

Writeup Penyisihan CTF PETIR

Tim BlackHat

Bina Nusantara University

Daftar Isi:

Forensic

corrupted

notbroken

Crypto

Caesar

Encoding

Hash Lookup

Hash Crack

Hashing

Vinegere

Reverse

Ezrevrse

Misc

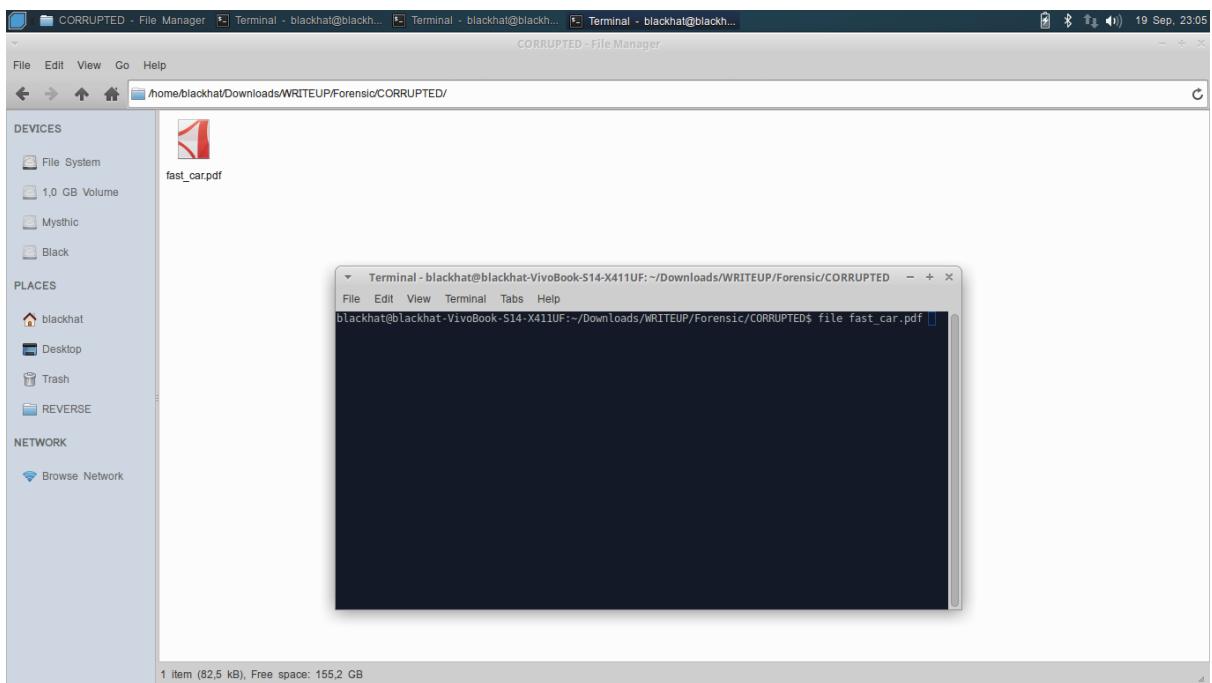
99levels

small-picture

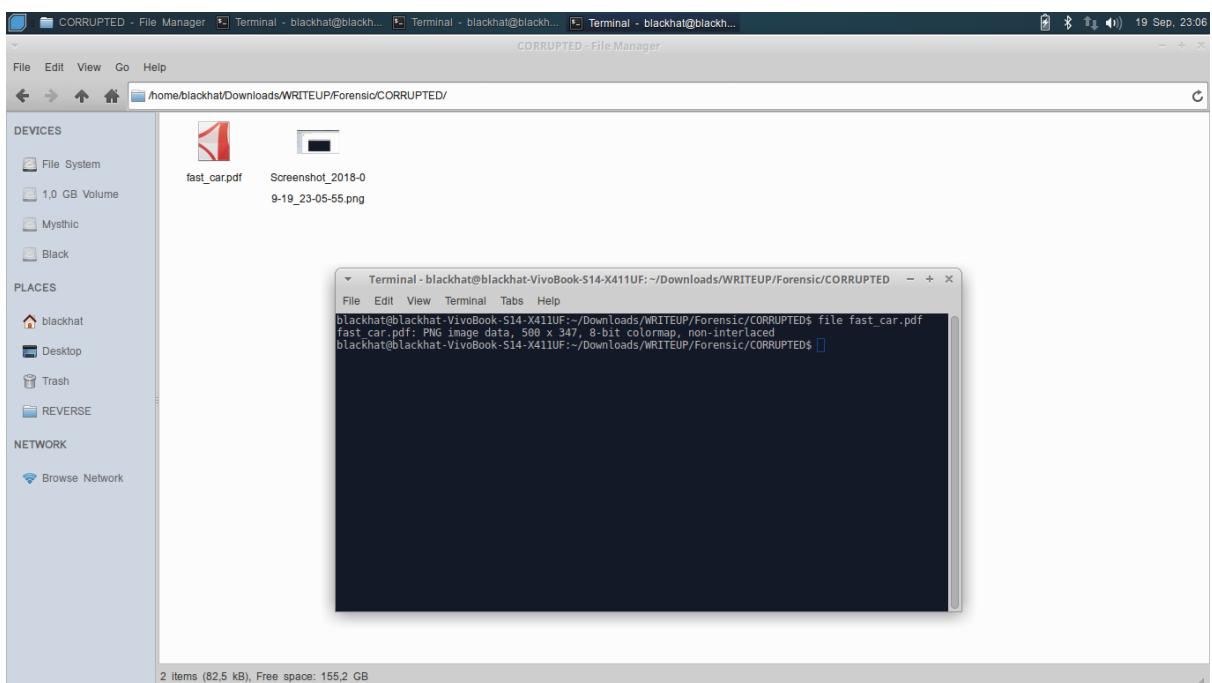
FORENSIC

corrupted (421 pts)

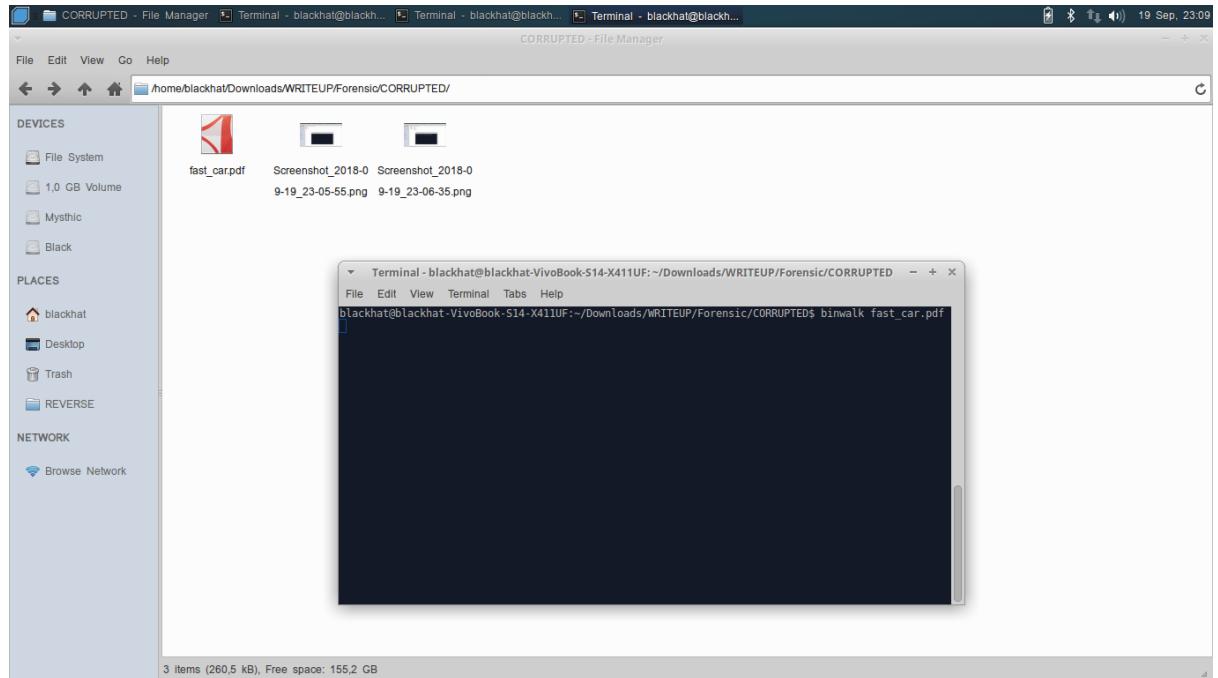
diberikan sebuah file bernama *fast_car.pdf* yang tidak diketahui apa file itu dan saat dibuka tidak dapat dijalankan karena itu untuk mengidentifikasi file tersebut saya membuka terminal dan menuju ke *directory* tersebut dan mengidentifikasi file tersebut dengan fungsi “file” di terminal linux



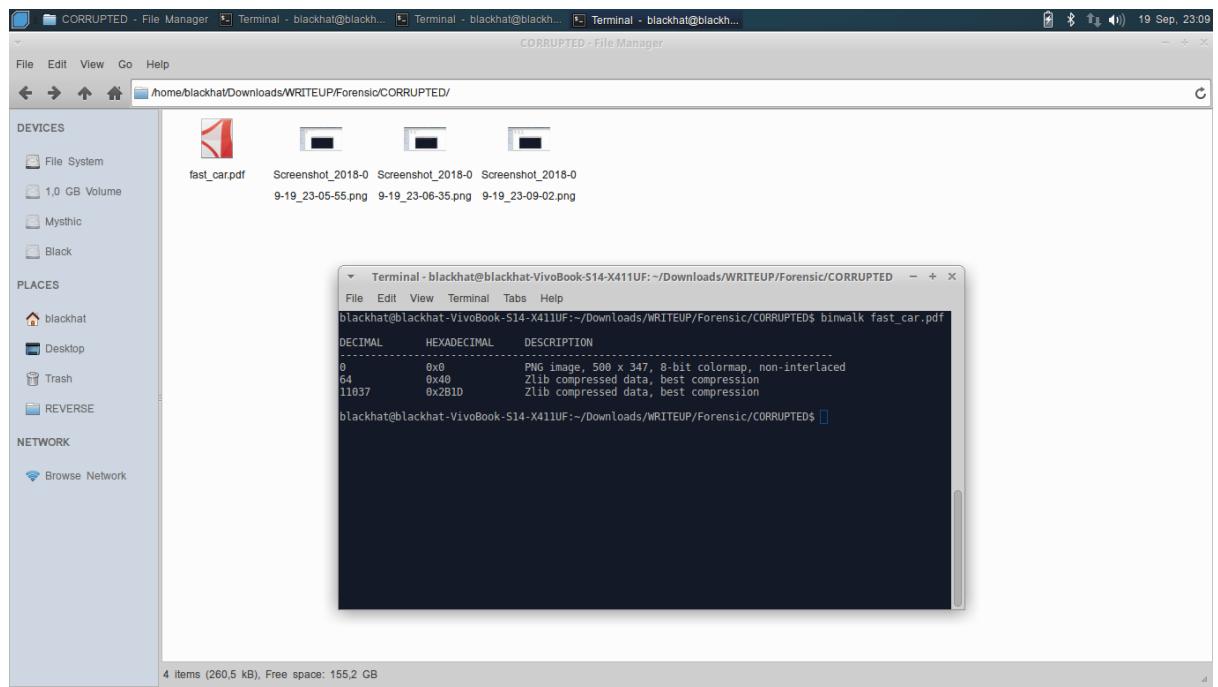
Dengan cara ketik “file_<file yang ingin diidentifikasi>”. Setelah saya identifikasi ternyata bukan file *pdf* namun file *png*.



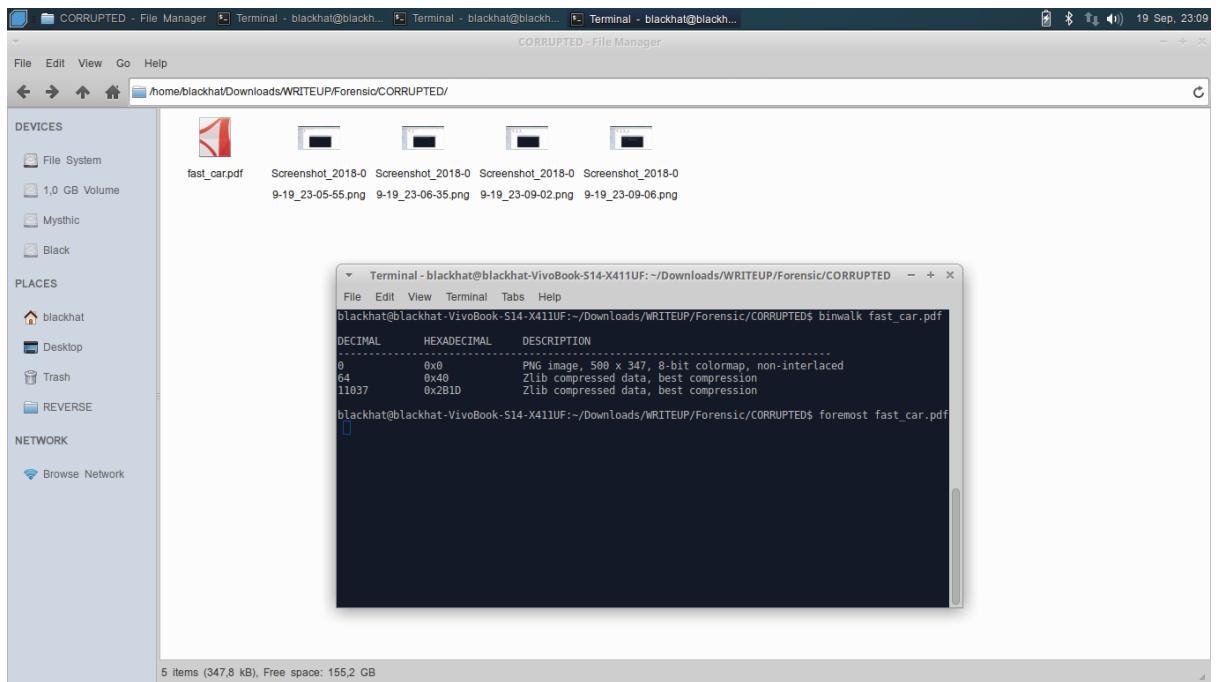
Setelah saya mengerti itu file *png*, saya ingin mengetahui apa saja yang ada didalam file tersebut dengan cara menggunakan fungsi “binwalk” pada terminal linux.



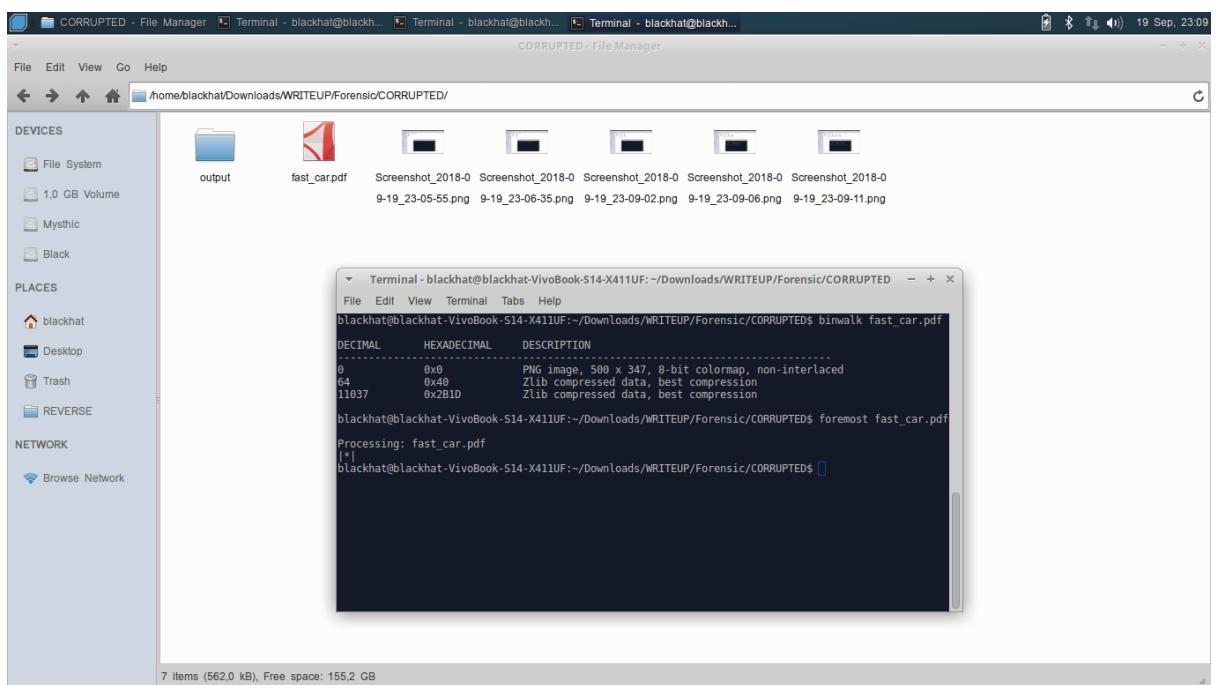
Dengan cara ketik “`binwalk <file yang ingin diketahui>`”, setelah saya jalankan ternyata benar terdapat *signature / file png* didalam file tersebut.



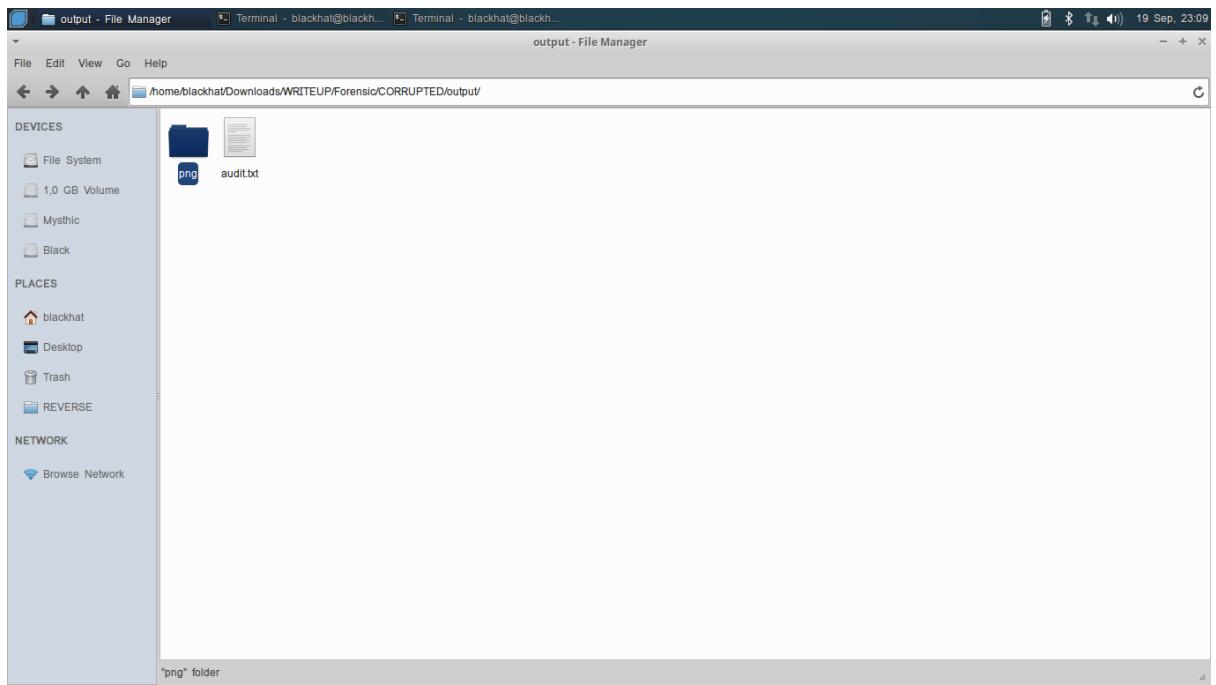
Lalu setelah saya mengetahuinya saya menggunakan fungsi “foremost” pada terminal linux untuk meng *extract* seluruh data yang ada didalam file tersebut.



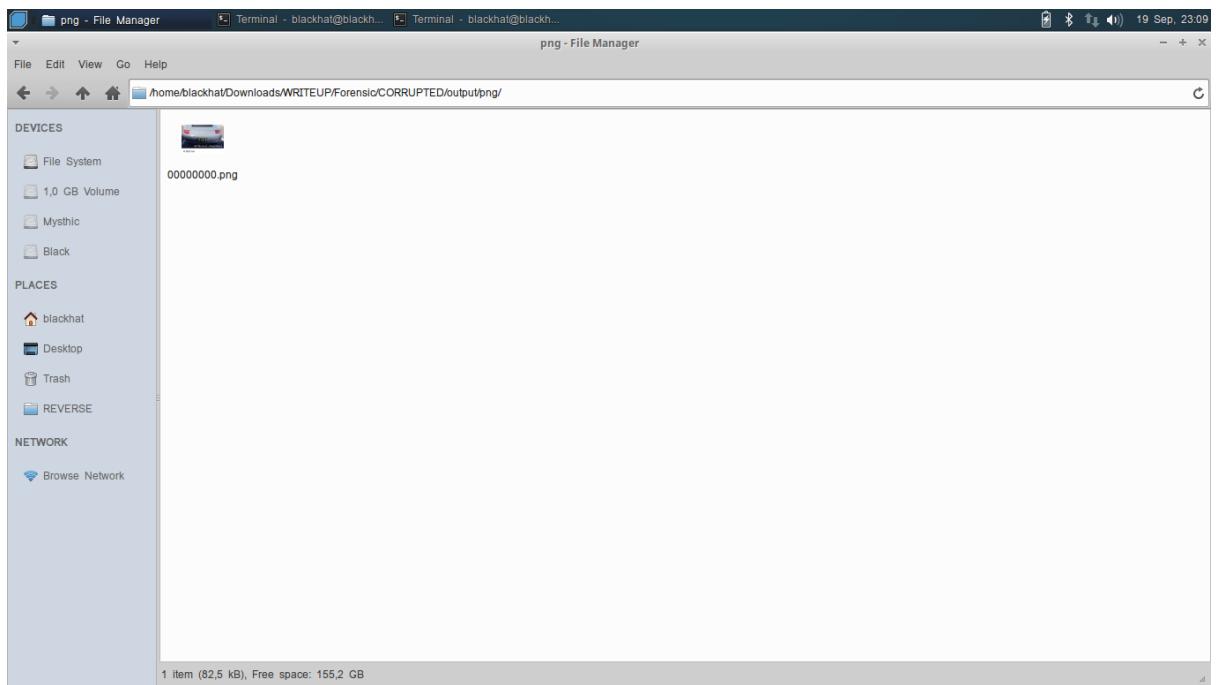
Dengan cara ketik “foremost_<file yang ingin di extract”, lalu saya mendapatkan folder output yang merupakan hasil extract semua file yang ada di dalam file tersebut.

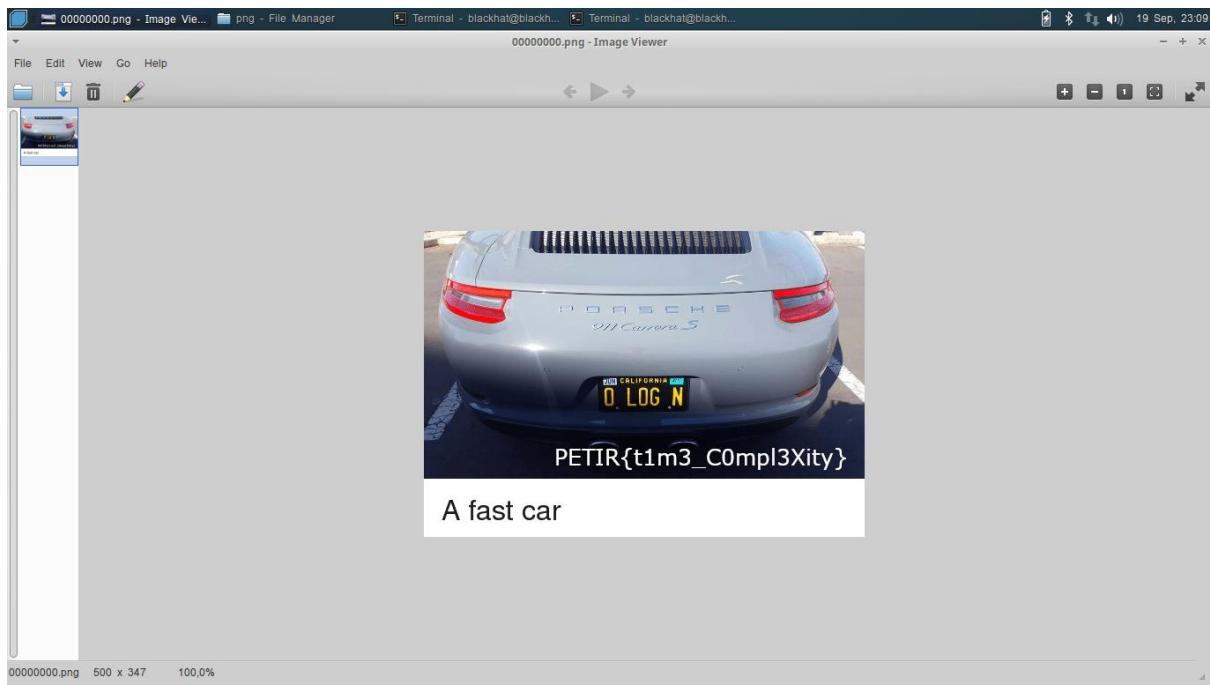


Setelah saya buka folder output, terdapat hasil file – file hasil extract dari dalam file tersebut.



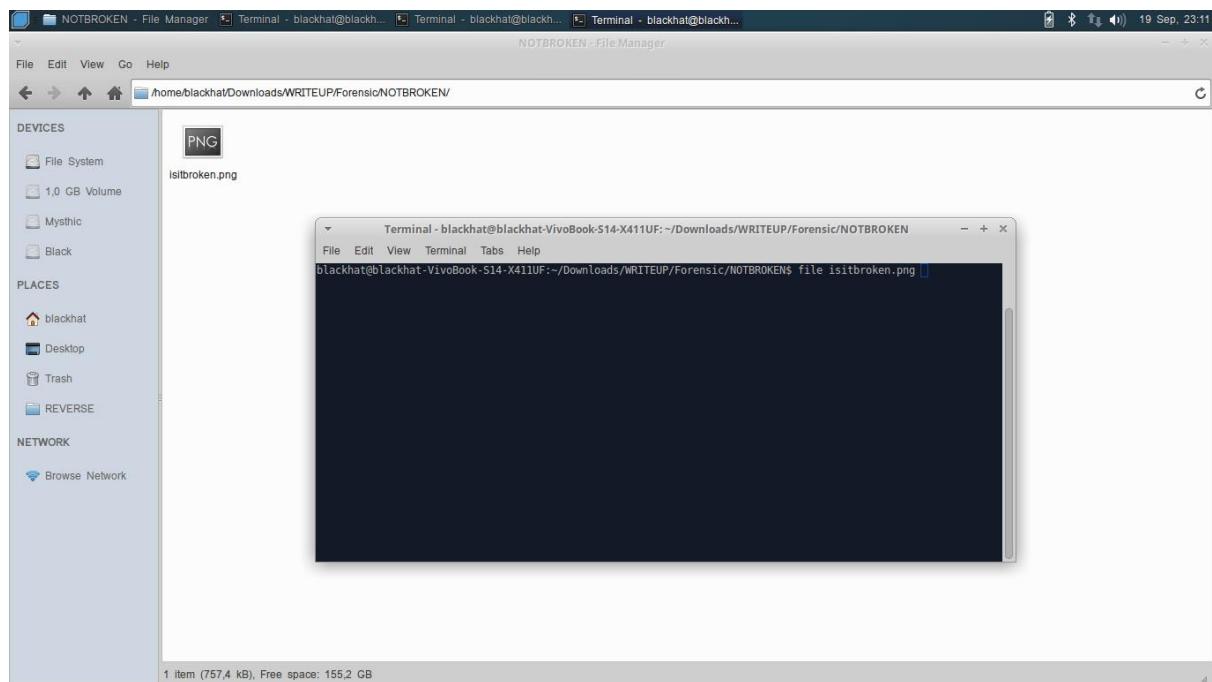
Terdapat folder png dan audit.txt, audit.txt adalah ringkasan apa saja yang dapat di extract menggunakan foremost di dalam file tersebut. Setelah saya membuka file png terdapat sebuah foto yang berisi flagnya.



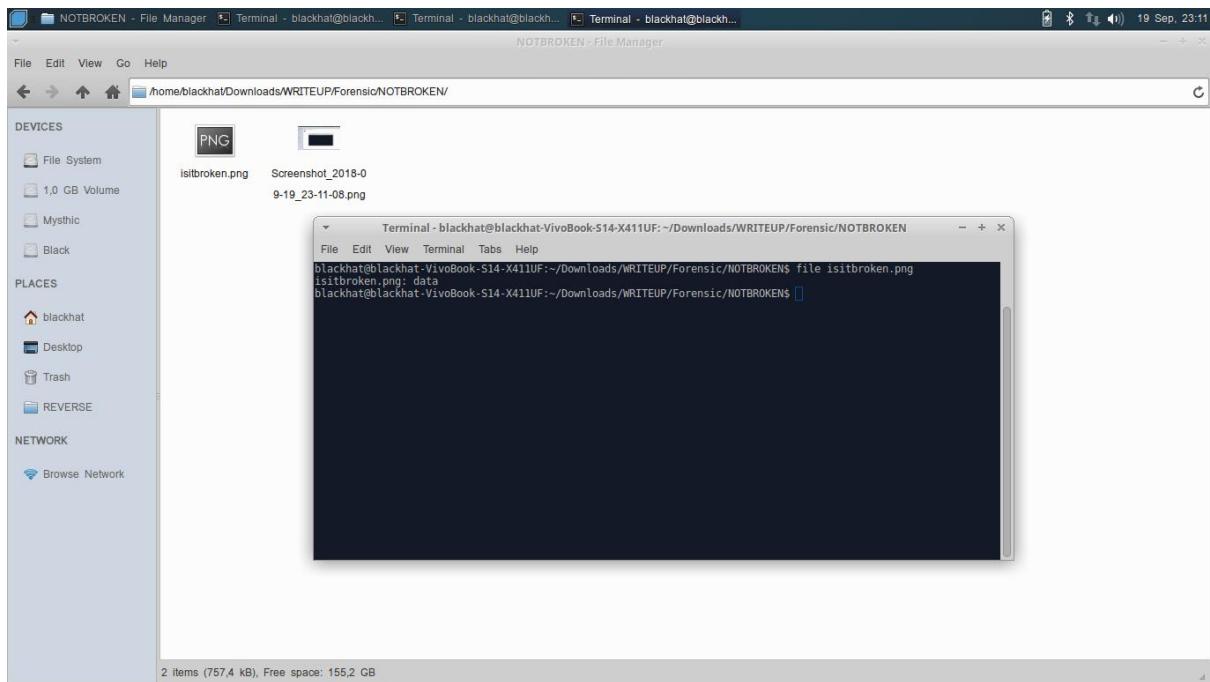


notbroken (448 pts)

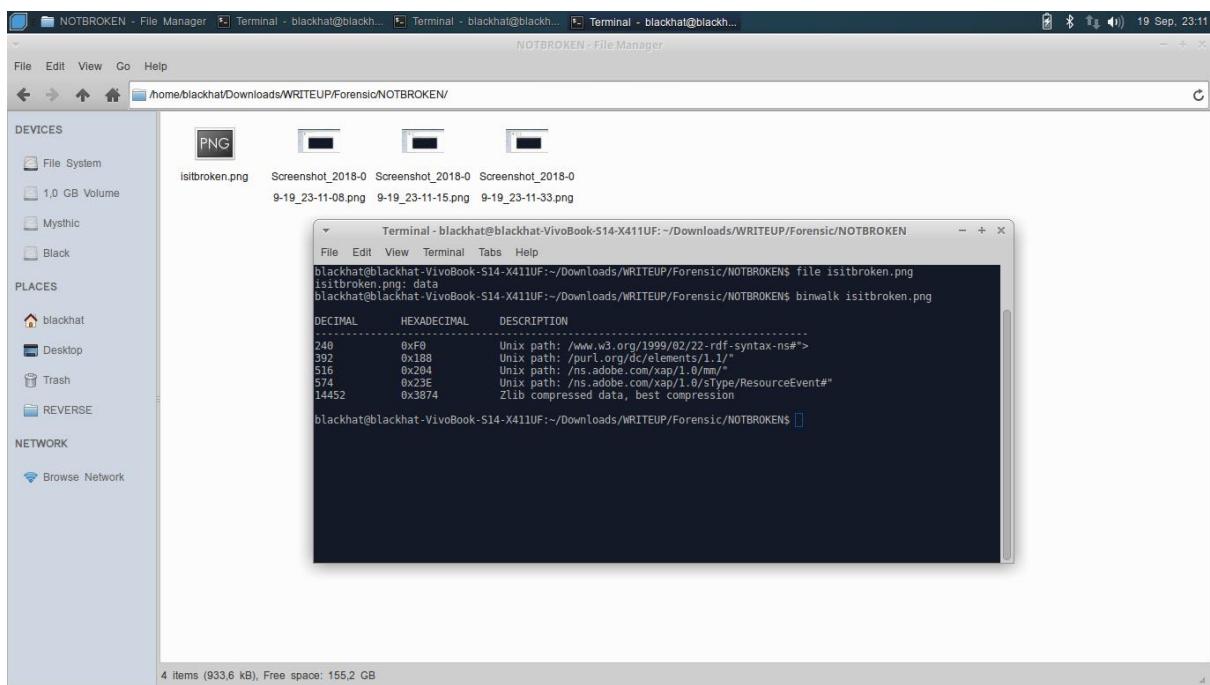
diberikan sebuah file PNG file dan ketika saat saya buka tidak dapat dibuka file tersebut oleh sebab itu saya membuka terminal dalam *directory* tersebut dan menggunakan fungsi “file” dalam terminal linux untuk mengetahuinya.



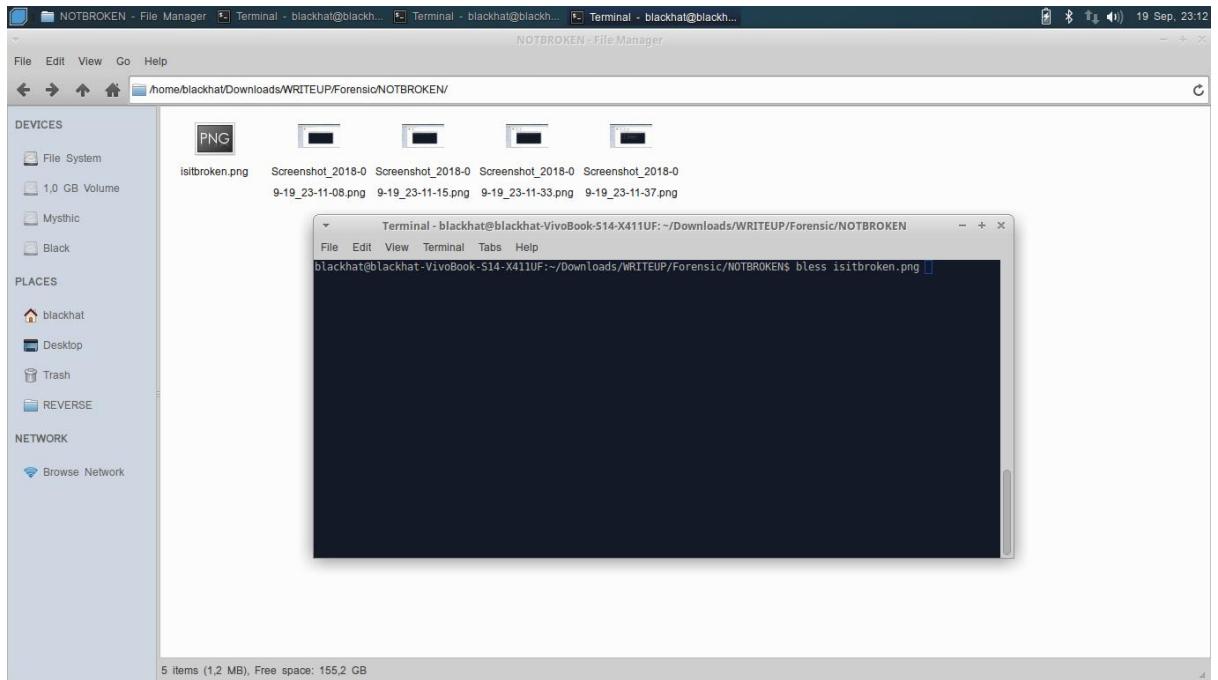
Sesuai expectasi saya menemukan bahwa file tersebut tidak sesuai dengan apa yang dituliskan yaitu file tersebut adalah file data



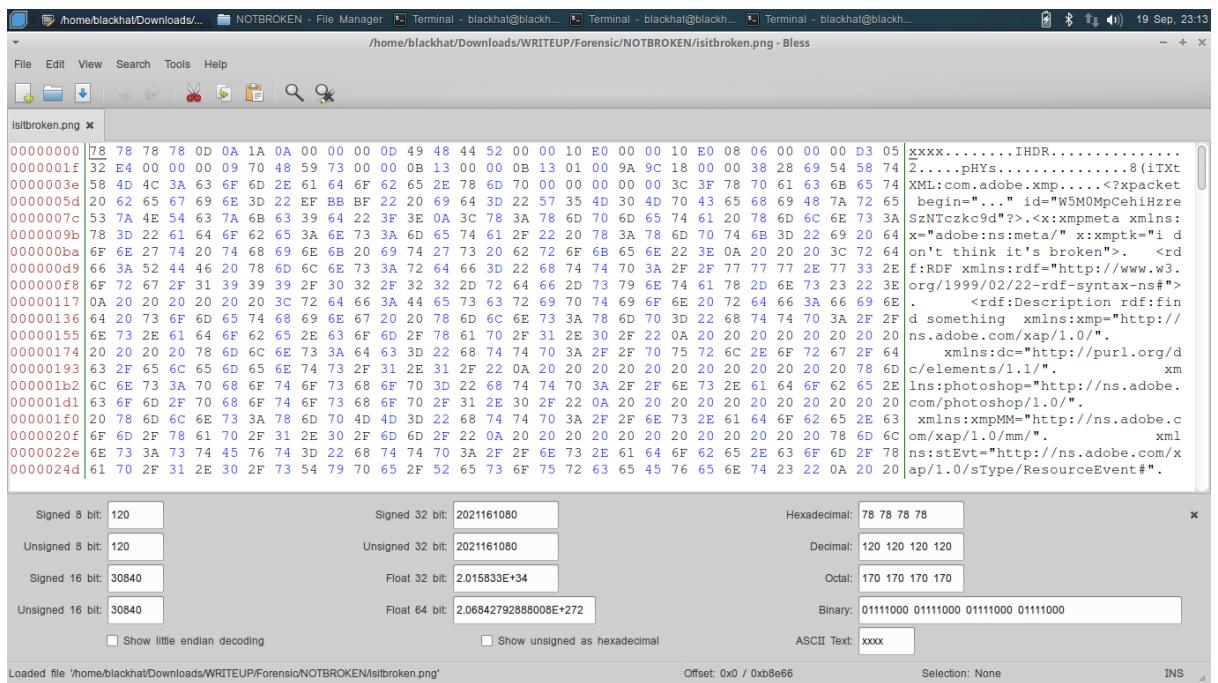
Lalu saya mengidentifikasi file tersebut dengan fungsi “binwalk” dan menghasilkan



Karena file tidak dapat di *extract* dengan foremost maka saya menggunakan aplikasi “Hex Editor” bernama “Bless” untuk mengetahui apa saja yang ada didalam file.



Didalam aplikasi bless kita dapat melihat kode ASCII dan Hex sehingga menghasilkan.



Setelah saya cari tau suluruhnya hingga paling bawah saya menemukan flagnya 😊.

The screenshot shows a hex editor window titled 'isitbroken.png' with the file path '/home/blackhat/Downloads/WRITEUP/Forensic/NOTBROKEN/isitbroken.png - Bless'. The main pane displays the binary content of the file, which includes the string 'PETIR{IT_ls_Br0k3n_F0r_scm3_R3asOn}' in ASCII. Below the hex dump, there are several conversion boxes:

- Signed 8 bit: 80
- Signed 32 bit: 1346720841
- Hexadecimal: 50 45 54 49
- Unsigned 8 bit: 80
- Unsigned 32 bit: 1346720841
- Decimal: 080 069 084 073
- Signed 16 bit: 20549
- Float 32 bit: 1.324254E+10
- Octal: 120 105 124 111
- Unsigned 16 bit: 20549
- Float 64 bit: 4.93951540474146E+76
- Binary: 01010000 01000101 01010100 01001001
- Show little endian decoding
- Show unsigned as hexadecimal
- ASCII Text: PETI

At the bottom, it shows the offset from 0xb8e44 to 0xb8e66 (0x23 bytes) and the instruction (INS).

CRYPTO

Caesar (359 pts)

Pada saat saya membuka soal tersebut saya diberikan kode yang tidak dapat dimengerti dengan Bahasa. Lalu saat saya membaca nama soal ini saya mengerti bahwa itu adalah Caesar Code

The screenshot shows a web browser window for the Petir Qualification challenge 'Caesar' with a value of 359 and 55 solves. The challenge page includes the flag: [GVKZl\[Ky1\]_tzgy3l_jy0lcU_y4m3_uz3u1](#). The interface includes a 'Challenge' tab, a 'Submit' button, and a sidebar with other challenges like Hashing, Vigenere, Hashcrack, DES, Encoding, and AES.

Setelah itu saya membuka web browser dan membuka website <http://rumkin.com/tools/cipher/> dan saya memilih Caesarian Shift untuk membantu saya dalam mentranslate kode tersebut.

Cipher Tools

Rumkin.com >> Web-Based Tools >> Ciphers and Codes

Let's say that you need to send your friend a message, but you don't want another person to know what it is. You can use a full-blown encryption tool, such as PGP. If the message isn't that important or if it is intended to be decrypted by hand, you should use a simpler tool. This is a page dedicated to simple text manipulation tools, which all can be replicated with just paper and pencil.

If you know of another cipher that you think should be on here, leave me a message below.

Affine	Similar to a Caesar cipher, but also adds in a multiplier to further scramble letters.
Atbash	A very simplistic cipher where you change A into Z, B into Y, and so on.
Baconian	Used to hide a message within another message, by using different typefaces or other distinguishing characteristics.
Base64	This is typically used to make binary data safe to transport as strictly text.
Bifid	Breaks information for each letter up and spreads it out in the encoded message. An easy and fairly secure pencil & paper cipher.
Caesarian Shift	Where ROT13 was based on you adding 13 to the letters, a Caesar cipher lets you add an arbitrary value. Again, you can do it with the cryptogram solver, but you can scroll through values of N pretty easily with this tool.
Keyed Caesar	Similar to a Caesar cipher, but you first alter the encoded alphabet with a word or phrase.
Columnar Transposition	Write a message as a long column and then swap around the columns. Read the message going down the columns. A simple cipher, but one that is featured on the Kryptos sculpture at the CIA headquarters.
Double Transposition	Because two is better than one. Used by the U.S. Army during World War II.
Cryptogram Solver	This helps you solve simple ciphers; the methods where you replace letter X with letter Y.
Gronsfeld	The exact same thing as a Vigenere cipher, but it uses numbers instead of a key word.

INDEX

- Affine
- Atbash
- Baconian
- Base64
- Bifid
- Caesar
 - Keyed
 - ROT13
 - Column Trans.
 - Double
 - Ubchi
- Cryptogram
- Gronsfeld
- Morse
- Numbers
- One Time Pad
- Playfair
- Railfence
- Rotate
- Skip
- Substitution
- Vigenere
 - Keyed
 - Autokey
- Crypto Solver
- Frequency

Setelah saya masuk saya mengopy kata – kata dalam soal tersebut dan memnempel pada kolom yang disediakan untuk mengisi kata apa yang ingin terjemahkan

Caesarian Shift

Rumkin.com >> Web-Based Tools >> Ciphers and Codes

This is a standard Caesarian Shift cipher encoder, also known as a rot-N encoder and is also a style of substitution cipher. This way, you can add one, two, or any number up to 25 to your string and see how it changes. This is an offshoot of the [rot13](#) encoder on this web site. To perform this shift by hand, you could just write the alphabet on two strips of paper. Line them up so the top strip's A matches the bottom strip's D (or something) and then you can encode. A simple test to see how this works would be to [insert the alphabet](#) into the encoder and then change the values of N.

This sort of cipher can also be known as a wheel cipher. This is where an inner wheel has the alphabet around the outside, and that is placed upon an outer wheel, also with the alphabet going around it. You can rotate the wheels so that ABC lines up with ABC, or ABC may line up with QRS.

To encode something, just pick an N and type in your message. To decode something, subtract the encryption N from 26 and it should be decoded for you.

N:

This is your encoded or decoded text:

INDEX

- Affine
- Atbash
- Baconian
- Base64
- Bifid
- Caesar
 - Keyed
 - ROT13
 - Column Trans.
 - Double
 - Ubchi
- Cryptogram
- Gronsfeld
- Morse
- Numbers
- One Time Pad
- Playfair
- Railfence
- Rotate
- Skip
- Substitution
- Vigenere
 - Keyed
 - Autokey
- Crypto Solver
- Frequency

N tersebut adalah syarat yang harus diberikan agar dapat merangkai sebuah kata yang baku atau flag dari soal tersebut. Caesar memiliki prinsip bahwa jika N = 1 maka alphabet akan bergeser 1 huruf A menjadi B, B menjadi C, C menjadi D dan seterusnya. Jika N = 2 maka alphabet akan bergeser 2 huruf, contoh A menjadi C, B menjadi D, D menjadi F dan seterusnya.

Setelah saya coba satu persatu lalu saya menemukan pada $N = 9$

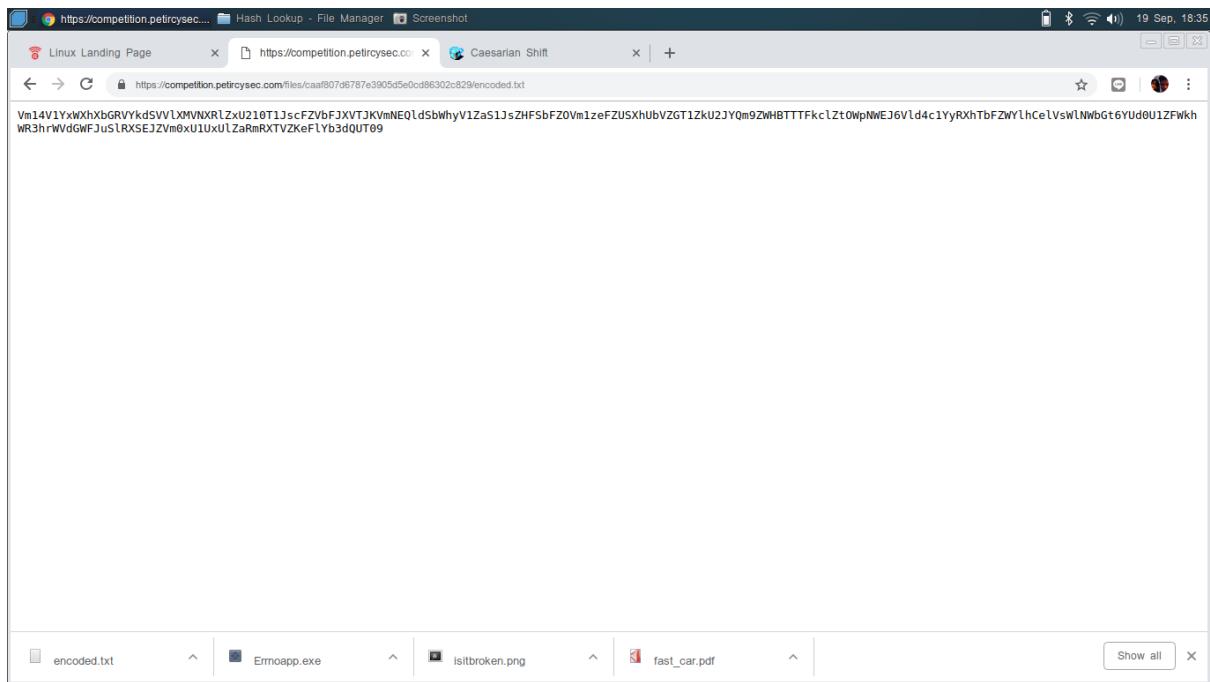
This screenshot shows a web-based Caesarian Shift cipher encoder. The URL is rumkin.com/tools/cipher/caesar.php. The page title is "Caesarian Shift". The sidebar on the right lists various cipher types and tools. The main area shows the input text "GVKZI{Ky1j_tzgy3i_jy0lcU_y4m3_u23u}" and the output text "PETIR(Th1s_ciph3r_sh0uID_h4v3_di3d)". A dropdown menu shows "N: 9".

Dan pada $N = 19$ atau alphabet bergeser 9 kali saya menemukan flagnya.

Encoding (436 pts)

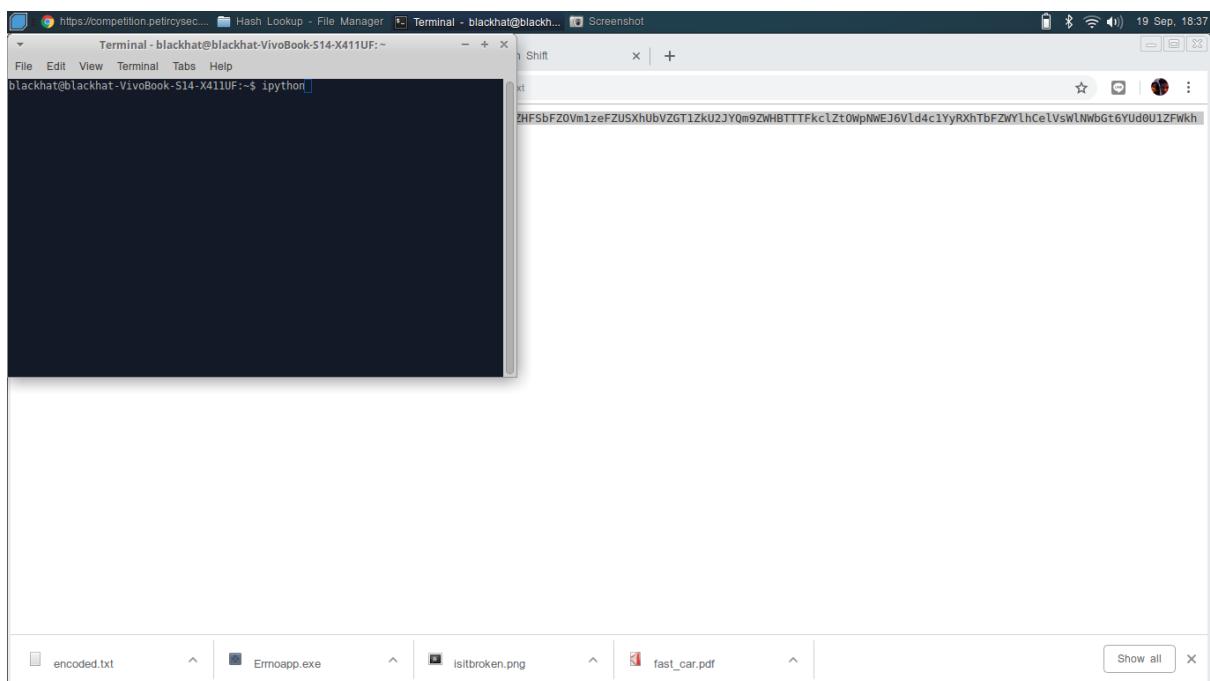
Diberikan sebuah file txt yang berisi file hasil encode dari sebuah kata.

This screenshot shows a challenge titled "Encoding 436" from the Petir Qualification competition. The challenge interface includes a "Challenge" tab, a "37 Solves" counter, and a "Submit" button. Below the challenge title, there is a file download link for "encoded.txt". The challenge area displays several flagged items with their respective scores: "corrupted" (421), "motorcycle" (448), "forensIKE" (499), "flaghunter" (500), "ritssluiting" (500), "babyshark" (500), and "rumpledforensic" (500). The sidebar on the left shows the category "Forensic". The bottom navigation bar includes links for "encoded.txt", "Ernoapp.exe", "isitbroken.png", and "fast_car.pdf".



Setelah saya mengidentifikasi saya mengerti bahwa ini adalah kode “base64” yang dapat diketahui dengan hurufnya tidak beraturan, huruf dan angka dicampur menjadi satu.

Saya akhirnya membuka terminal dalam *directory* tersebut dan menggunakan fungsi “ipython” untuk menerjemahkan file menjadi text.



Setelah itu saya mengopy kode yang ada dalam file text tersebut dan diberi tanda kutip di “ipython” dan mendecode file tersebut dengan kodennya yaitu base64. Dengan cara “<kode>”.decode(“base64”).

```
Terminal - blackhat@blackhat-VivoBook-S14-X411UF:~
```

```
File Edit View Terminal Tabs Help
```

```
blackhat@blackhat-VivoBook-S14-X411UF:~$ ipython
```

```
Python 2.7.12 (default, Dec 4 2017, 14:59:18)
```

```
Type "copyright", "credits" or "license" for more information.
```

```
IPython 2.4.1 -- An enhanced Interactive Python.
```

```
? -> Introduction and overview of IPython's features.
```

```
%quickref -> Quick reference.
```

```
help -> Python's own help system.
```

```
object? -> Details about 'object', use 'object??' for extra details.
```

```
In [1]: "Vm14V1YxWxbGRVYkdSVVlXmNxRlZxU210T1JscFZbJXfTKVmNEQdSbwHvV1ZaS1zZHFSFBZ0Vm1zeFZUSXhUbVZGT1ZkU2JY0m9ZWHBTTFkclZt0wpME36Vld4c1YyRkhfbZnYlhCevswLNwGt6YUd01ZFWkh"
```

```
Out[1]: 'Vm14V1YxWldubGRUYw1SwFVqSmtoRlpVu1RwU2JvcDBwRmrhWZKRldqRlVNMsxVTIxTmF0Vd5bxBoXpSMdrVm9jMXBzWxsV2ExSLVVbxBzUzSVlkzaGtSVEZHVGxkyWfxRnJTwHBYVm1SU1RFZFdWVJxyXowPQ=='
```

```
In [2]:
```

Setelah saya decode saya belum menemukan flagnya lalu saya coba decode lagi dengan cara yang sama.

```
Terminal - blackhat@blackhat-VivoBook-S14-X411UF:~
```

```
File Edit View Terminal Tabs Help
```

```
blackhat@blackhat-VivoBook-S14-X411UF:~$ ipython
```

```
Python 2.7.12 (default, Dec 4 2017, 14:59:18)
```

```
Type "copyright", "credits" or "license" for more information.
```

```
IPython 2.4.1 -- An enhanced Interactive Python.
```

```
? -> Introduction and overview of IPython's features.
```

```
%quickref -> Quick reference.
```

```
help -> Python's own help system.
```

```
object? -> Details about 'object', use 'object??' for extra details.
```

```
In [1]: "Vm14V1YxWxbGRVYkdSVVlXmNxRlZxU210T1JscFZbJXfTKVmNEQdSbwHvV1ZaS1zZHFSFBZ0Vm1zeFZUSXhUbVZGT1ZkU2JY0m9ZWHBTTFkclZt0wpME36Vld4c1YyRkhfbZnYlhCevswLNwGt6YUd01ZFWkh"
```

```
Out[1]: 'Vm14V1YxWldubGRUYw1SwFVqSmtoRlpVu1RwU2JvcDBwRmrhWZKRldqRlVNMsxVTIxTmF0Vd5bxBoXpSMdrVm9jMXBzWxsV2ExSLVVbxBzUzSVlkzaGtSVEZHVGxkyWfxRnJTwHBYVm1SU1RFZFdWVJxyXowPQ=='
```

```
In [2]:
```

Lalu saya decode hingga 5x saya akhirnya menemukan flagnya.

```

Terminal - blackhat@blackhat-VivoBook-S14-X411UF:~ [2]: KeyboardInterrupt
Terminal - blackhat@blackhat-VivoBook-S14-X411UF:~ [2]: "Vm1vV1YxWxhXbGRVYkdSVVLXMVNXRlZxU210T1JscFZVbFJXVTJKVmNEQldSbWhyV1ZaS1JsZHFSbFZ0Vm1zeF
File Edit View Terminal Tabs Help
File Edit View Search Tools Documents Help
Save - + ×
Terminal - blackhat@blackhat...
Terminal - blackhat@blackhat...
*Untitled Document 1
File Edit View Terminal Tabs Help
File Edit View Search Tools Documents Help
Save - + ×
Vm1vV1YxWxhXbGRVYkdSVVLXMVNXRlZxU210T1JscFZVbFJXVTJKVmNEQldSbWhyV1ZaS1JsZHFSbFZ0Vm1zeF
FOVd5xb0yoXpSM1drVm9jMXBzVwxzVExSVVbxBzlvSVlkzaGtVzEHGxkWFxRnJTwBYm15U1R
ZF0dMvVxyXwvP0==", decode("base64")
out[2]: "VlWV1ZWTdtaamXUjJKNFZURTVsbUp0VFhKVJFw)FUMVkiU21NeE9WRphazR3WkVoc1p
sULLw11UumpSRVRYshkRTFGt]daawFrxSx0VmRStUdWV1Ra2o==", decode("base64")
out[3]: "VlWV1ZWTdtaamXUjJKNFZURTVsbUp0VFhKVJFw)FUMVkiU21NeE9WRphazR3WkVoc1p
sULLw11UumpSRVRYshkRTFGt]daawFrxSx0VmRStUdWV1Ra2o==", decode("base64")
out[4]: "VUVVVNWSjdwR24yTERnJTxdaREZ1T1YS5Mx0Vfjak4wZEhsZLRYVRTR]eTUCxdE1
FNWZlaikzwdrNGWtjk=", decode("base64")
out[5]: "UEVUSVJ7VGgxU19FbmNkZ0Fu0V9jci19Q;jN0dHlFTXVDSF9DMg1tME5fbjB3YwQ0eVn9", decode("base64")
out[5]: "PETIR{Th15_Encd1n9_Is_Pratty_MuCh_CommN_n@wad4$}"
In [6]: [Desktop]
[Trash]
[REVERSE]
NETWORK
Browse Network
"Document.docx" (11,9 kB) Microsoft Word Document

```

Hash Lookup (328 pts)

Diberikan sebuah soal yang berisi “hash md5” dan diberikan perintah untuk menjadikan text tersebut menjadi plaintext agar menemukan flagnya.

Petir Qualification Notice Teams

Challenge 61 Solves

Hash lookup

328

Flag adalah plaintext dari hash MD5
5f4dcc3b5aa765d61d837deb882cf99

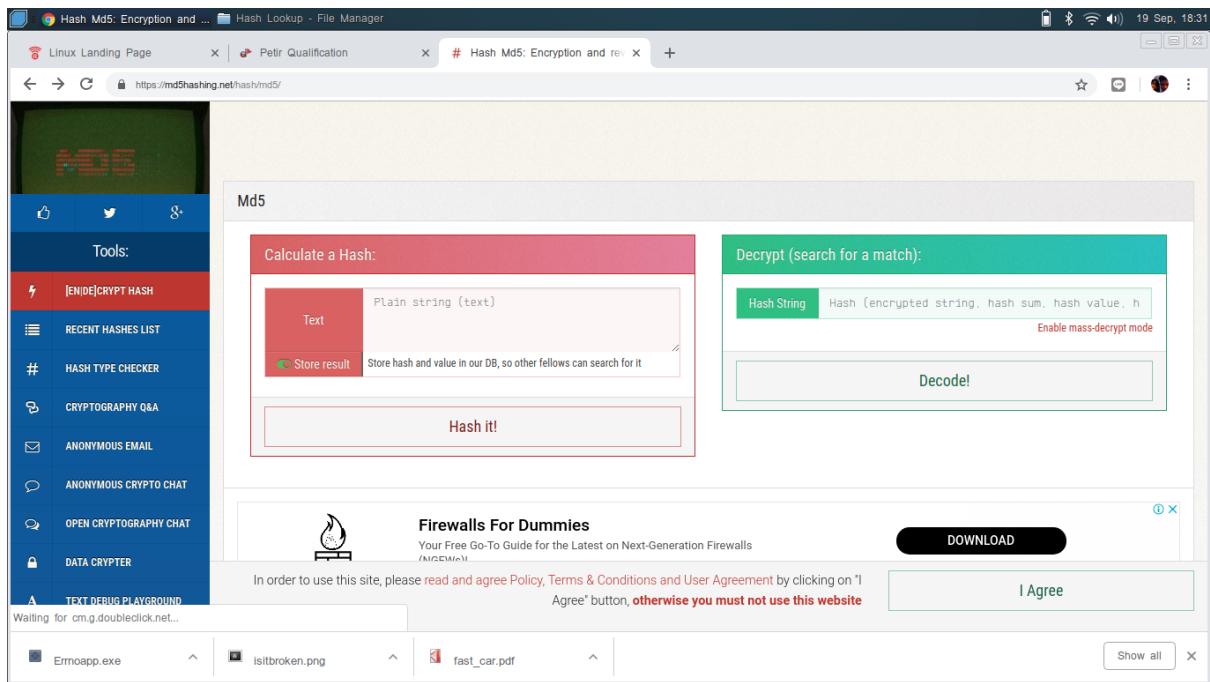
Flag Submit

Crypto

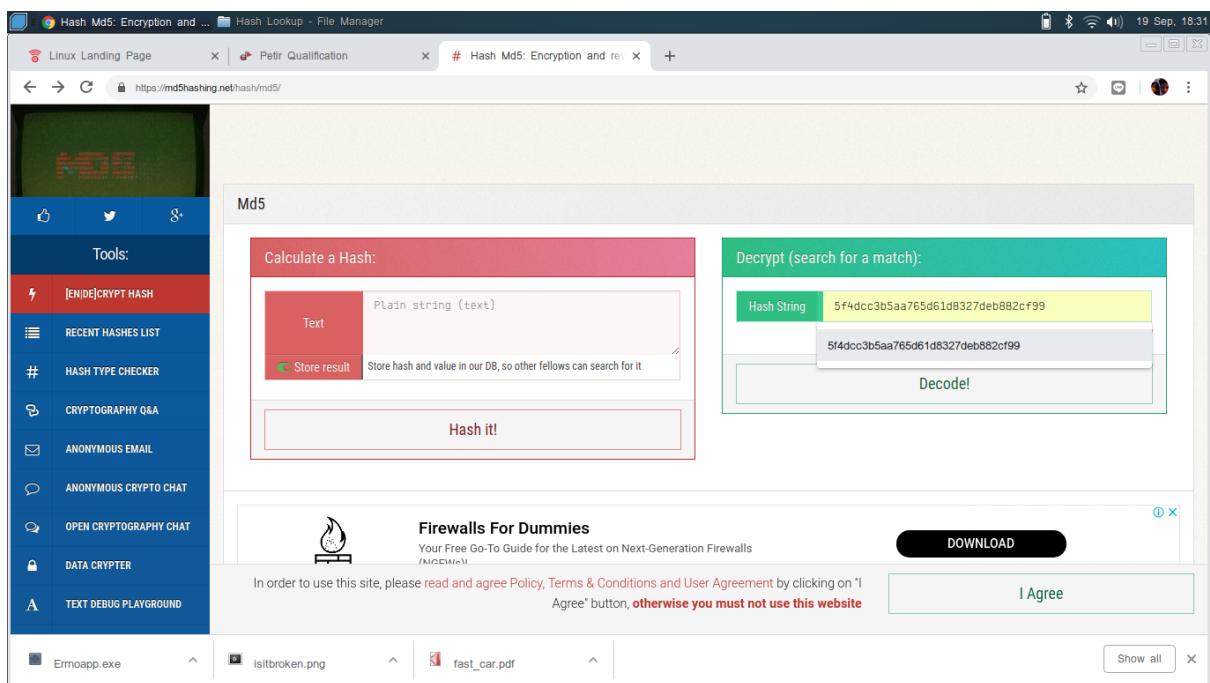
Hashing	Vigenere	Hashcrack	DES
310	439	477	481
LCG	RSA		

secretplace 500

Setelah saya mengetahui bahwa itu adalah “hash md5” saya mencari di google deryptor untuk mendecrypt dari hash md5 menjadi plaintext. Lalu saya menemukan website <http://md5hashing.net/hash/md5/>



Lalu saya mengopy teks tersebut dan memasukannya dalam kolom decode.



Setelah saya decode saya akhirnya menemukan plain teks dari “hash md5” tersebut dan itu adalah flagnya.

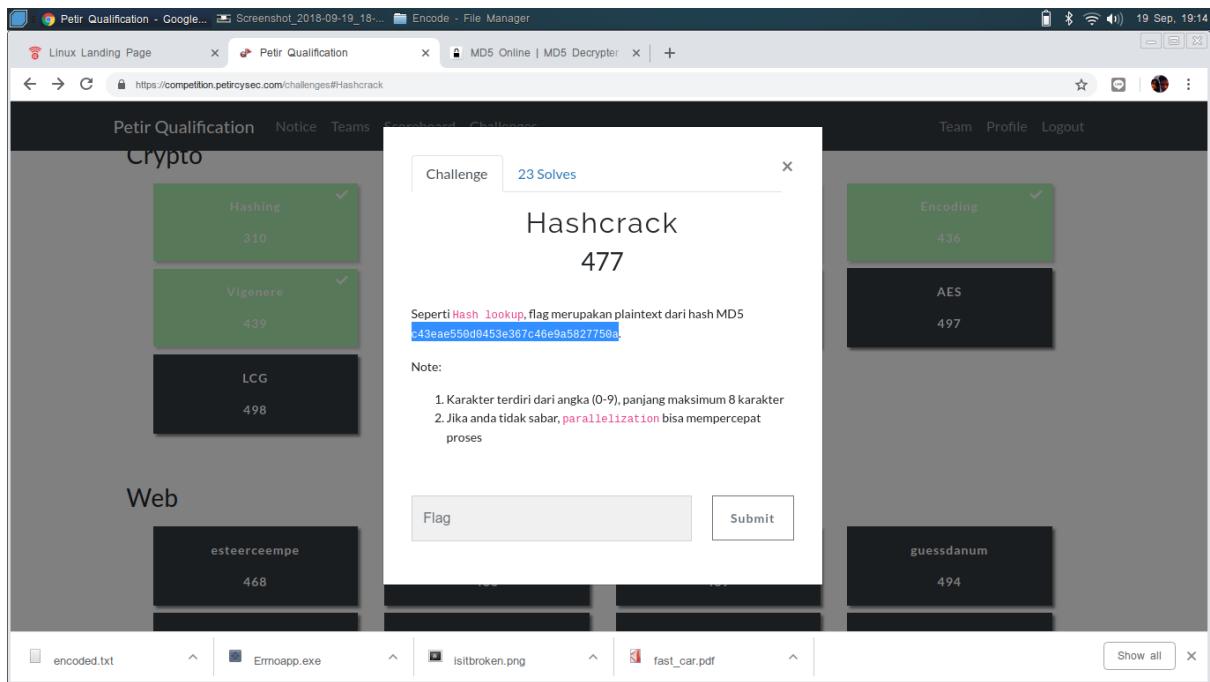
The screenshot shows a web browser window with the URL <https://mdHashing.net/has/hMd5/5f4dcc3b5aa765d61d8327deb882cf99>. The page displays a 'Decoded value:' field containing 'password' and an 'Original Hash (Md5):' field containing '5f4dcc3b5aa765d61d8327deb882cf99'. A sidebar on the left lists various tools, including 'EN/DECRYPT HASH', 'RECENT HASHES LIST', 'HASH TYPE CHECKER', 'CRYPTOGRAPHY Q&A', 'ANONYMOUS EMAIL', 'ANONYMOUS CRYPTO CHAT', 'OPEN CRYPTOGRAPHY CHAT', 'DATA CRYPTER', and 'TEXT DEBUG PLAYGROUND'. A message at the bottom states: 'In order to use this site, please read and agree Policy, Terms & Conditions and User Agreement by clicking on "I Agree" button, otherwise you must not use this website.' A green button labeled 'I Agree' is visible.

Yaitu “password”. Lalu saya kembali kedalam soalnya dan saya beri tambahan didepan PETIR{password}.

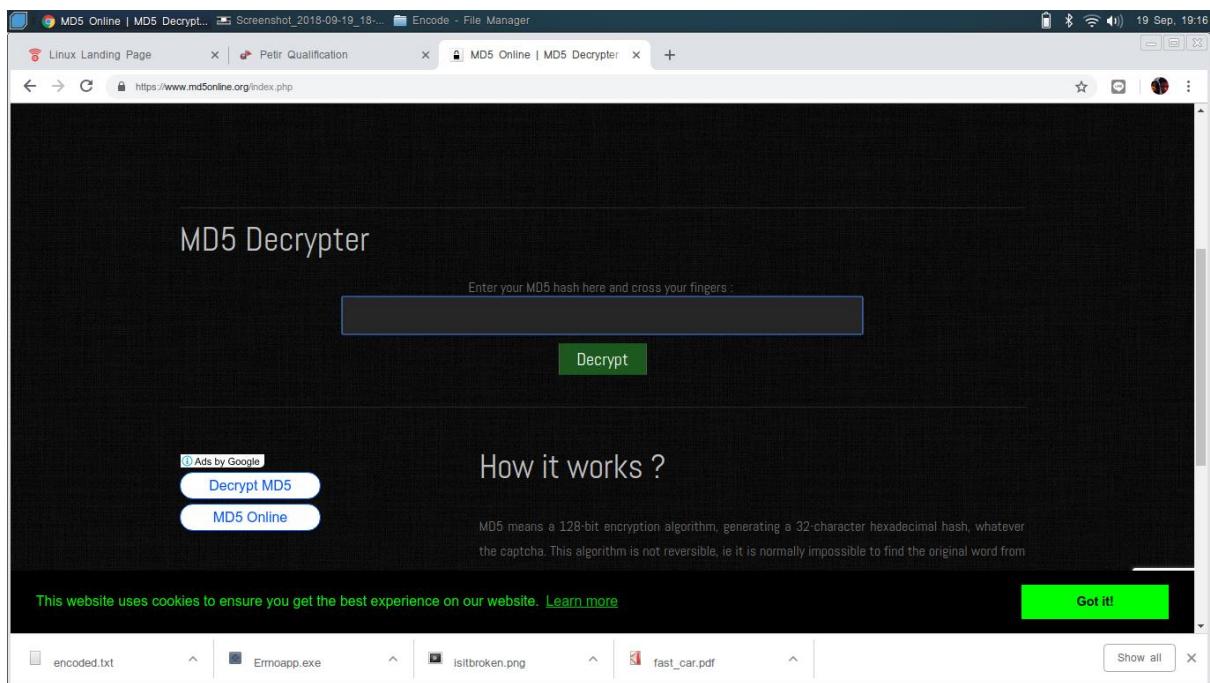
The screenshot shows a web browser window with the URL <https://competition.petircysec.com/challenges/#Hash%20lookup>. The page displays a 'Challenge' modal titled 'Hash lookup' with the number '328'. Inside the modal, it says 'Flag adalah plaintext dari hash MD5' and shows the hash '5f4dcc3b5aa765d61d8327deb882cf99'. Below this is a text input field containing 'PETIR{password}' and a 'Submit' button. The background shows a grid of challenges: 'Ar33thmatic' (500), 'slow somach' (500), 'secretplace' (500), 'Encoding' (436), 'AES' (497), 'Hashcrack' (477), 'DES' (481), 'RSA' (430), 'Vigenere' (439), 'LCG' (310), and 'Hashing' (310). A sidebar on the left lists 'Petir Qualification', 'Notice', and 'Teams'. A message at the bottom states: 'In order to use this site, please read and agree Policy, Terms & Conditions and User Agreement by clicking on "I Agree" button, otherwise you must not use this website.' A green button labeled 'I Agree' is visible.

Hashcrack (477 pts)

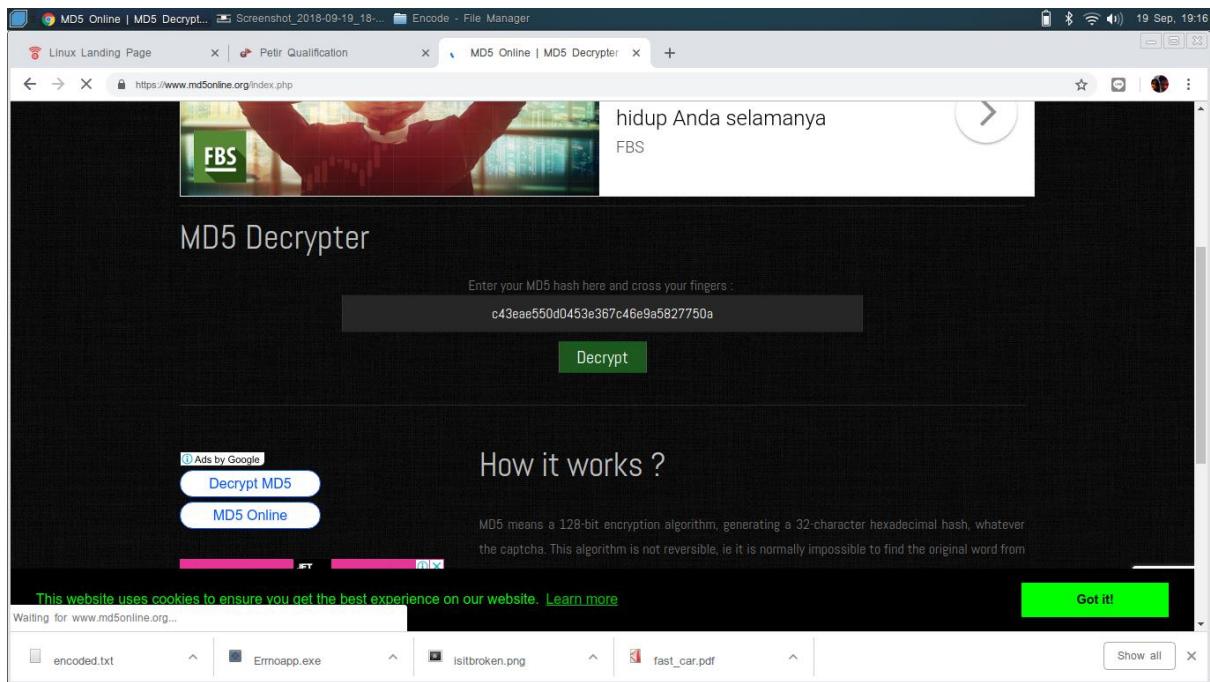
Sama seperti Hash Lookup, diberi soal adalah kode “hash md5” yang harus di decrypt menjadi plain text untuk menemukan flagnya.



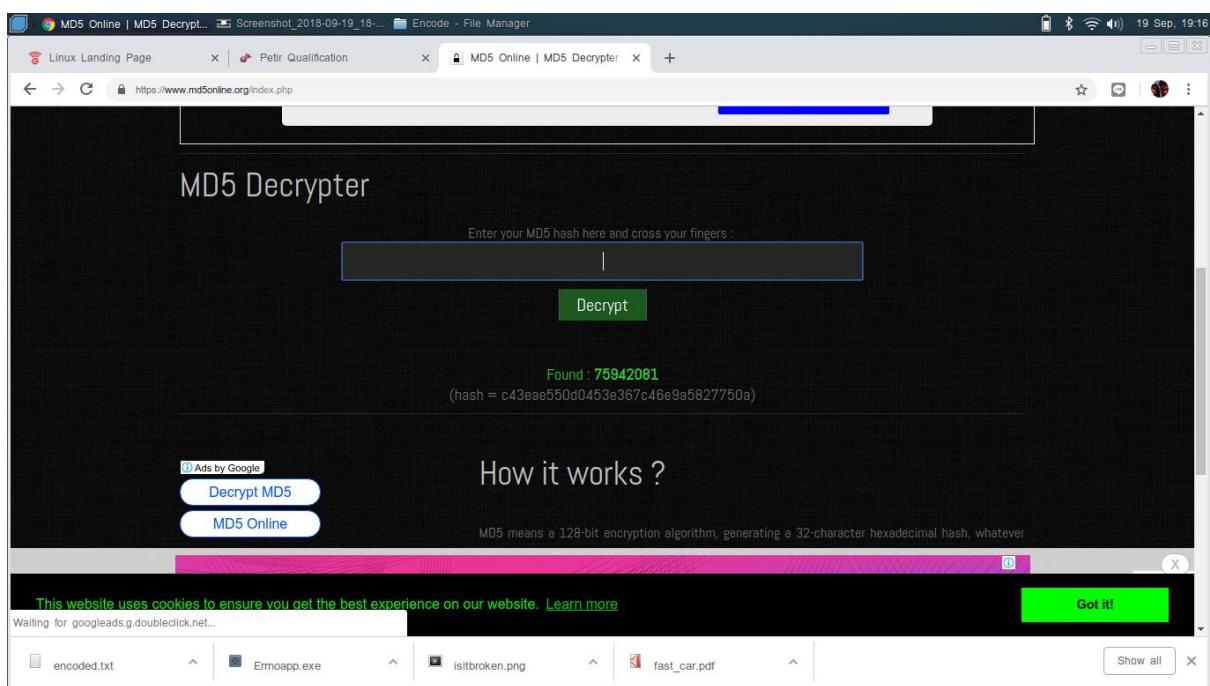
Setelah mengerti saya mencari di web browser / google, website untuk mendecrypt nya dan akhirnya saya mendapatkan website <http://md5online.org/index.php>



Lalu setelah itu saya copy kode tersebut dan paste pada kolom decrypt tersebut



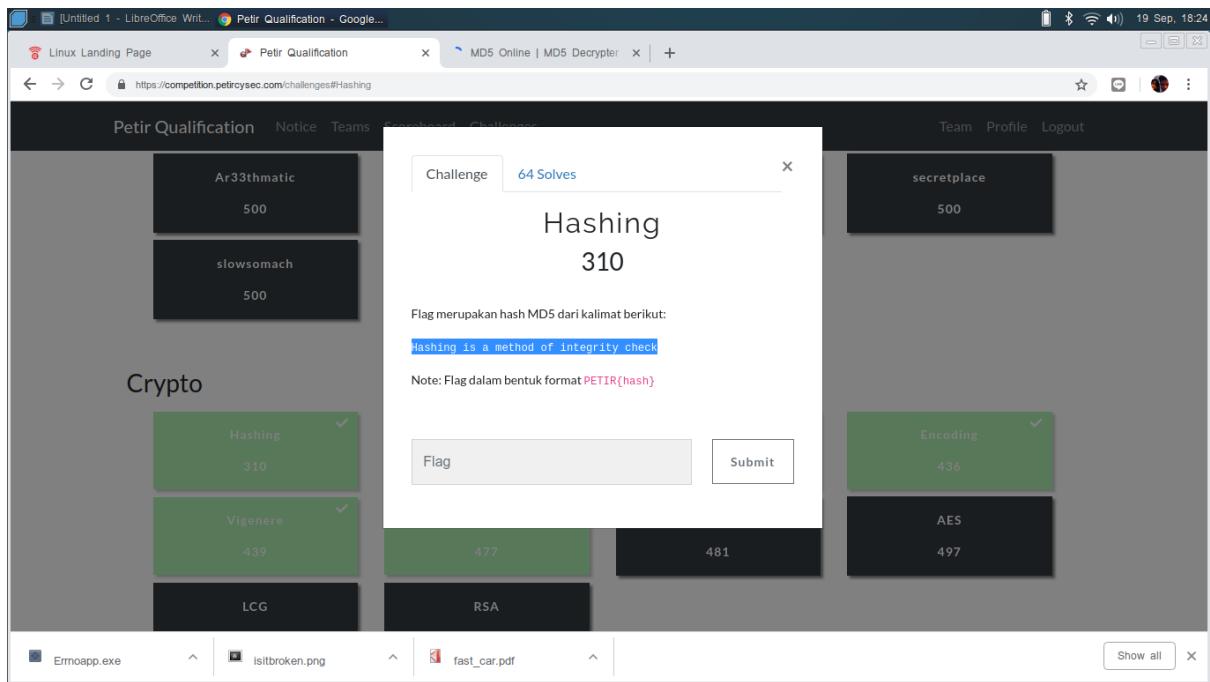
Setelah itu saya klik decrypt dan akhirnya keluar hasil plain teks dari “hash md5” tersebut dan itu adalah flagnya.



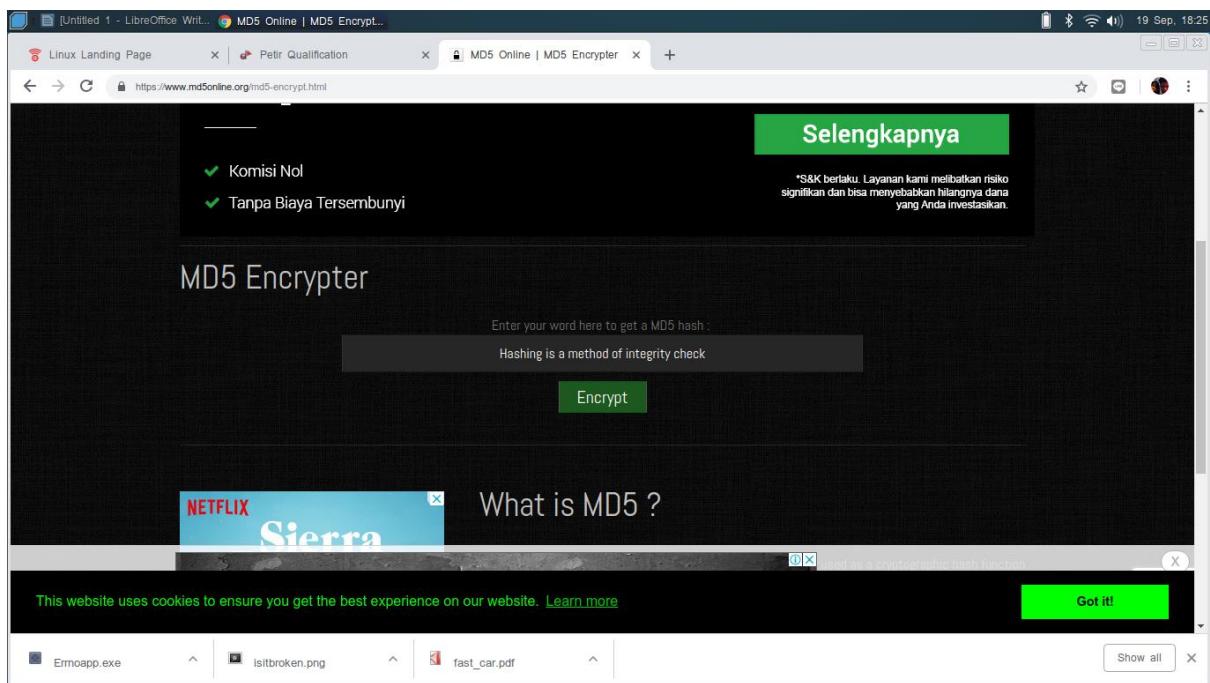
Dan kembali kedalam website soal saya beri tambahan pada bagian depan yaitu PETIR{75942081} dan itu adalah flagnya.

Hashing (310 pts)

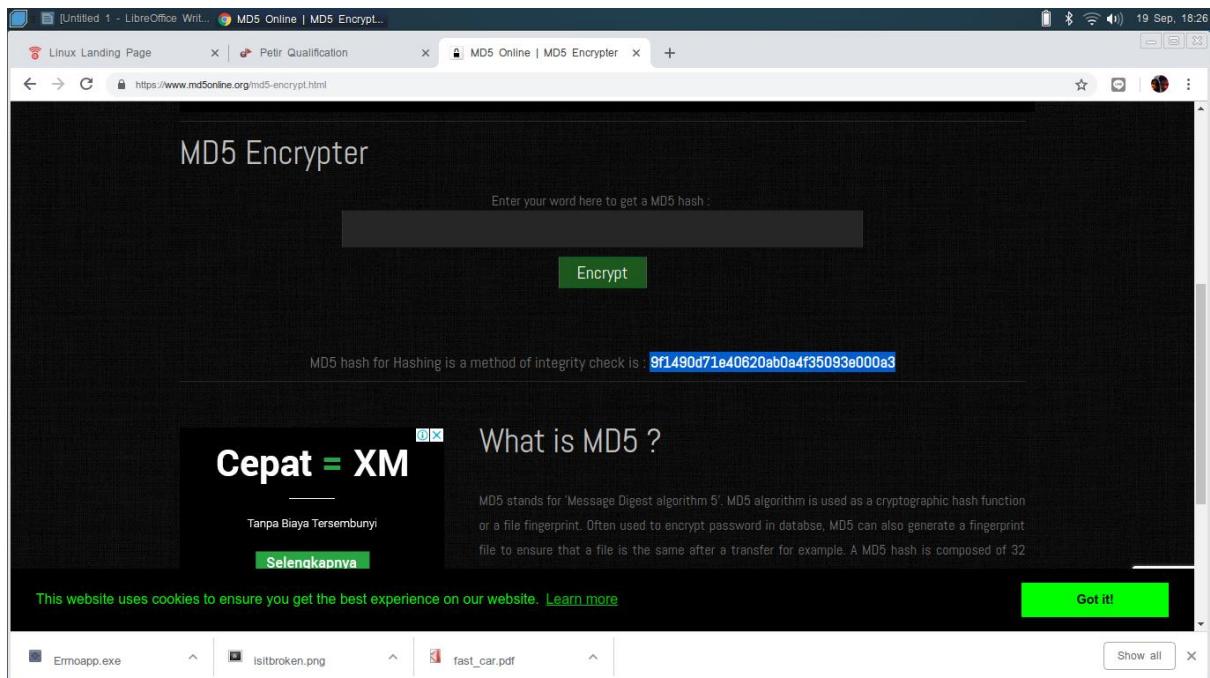
Diberi soal plain teks yang diperintahkan untuk merubahnya menjadi “hash md5”.



Setelah mengetahui perintah tersebut, saya langsung mencari dalam web browser / google untuk encrypt menjadi “hash md5” dan saya menemukan website <http://md5online.org/md5-encrypt.html>



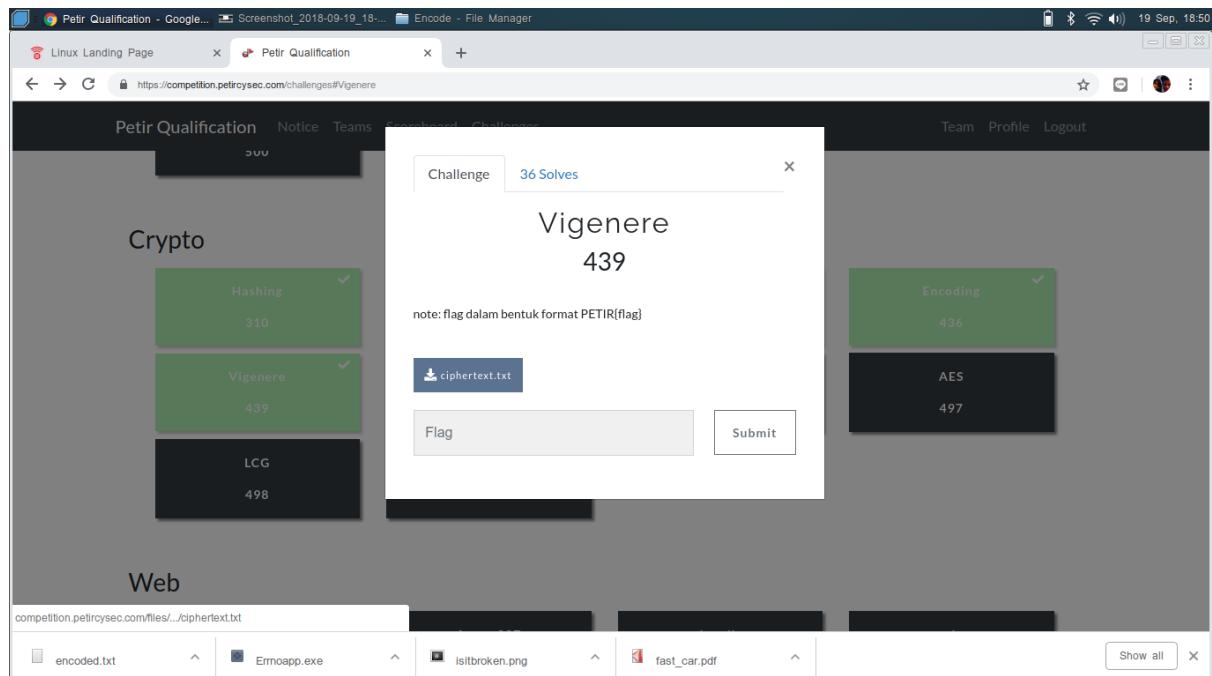
Lalu saya mengopy plain teks tersebut dan paste dalam kolom tersebut. Lalu saya klik encrypt dan keluar teks enkripsi dari plain teks tersebut menjadi “hash md5” dan itu adalah flagnya.



Setelah itu kembali ke halaman soal dan diberi tambahan diawal PETIR{hasil enkripsi hash md5} lalu itu adalah flagnya.

Vigenere (439 pts)

Diberi soal berupa chiper text vinegeree yang harus didekripsi menjadi plain teks tanpa mengetahui keynya.



Lalu setelah saya tau bahwa ini adalah kode vigenere lalu saya langsung cari vigenere tools yang ada di google dan saya menemukan tools yang dapat menemukan keynya.

The screenshot shows the Cryptii web application interface for the Vigenère cipher. In the 'Plaintext' section, the input text is "The quick brown fox jumps over 13 lazy dogs.". In the 'KEY' section, the key is set to "cryptii". The 'VARIANT' is set to "Standard". The 'ALPHABET' is set to "abcdefghijklmnopqrstuvwxyz". The 'KEY MODE' is set to "Repeat". The 'CASE SENSITIVITY' is set to "Yes". The 'FOREIGN CHARS' dropdown has "Include" selected. The 'Encode' button is highlighted in orange. The 'Ciphertext' section shows the output: "vyc fnqkm spdpv nqo hjfxa qmcg 13 eiha umvl.". Below the ciphertext, a message indicates it was encoded in 0.29ms.

Saya copy kode vigenere pada bagian “ciphertext” lalu itu akan otomatis ke dekripsi menjadi plaintext dan langsung mendapatkan keynya.

The screenshot shows the Cryptii web application interface for the Vigenère cipher. In the 'Ciphertext' section, the input text is "vyc fnqkm spdpv nqo hjfxa qmcg 13 eiha umvl.". In the 'KEY' section, the key is set to "cryptii". The 'VARIANT' is set to "Standard". The 'ALPHABET' is set to "abcdefghijklmnopqrstuvwxyz". The 'KEY MODE' is set to "Repeat". The 'CASE SENSITIVITY' is set to "Yes". The 'FOREIGN CHARS' dropdown has "Include" selected. The 'Decode' button is highlighted in orange. The 'Plaintext' section shows the output: "The quick brown fox jumps over 13 lazy dogs.". Below the plaintext, a message indicates it was decoded in 0.8ms.

Atau di tools lain juga mungkin dapat menemukan keynya atau dapat juga dengan cara diitung (http://apri-yulianto.blogspot.com/2012/12/pehitungan-sandi-vigenere_24.html).

Lalu saya menemukan flagnya.

The screenshot shows the Cryptii web application interface for a Vigenère cipher. On the left, under 'Plaintext', there is a text area containing the following message:

```
there was a tale of man who
tried to use polyalphabetic
cipher but it was too weak
against metal demon. he tried
every key but it wasn't even a
match. the flag is
thispolyalphabeticciphertextisstillw
eak
```

On the right, under 'Ciphertext', there is a text area containing the following encoded message:

```
Momjm onl a dejx vn eif jao
dvgxk bg ckr iovcyewpsjwgbc
mmnalz tcl vn wk w hv ewic
nzasrqm tmld qxmyr. Fx azamv
roebc ix f jmb ag pacr'r xcmf i
enmcr. Xfx mtso af
MhswnhsgSthUtBoXgVjqHPwEbsCxgesE
wIC
```

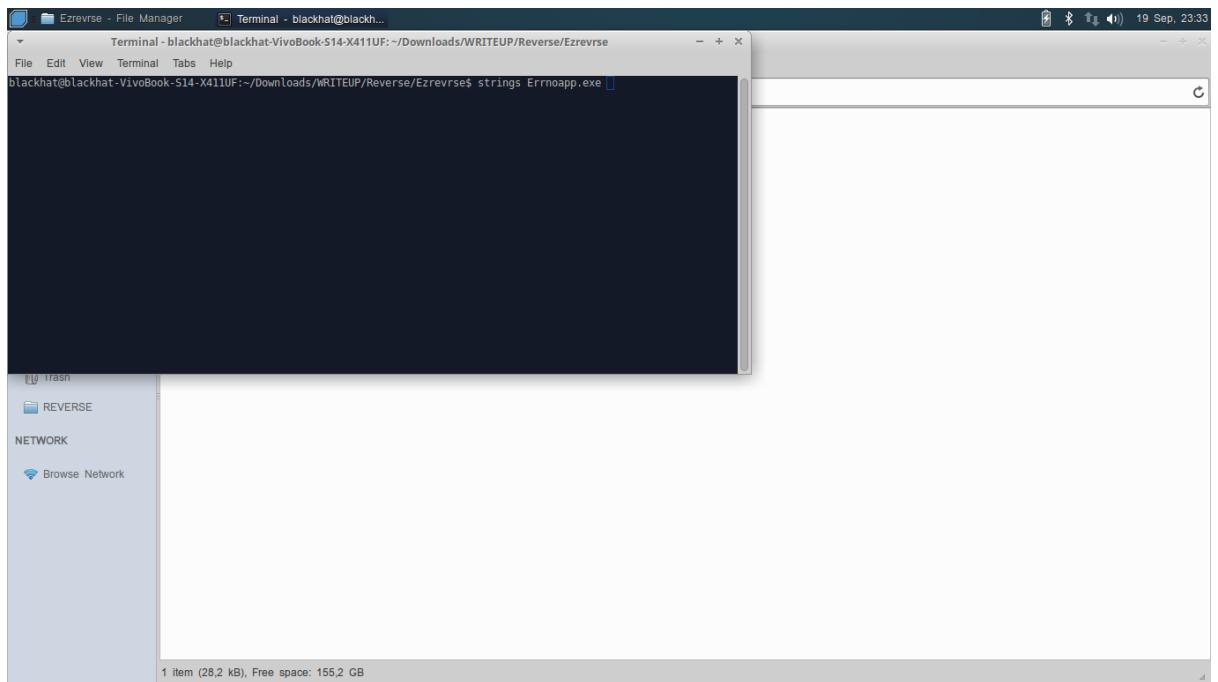
The central part of the interface is the 'Vigenère cipher' configuration section, which includes fields for 'VARIANT' (Standard), 'KEY' (thisisntakey), 'KEY MODE' (Repeat), 'ALPHABET' (abcdefghijklmnopqrstuvwxyz), 'CASE SENSITIVITY' (Yes No), and 'FOREIGN CHARS' (Include Ignore). Below this section, a note indicates: '← Decoded 198 chars in 0.49ms'.

Kembali dalam halaman soal dan menambahkan PETIR{<hasil dekripsi>}.

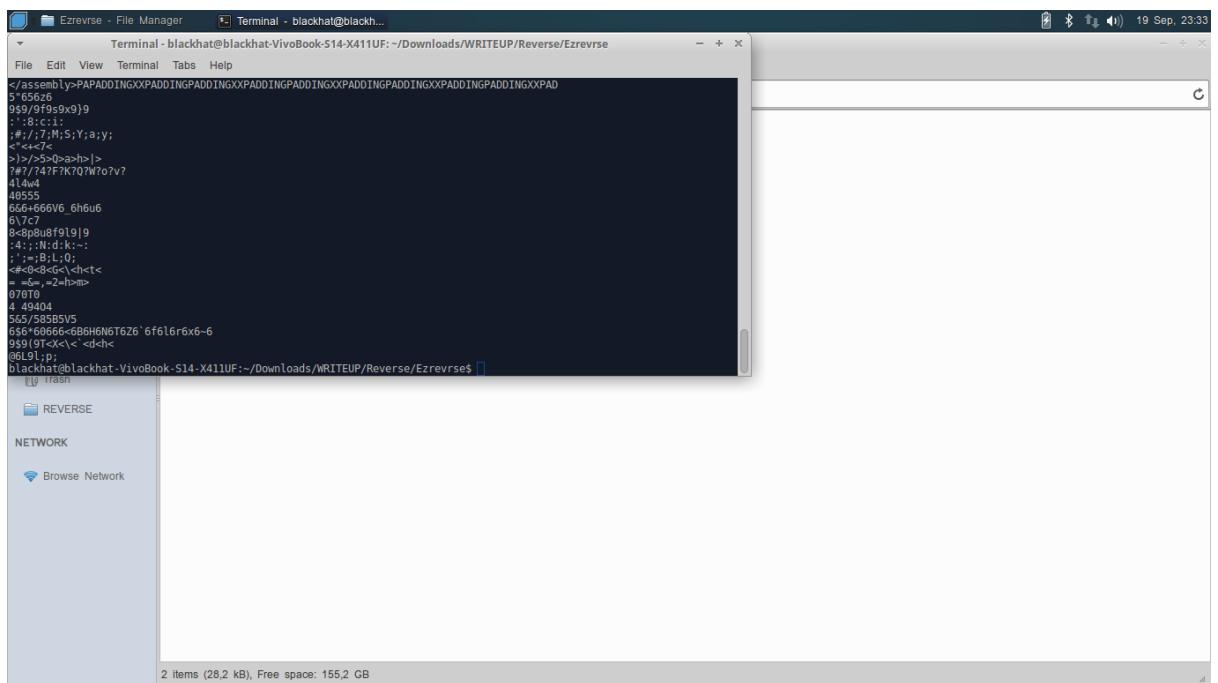
REVERSE

Ezrevrse (472 pts)

Diberi soal program Errnoapp.exe yang dapat dijalankan, tetapi saya pertama – tama yang saya lakukan adalah mencari strings yang ada dalam app tersebut dengan membuka terminal didalam *directory* tersebut.



Lalu saya ketik fungsi “strings” untuk mengetahui semua strings dalam file tersebut dengan cara “strings_<file>” lalu enter untuk menjalankan.



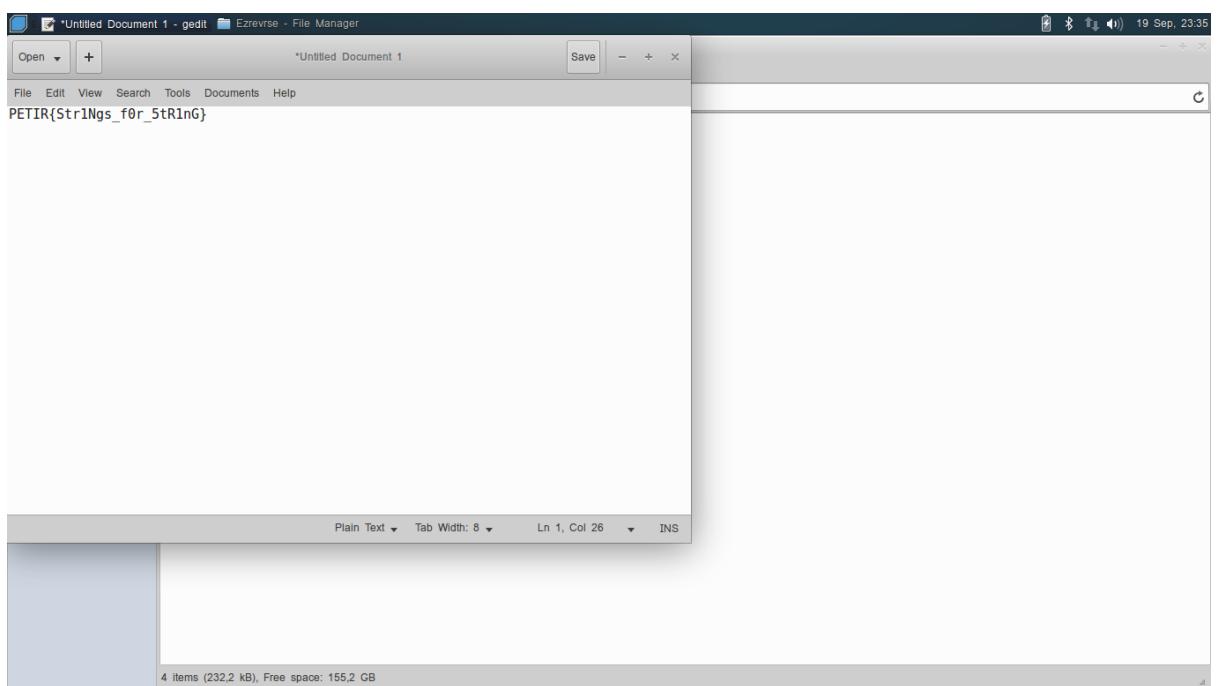
Setelah keluar semua strings saya mencari apakah ada flag didalam sini dan ternyata saya menemukan flagnya.

The screenshot shows a Linux desktop environment. In the top right corner, there is a system tray with icons for battery, signal strength, and volume. The date and time '19 Sep, 23:35' are also displayed. Below the tray, there is a terminal window titled 'Terminal - blackhat@blackhat-VivoBook-S14-X411UF: ~/Downloads/WRITEUP/Reverse/Ezrevrse'. The terminal output is as follows:

```
RhtsA
P_VVV
tR+
>rK
=MuA
hTaA
hbaA
hDA
hbaA
t4h0bA
Flag is Str1NgS_f0r_5tR1nG
Stack around 'alloca' corrupted
Local variable used before initialization
Stack memory corruption
Cast to smaller type causing loss of data
Stack pointer optimization
Stack Runtime Check Error
Stack memory around 'alloca' was corrupted
A local variable was used before it was initialized
Stack memory was corrupted
A cast to a smaller data type has caused a loss of data. If this was intentional, you should mask the source of the cast with the appropriate bitmask. For example:
char c = (i & 0xFF);
Changing the code in this way will not affect the quality of the resulting optimized code.
```

To the left of the terminal is a file manager window titled 'Ezrevrse - File Manager'. The sidebar on the left lists 'REVERSE' and 'NETWORK'. Under 'REVERSE', there is a folder icon labeled 'REVERSE'. Under 'NETWORK', there is a Wi-Fi icon labeled 'Browse Network'. At the bottom of the file manager window, it says '3 items (75,9 kB), Free space: 155,2 GB'.

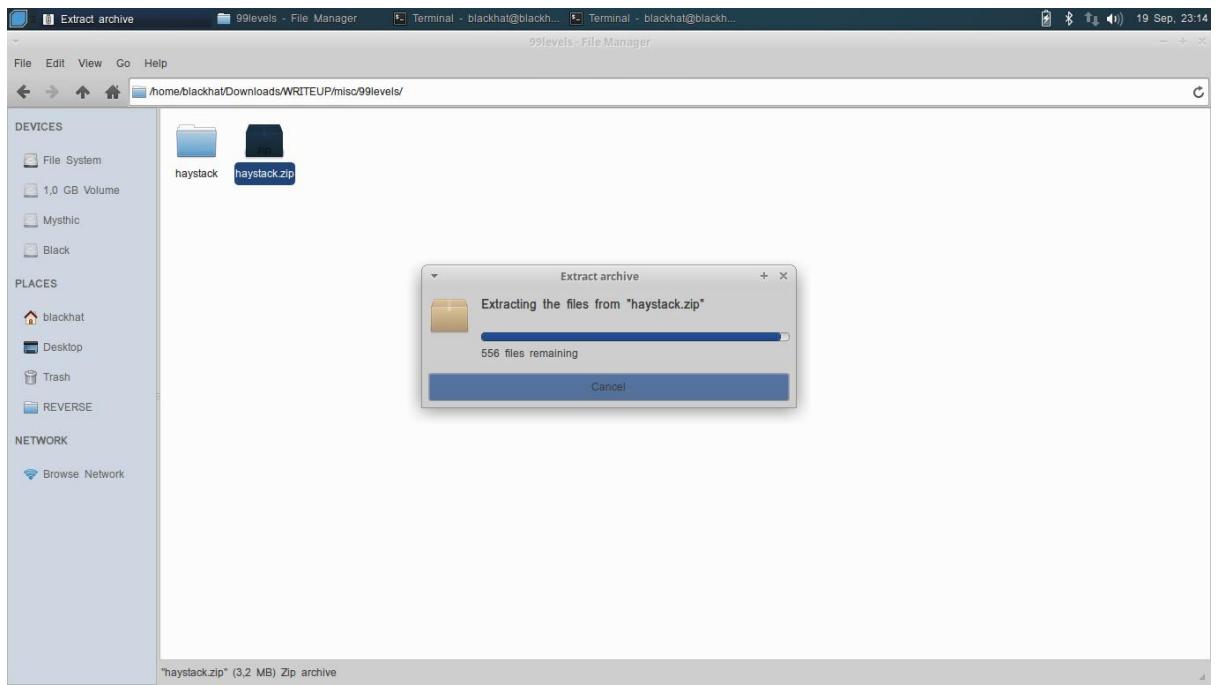
Saya langsung kembali ke dalam website soal dan ditambahkan didepannya dengan PETIR{<flag>}.



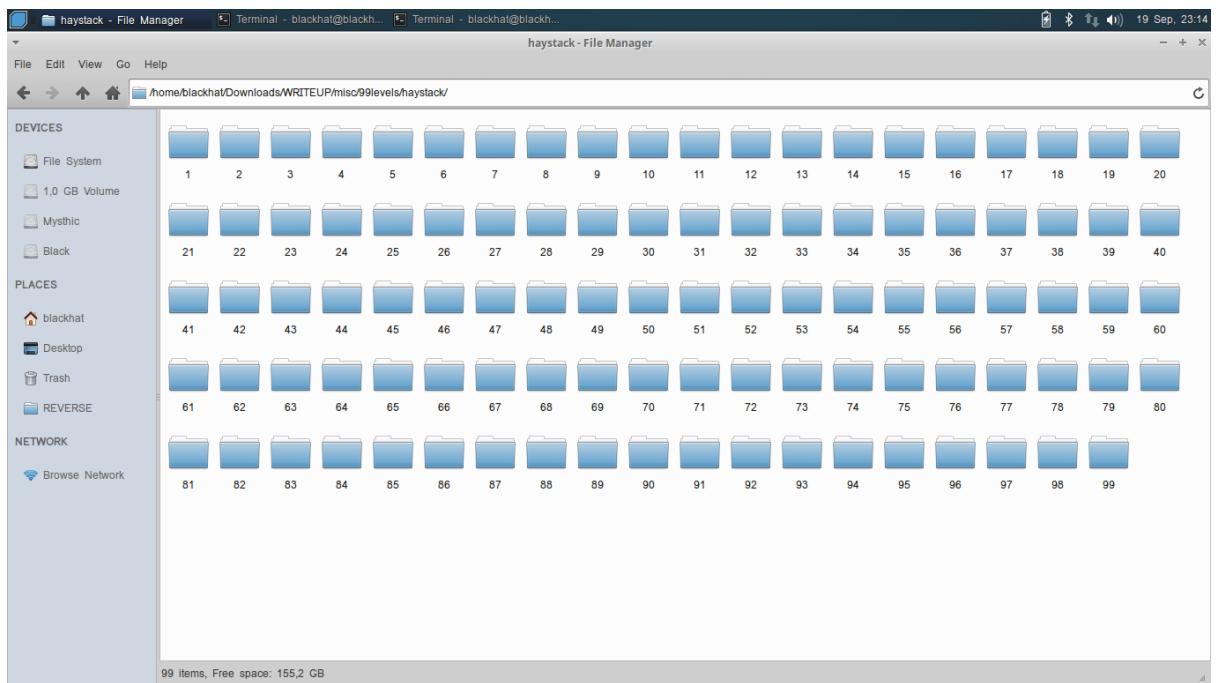
MISC

99levels (392 pts)

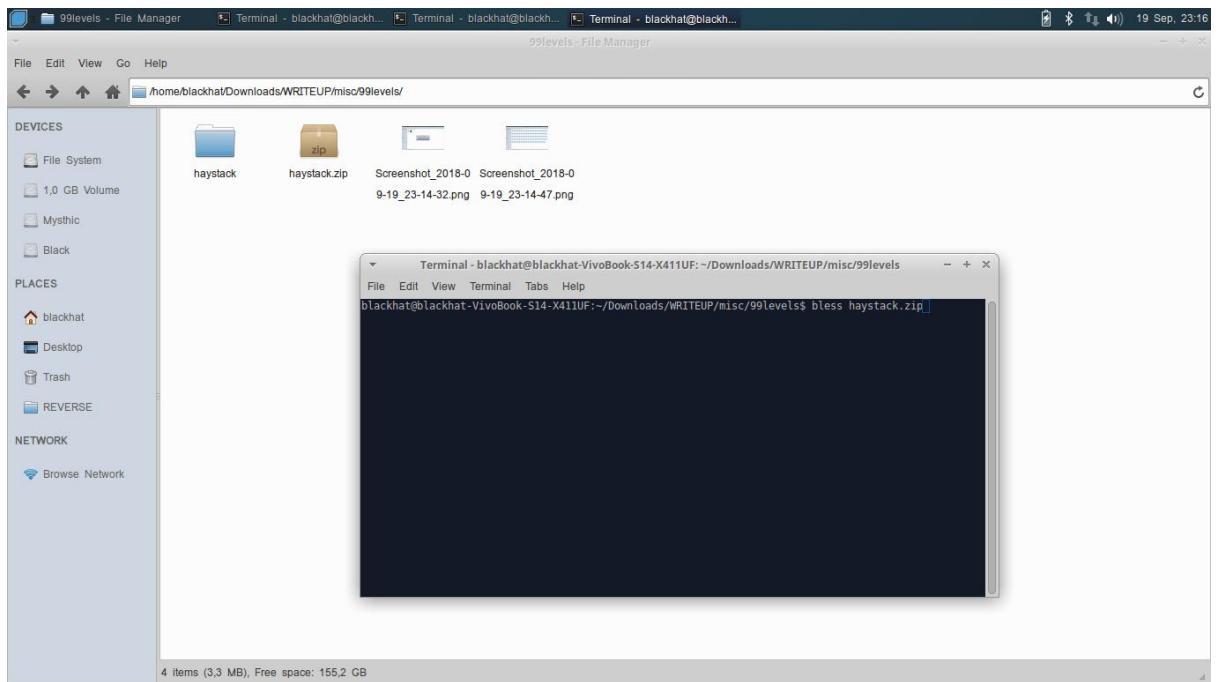
Diberi soal sebuah file .zip, lalu saya ekstraks file tersebut.



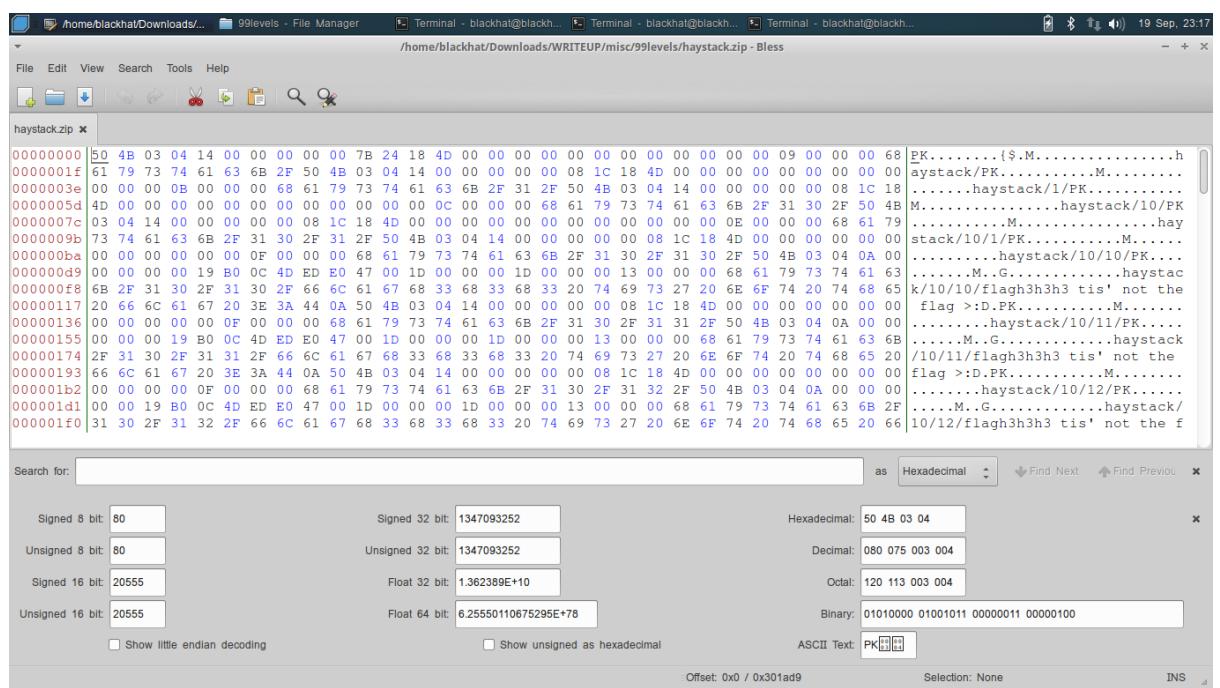
Setelah saya ekstraks saya menemukan banyak folder didalam hasil ekstraks tersebut.



Karena terlalu banyak file dan jika satu per satu dibuka akan memakan banyak waktu, saya akhirnya memutuskan untuk membuka aplikasi "bless" pada file .zip untuk membantu saya dalam mencari file.



Lalu setelah saya membuka aplikasi tersebut saya menggunakan CTRL + F untuk mencari flag nya.



Saya mencari teks PETIR lalu akhirnya saya menemukan file tersebut.

The screenshot shows the Immunity Debugger interface with several windows open:

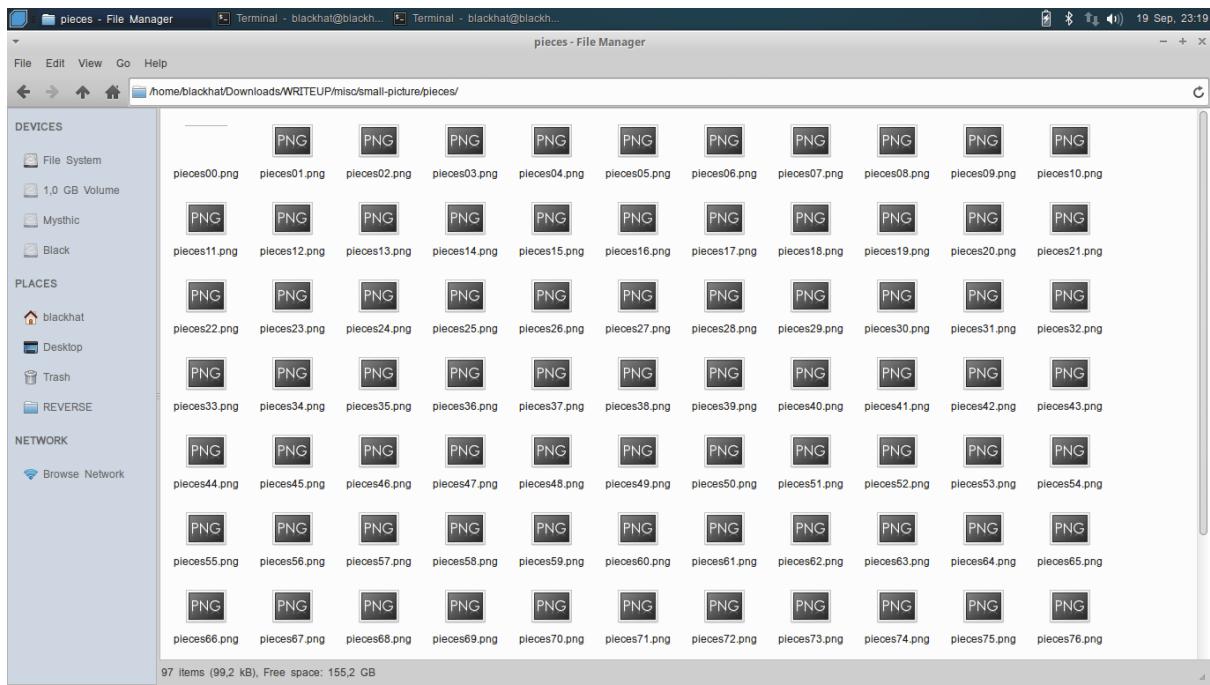
- File Manager:** Shows the path /home/blackhat/Downloads/... and file 99levels - File Manager.
- Terminal:** Three terminals are visible, all titled blackhat@blackh... with different command-line prompts.
- Assembly Dump:** The main pane displays assembly code for the file haystack.zip. A search result for "PETIR" is highlighted at offset 0x00000000.
- Registers:** Shows CPU registers including RAX, RBX, RCX, etc., with their current values.
- Stack Dump:** Shows the state of the stack across multiple frames.
- Search Results:** A search results pane for "PETIR" containing 10 entries, each with a hex dump, assembly, and ASCII representation.
- Registers/Stack Dump:** A pane showing signed/unsigned 8-bit, 16-bit, and 32-bit values along with their decimal, octal, and binary representations.

Dan akhirnya saya mendapatkan flagnya.

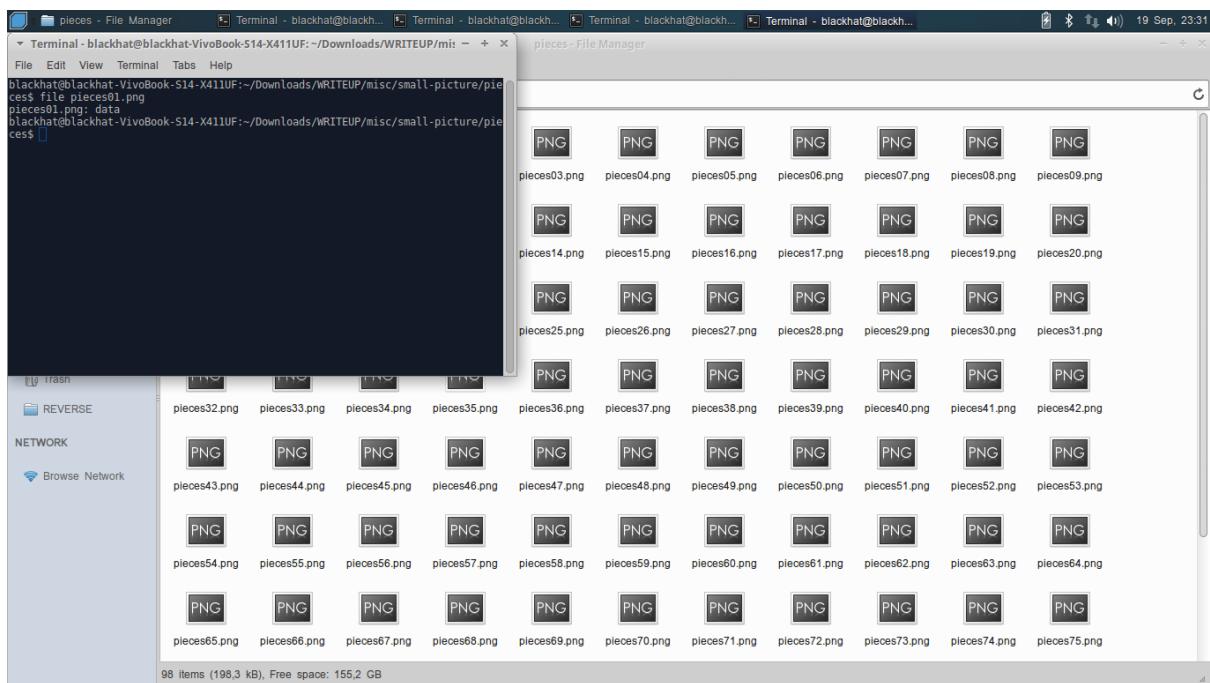
Atau bisa juga menggunakan fungsi “cat” pada terminal linux untuk membuka setiap file tanpa harus membukanya satu per satu.

small-picture (487 pts)

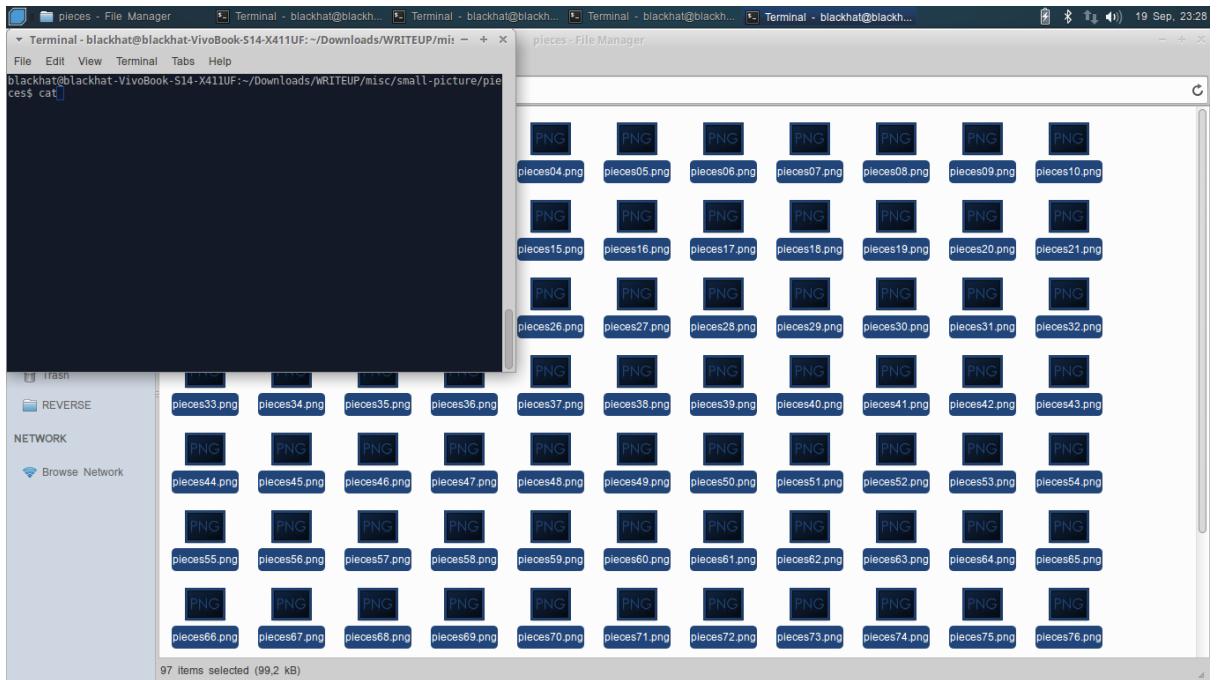
Diberi soal .zip yang saya ekstraks dan menghasilkan banyak file png didalamnya.



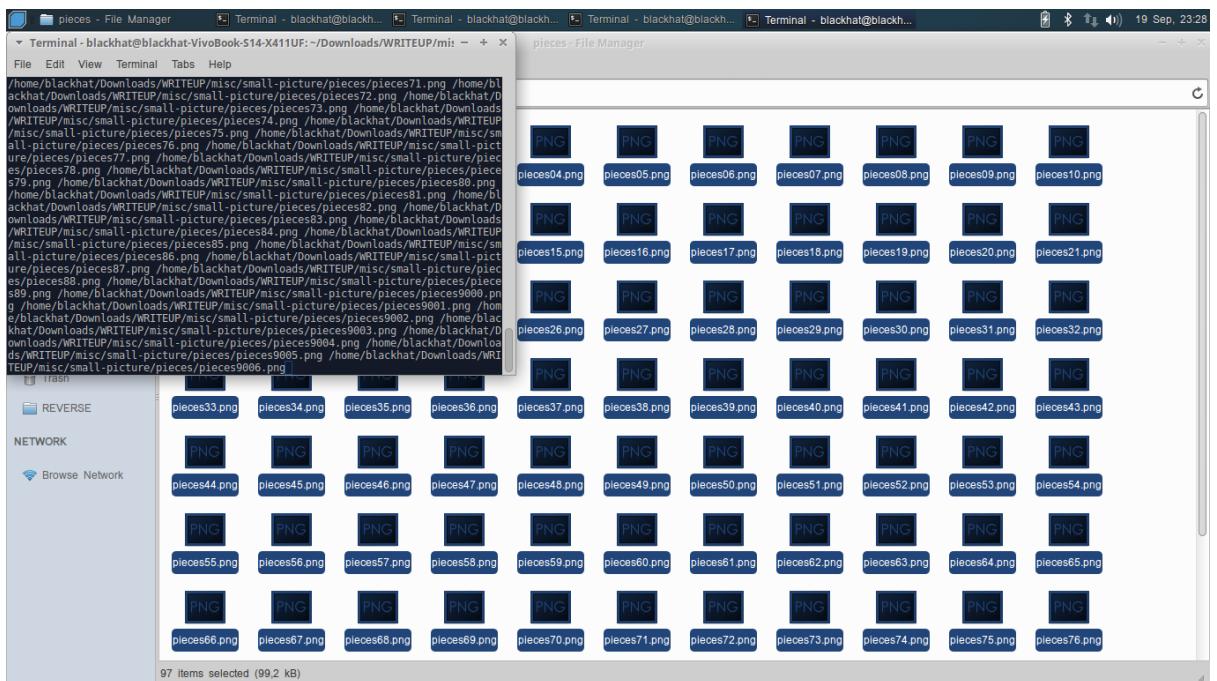
Lalu setelah saya perhatikan kenapa hanya 1 file saja yang ada previewnya sedangkan yang lain tidak ada. Lalu saya mencari tau dengan cara membuka terminal dan mengidentifikasi salah satu file dengan fungsi “file” dalam terminal linux.



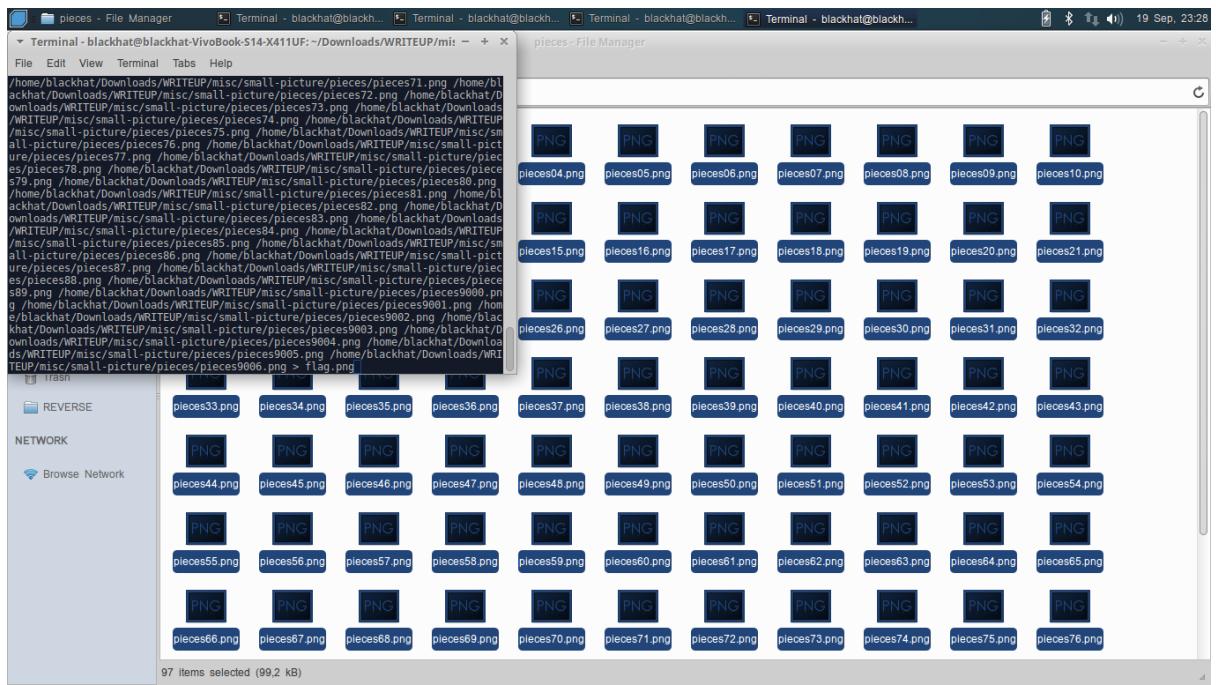
Dan ternyata hanya satu file yang benar – benar PNG dan sisanya hanyalah sebuah data lalu saya berpikir dan mencoba untuk menjadikan satu semua file dengan fungsi “cat” pada terminal linux.



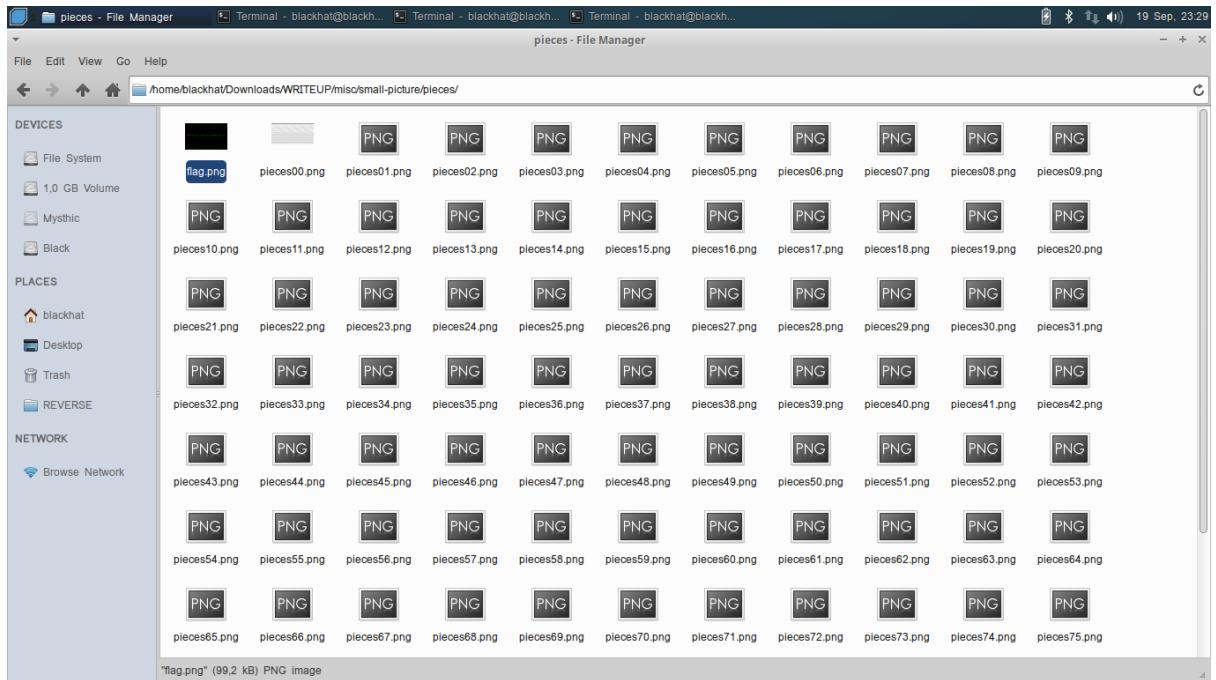
Dengan cara ketik “cat <tarik semua file ke terminal>”



Lalu diberi panah untuk menjadikan satu menjadi file apa seperti contoh saya menggabungkan menjadi file bernama “flag.png”



Lalu klik enter maka semua file menjadi satu file.



Ternyata file tersebut adalah flag nya yang berupa gambar PNG.

