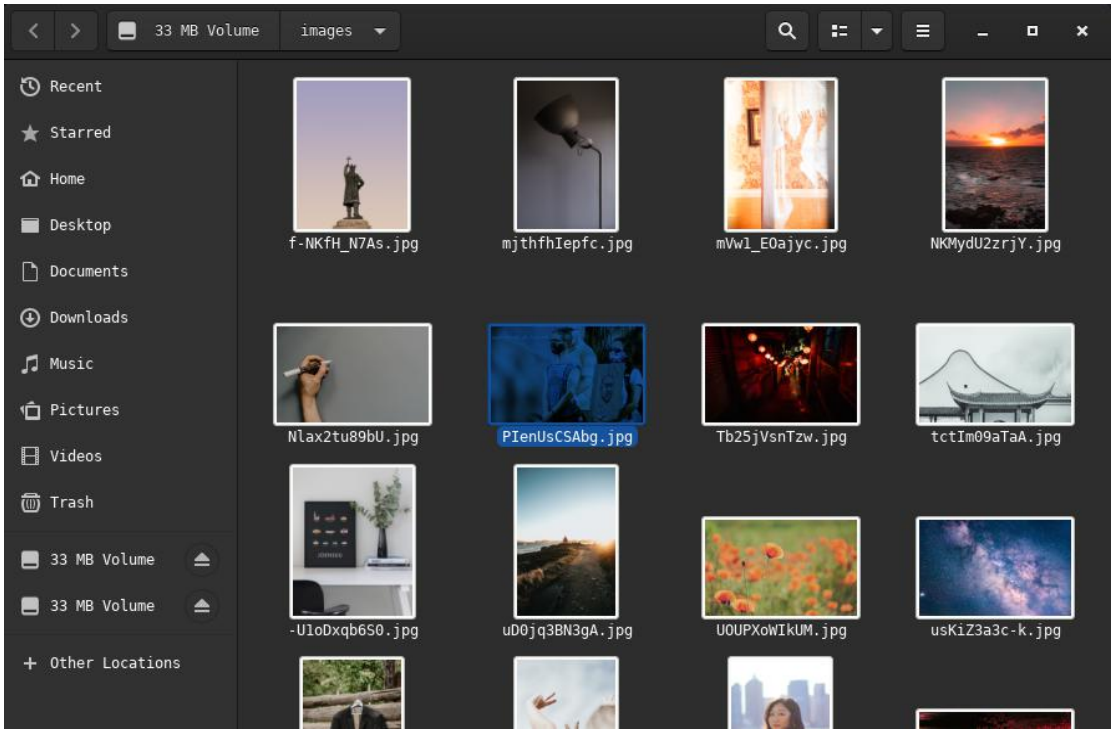
```
root@kali:~/icyption# e2fsck data.img
e2fsck 1.45.3 (14-Jul-2019)
ext2fs_open2: Bad magic number in super-block
e2fsck: Superblock invalid, trying backup blocks...
data.img was not cleanly unmounted, check forced.
Pass 1: Checking inodes, blocks, and sizes
Pass 2: Checking directory structure
Pass 3: Checking directory connectivity
Pass 4: Checking reference counts
Pass 5: Checking group summary information
Free blocks count wrong for group #0 (7803, counted=6436).
Fix<y>? yes
Free blocks count wrong for group #1 (7817, counted=5356).
Fix<y>? yes
Free blocks count wrong for group #2 (7942, counted=6988).
Fix<y>? yes
Free blocks count wrong (30354, counted=25572).
Fix<y>? yes
Free inodes count wrong for group #2 (1984, counted=1952).
Fix<y>? yes
Directories count wrong for group #2 (0, counted=1).
Fix<y>? yes
Free inodes count wrong (7925, counted=7893).
Fix<y>? yes

data.img: ***** FILE SYSTEM WAS MODIFIED *****
data.img: 43/7936 files (69.8% non-contiguous), 6172/31744 blocks
```

Setelah melakukan fix pada file tersebut kita dapat melihat lagi jenis filenya dan melakukan mount “data.img” ke sebuah folder.

```
root@kali:~/icyption# file data.img
data.img: Linux rev 1.0 ext2 filesystem data,
UUID=6fdb6fa9-bc9c-4970-a0be-8ad1de6ea848 (large files)
root@kali:~/icyption# mount data.img data
root@kali:~/icyption# ls data
images    lost+found
```

Lalu saya mencoba melihat lihat apa yang ada di dalam folder folder tersebut dan menemukan sebuah file gambar yang berisi flag.

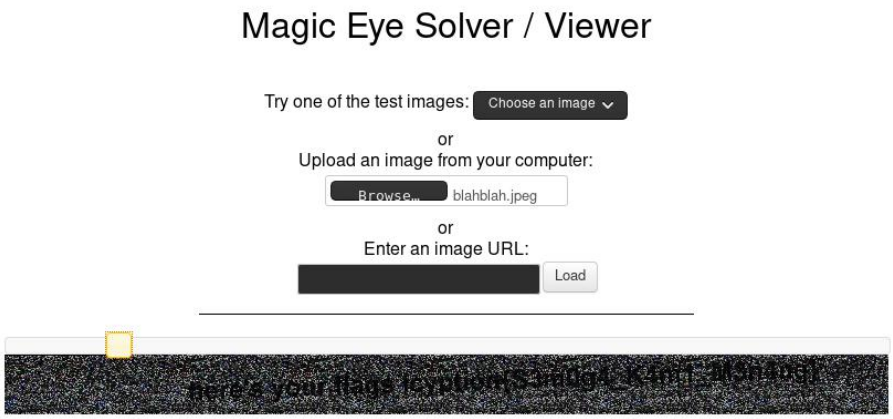


Maka kita buka filenya dan kita akan mendapatkan flag dari soal ini.

Flag: icyption{f1n4lly_y0u_f1nd_m3}

2. Wonderful Painting (80pts)

Pada soal ini kita diberikan sebuah file gambar dengan nama “blahblah.jpeg” dan ketika dibuka hanya menampilkan gambar yang tidak jelas maka kita mencoba menyelesaikan soal ini dengan stereogram.



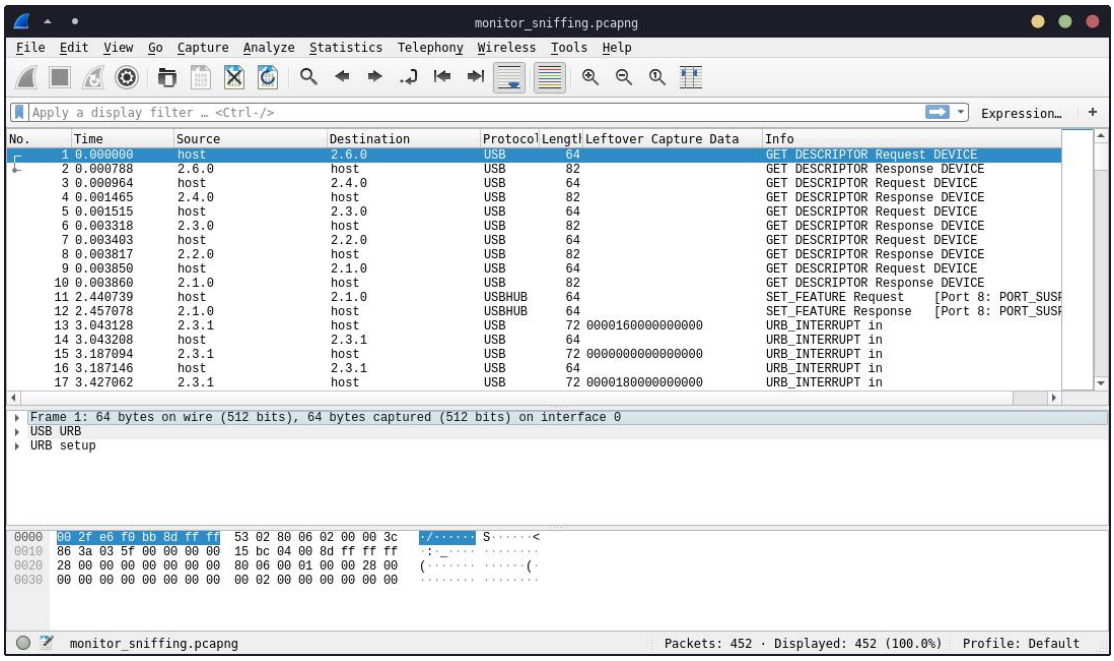
Dengan bantuan web ini kita akan dapat melihat flag yang terdapat pada gambar tersebut, sebenarnya kami sudah mencoba dengan stegsolve.jar namun gambar kurang jelas untuk dibaca.

Flag: icyption{S3m0g4_K4m1_M3n4n

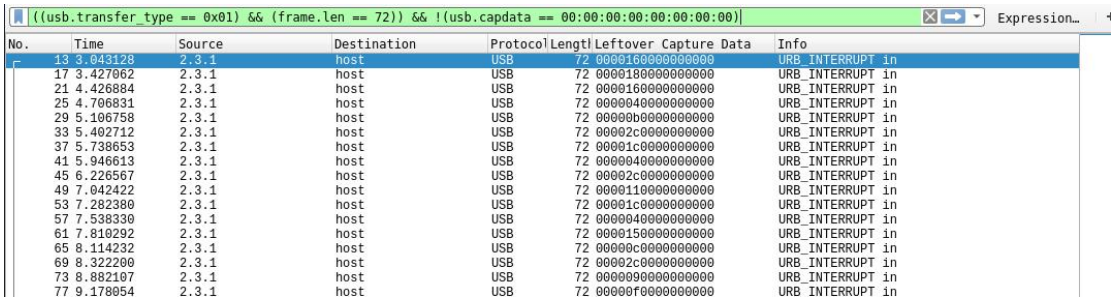
Category Misc

1. sniff sniff (175pts)

Diberikan sebuah file pcapng dan ketika kita buka dengan wireshark berisi banyak sekali packet USB.



Setelah melihat terdapat banyak packet USB maka saya menduga kita harus mencari tahu apa yang diketikan oleh pembuat soal pada keyboardnya. Maka saya melakukan filter pada leftover agar hanya ditampilkan yang memiliki leftover saja dan menyimpannya ke dalam file csv. Berikut pattern filter yang saya gunakan.



Nah sekarang yang ditampilkan hanya yang memiliki leftover dan kita bisa meng-export-nya menjadi file csv. Lalu mengambil leftovernya saja dengan command berikut.

```
root@kali:~/icyption# cat icyption.csv | cut -d ',' -f 7 | cut -d "" -f 2 | grep -vE
"Leftover Capture Data" > hexoutput.txt
root@kali:~/icyption# cat hexoutput.txt
0000160000000000
0000180000000000
0000160000000000
0000040000000000
[REDACTED]
```

Setelah itu kita dapat merubah hex tersebut menjadi keystroke, dengan menggunakan script berikut.

```
#!/usr/bin/python
# coding: utf-8
# source: https://gist.github.com/ImAnEnabler/091a9e1ee2d6a0805408e009e2f4a2b5
from __future__ import print_function
import sys,os

lcasekey = {}
ucasekey = {}

lcasekey[4]="a";                ucasekey[4]="A"
lcasekey[5]="b";                ucasekey[5]="B"
lcasekey[6]="c";                ucasekey[6]="C"
lcasekey[7]="d";                ucasekey[7]="D"
lcasekey[8]="e";                ucasekey[8]="E"
lcasekey[9]="f";                ucasekey[9]="F"
lcasekey[10]="g";               ucasekey[10]="G"
lcasekey[11]="h";               ucasekey[11]="H"
lcasekey[12]="i";               ucasekey[12]="I"
lcasekey[13]="j";               ucasekey[13]="J"
lcasekey[14]="k";               ucasekey[14]="K"
lcasekey[15]="l";               ucasekey[15]="L"
lcasekey[16]="m";               ucasekey[16]="M"
lcasekey[17]="n";               ucasekey[17]="N"
lcasekey[18]="o";               ucasekey[18]="O"
lcasekey[19]="p";               ucasekey[19]="P"
lcasekey[20]="q";               ucasekey[20]="Q"
lcasekey[21]="r";               ucasekey[21]="R"
lcasekey[22]="s";               ucasekey[22]="S"
lcasekey[23]="t";               ucasekey[23]="T"
lcasekey[24]="u";               ucasekey[24]="U"
lcasekey[25]="v";               ucasekey[25]="V"
lcasekey[26]="w";               ucasekey[26]="W"
lcasekey[27]="x";               ucasekey[27]="X"
lcasekey[28]="y";               ucasekey[28]="Y"
lcasekey[29]="z";               ucasekey[29]="Z"
lcasekey[30]="1";               ucasekey[30]="!"
lcasekey[31]="2";               ucasekey[31]="@"
lcasekey[32]="3";               ucasekey[32]="#"
lcasekey[33]="4";               ucasekey[33]="$"
lcasekey[34]="5";               ucasekey[34]="%"
lcasekey[35]="6";               ucasekey[35]="^"
lcasekey[36]="7";               ucasekey[36]="&"
lcasekey[37]="8";               ucasekey[37]="*"
lcasekey[38]="9";               ucasekey[38]="("
lcasekey[39]="0";               ucasekey[39]=")"
lcasekey[40]="Enter";           ucasekey[40]="Enter"
lcasekey[41]="esc";             ucasekey[41]="esc"
lcasekey[42]="del";             ucasekey[42]="del"
lcasekey[43]="tab";             ucasekey[43]="tab"
lcasekey[44]=" ";               ucasekey[44]=" "
lcasekey[45]="-";               ucasekey[45]="_"
lcasekey[46]="=";               ucasekey[46]="+"
lcasekey[47]="[";               ucasekey[47]="{"
lcasekey[48]="]";               ucasekey[48]="}"
lcasekey[49]="\\";              ucasekey[49]="|"
lcasekey[50]=" ";               ucasekey[50]=" "
lcasekey[51]=";";               ucasekey[51]=":"
```

```
lcasekey[52]="'";                ucasekey[52]="\""
lcasekey[53]="`";                ucasekey[53]="~"
lcasekey[54]=",";                ucasekey[54]="<"
lcasekey[55]=".";                ucasekey[55]=">"
lcasekey[56]="/";                ucasekey[56]="?"
lcasekey[57]="CapsLock";     ucasekey[57]="CapsLock"
lcasekey[79]="RightArrow"; ucasekey[79]="RightArrow"
lcasekey[80]="LeftArrow";   ucasekey[80]="LeftArrow"
lcasekey[84]="/";                ucasekey[84]="/"
lcasekey[85]="*";                ucasekey[85]="*"
lcasekey[86]="-";                ucasekey[86]="-"
lcasekey[87]="+";                 ucasekey[87]="+"
lcasekey[88]="Enter";          ucasekey[88]="Enter"
lcasekey[89]="1";                 ucasekey[89]="1"
lcasekey[90]="2";                 ucasekey[90]="2"
lcasekey[91]="3";                 ucasekey[91]="3"
lcasekey[92]="4";                 ucasekey[92]="4"
lcasekey[93]="5";                 ucasekey[93]="5"
lcasekey[94]="6";                 ucasekey[94]="6"
lcasekey[95]="7";                 ucasekey[95]="7"
lcasekey[96]="8";                 ucasekey[96]="8"
lcasekey[97]="9";                 ucasekey[97]="9"
lcasekey[98]="0";                 ucasekey[98]="0"
lcasekey[99]=".";                ucasekey[99]="."

if len(sys.argv) == 2:
        keycodes = open(sys.argv[1])
        for line in keycodes:
                bytesArray = bytearray.fromhex(line.strip())
                val = int(bytesArray[2])
                if val > 3 and val < 100:
                        if bytesArray[0] == 0x02 or bytesArray[0] == 0x20 :
                                print(ucasekey[int(bytesArray[2])], end=''),
                        else:
                                print(lcasekey[int(bytesArray[2])], end=''),
else:
        print("USAGE: python %s [filename]" % os.path.basename(__file__))
```

Setelah itu kita dapat run script diatas dan akan menghasilkan string yang telah di decode.

```
root@kali:~/icyption# python x.py hexoutput.txt
susah ya nyari flags nya ? ini lho flags nya :
icyptiion{Ww1r3sh4rk_n0t_only_f0r_n3tw0rks} . kettemu kan
```

Namun terdapat kesalahan yang menyebabkan beberapa huruf diketik 2x dan kita mencoba membetulkannya secara manual.

Flag: icyption{W1r3sh4rk_n0t_only_f0r_n3tw0rks}

Category Recon

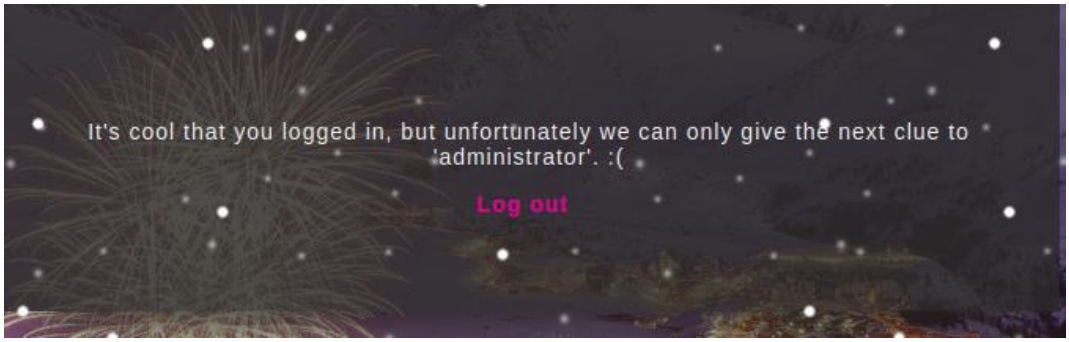
1. Any information on this website (100pts)

Buka link yang ada didescripsi soal, lalu klik pada button ‘LOGIN’ dan akan diarahkan pada page login. Liat source code, terdapat string yang terencode dengan base64.

```
            </div>
        </div>
    <script type="text/javascript" src="js/jquery-2.1.4.min.js"></script>
    <!-- VkhKNUlHZDFaWE4wTDJkMVpYTjBDZz09Cg== -->
</body>
```

```
abdullahnz ~/CTF/FINALICY echo "VkhKNUlHZDFaWE4wTDJkMVpYTjBDZz09Cg==" | base64 -d | base64 -d
Try guest/guest
```

Login dengan username & password tersebut, lalu continue.

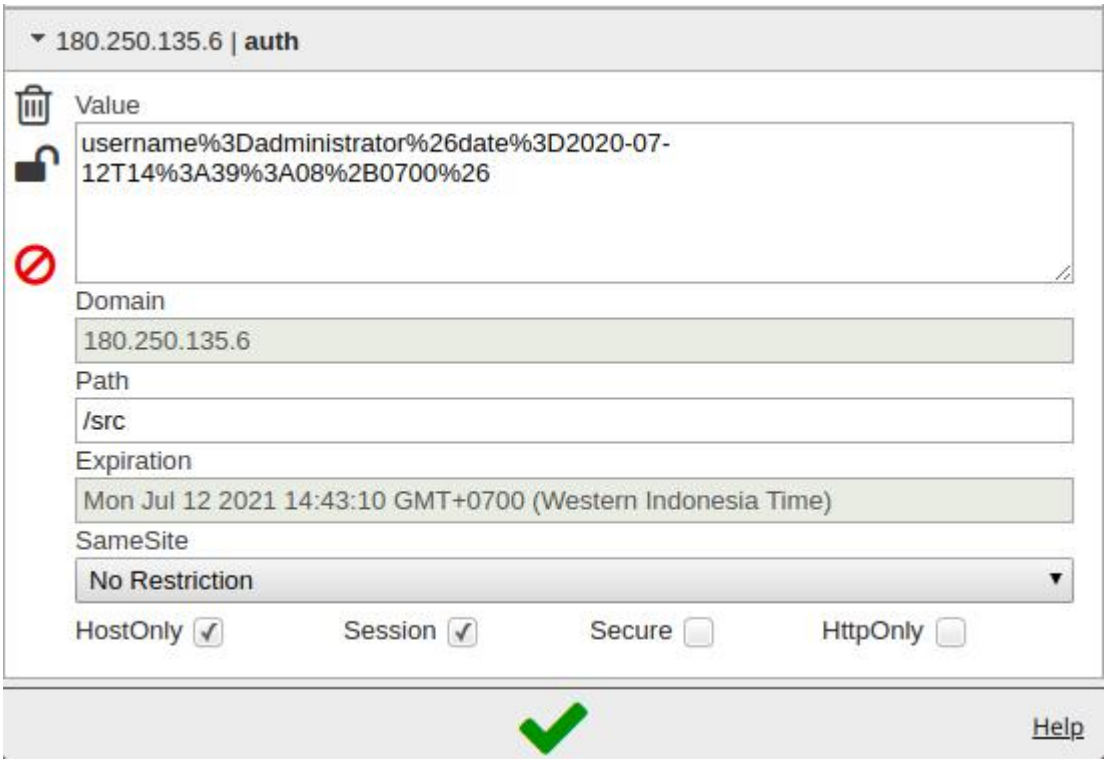


Didapatkan clue, yaitu “administrator”. Lalu check header website, didapatkan cookie data user yang dipakai untuk login tadi.

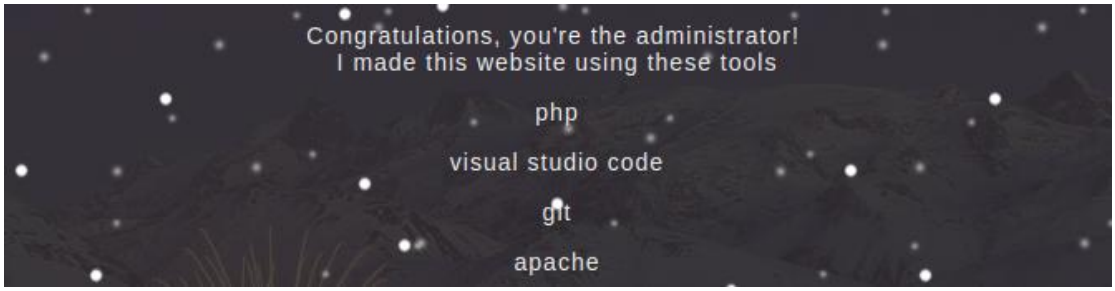
▾ **Response Headers** view source

Connection: keep-alive
Content-Encoding: gzip
Content-Type: text/html; charset=UTF-8
Date: Sun, 12 Jul 2020 07:39:08 GMT
Server: nginx/1.15.5 (Ubuntu)
Set-Cookie: auth=username%3Dguest%26date%3D2020-07-12T14%3A39%3A08%2B0700%26
Transfer-Encoding: chunked

Edit username pada cookie yang didapat menjadi “administrator”. Kami menggunakan EditThisCookie yang ada pada Chrome.



Lalu refrest halaman web, lalu didapatkan informasi mengenai website.



Akses folder git pada http://180.250.135.6:8080/src/.git. dan menampilkan “403 Forbidden”. Lalu dump menggunakan GitDumper.

GitTools: https://github.com/internetwache/GitTools/tree/master/Dumper

```
abdullahnz ~/GitTools/Dumper ./gitdumper.sh http://180.250.135.6:8080/src/.git/ out
###########
# GitDumper is part of https://github.com/internetwache/GitTools
#
# Developed and maintained by @gehaxelt from @internetwache
#
# Use at your own risk. Usage might be illegal in certain circumstances.
# Only for educational purposes!
###########


[*] Destination folder does not exist
[+] Creating out/.git/
[+] Downloaded: HEAD
[-] Downloaded: objects/info/packs
[+] Downloaded: description
[+] Downloaded: config
[+] Downloaded: COMMIT_EDITMSG
[+] Downloaded: index
[-] Downloaded: packed-refs
[-] Downloaded: refs/heads/master
[-] Downloaded: refs/remotes/origin/HEAD
[-] Downloaded: refs/stash
[-] Downloaded: logs/HEAD
[-] Downloaded: logs/refs/heads/master
[-] Downloaded: logs/refs/remotes/origin/HEAD
[-] Downloaded: info/refs
[+] Downloaded: info/exclude
```

Check folder git dalam direktori out, didapatkan flag pada config file.

```
abdullahnz@zeroday: ~/GitTools/Dumper

abdullahnz ~/GitTools/Dumper cat out/.git/config
[core]
        repositoryformatversion = 0
        filemode = true
        bare = false
        logallrefupdates = true
        #icyption{1N1_kaN_Y4Ng_kaMu_Cari_h3he}

abdullahnz ~/GitTools/Dumper _
```

FLAG : icyption{1N1_kaN_Y4Ng_kaMu_Cari_h3he}

Category Encryption

1. Hannah needs your help (200 pts)

Diberikan cipher yang merupakan angka hasil encripsi perbyte flag dan *sesuatu,* karena saya tidak paham apa yang dimaksud *totient function*. Singkat saja ini merupakan hasil enkripsi RSA karena ada *phi, N, totiend(N).* (sebenarnya penulis juga tidak tahu banyak tentang istilah-istilah yang ada di-RSA.)
Dengan kemampuan matematika penulis, penulis bisa langsung menemukan faktorisasi N(142), yaitu 11 dan 13 (pq). Atau bisa dengan menggunakan factordb.com untuk mencari faktorisasi N.

digits	number
3 (show)	143 = 11 · 13

Berikut solver yang kami buat untuk mendeskripsi cipher.

```
# -*- coding: utf8 -*-

from Crypto.Util.number import *
import gmpy2

def decrypt(e, cipher):
	N = 143; p = 11; q = 13
	phi = (p - 1) * (q - 1)
	result = ""
	for c in cipher:
		d = gmpy2.invert(e, phi)
		result += long_to_bytes(pow(c, d, N))
	return result

cipher = [118, 44, 121, 18, 129, 118, 45, 33, 7, 21, 116, 21, 13, 33, 38, 17, 49, 13,
100, 13, 17, 49, 13, 100, 13, 17, 80, 39, 80, 13, 91, 17, 50, 39, 38, 13, 5]

for e in range(0x10001):
	try:
		msg = decrypt(e, cipher)
		if msg.startswith('icyption{'):
			print (e, msg)
			break
	except:
		pass
```

Karena ‘e’ tidak diketahui, maka kami mem-bruteforce nilai ‘e’ dari 0-65537. Dan break program saat flag ditemukan.

```
abdullahnz@zeroday: ~/CTF/FINALICY

abdullahnz ~/CTF/FINALICY python solver_encryption.py
(7, 'icyption{m3m4ng_r4d4_r4d4_sus4h_jug4}')
```

FLAG : icyption{m3m4ng_r4d4_r4d4_sus4h_jug4}