LAPORAN GEMASTIK 8 - KEAMANAN JARINGAN DAN SISTEM INFORMASI

PENYISIHAN 2

16 September 2015

Tim : CSI_Oryza
Univ : Institut Pertanian Bogor
Anggota : Herdian Nugraha
Fachrizal Oktavian
Gusti Bimo Marlawanto

Summary

Pada babak penyisihan 2 ini kami mendapatkan beberapa challenge terkait keamanan informasi, forensik digital, dan keamanan web. Empat dari lima challenge (A,B,C,E) berhasil kami selesaikan.

Nomor Soal	Jawaban	Waktu
A	`pemuda`	1:20:48
B	`kamiagenperubahan`	1:47:09
C	`2810192828102015`	0:37:01
D	-	-
E	`rudnumgnatnapsuretujam`	0:50:42

[SOLVED]

Soal A : You Never Know If You Never Try

Unduh file berikut dan kemudian selesaikan tantangan yang terdapat dalam file tersebut.

Challenge

Kumpulkan jawaban dalam sebuah source code Python berekstensi (.py). Dua baris kode adalah template yang akan ditambahkan dengan jawaban problem ini.

```
soal1 = raw_input('')

print('put your answer here')  #Tuliskan jawaban diantara tanda petik
```

File challenge:

Challenge

Analisa file:

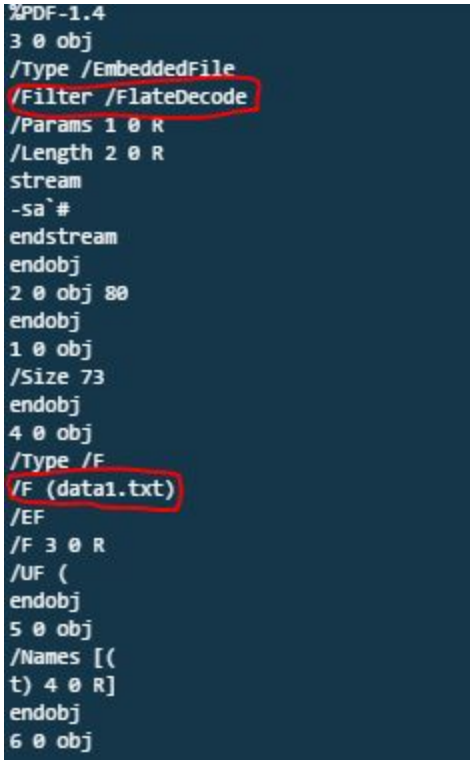
File tersebut tidak diberi ekstensi. Pertama kami kira file tersebut merupakan binary executable, lalu kami lakukan deteksi ekstensi menggunakan **file** , setelah mengetahui ekstensi file tersebut yang merupakan **Zip archive**. Kami lakukan unzip file tersebut, ternyata di dalamnya terdapat 5 file pdf.

```
hrdn@hacking:~/workspace/Gemastik8 $ ls
soal1
hrdn@hacking:~/workspace/Gemastik8 $ file soal1
soal1: Zip archive data, at least v1.0 to extract
hrdn@hacking:~/workspace/Gemastik8 $ unzip soal1
Archive:  soal1
   creating: Files/
  inflating: Files/file5.pdf
  inflating: Files/file4.pdf
  inflating: Files/file3.pdf
  inflating: Files/file2.pdf
  inflating: Files/file1.pdf
hrdn@hacking:~/workspace/Gemastik8 $ ls
Files/  soal1
hrdn@hacking:~/workspace/Gemastik8 $
```

Kami mencoba **foremost,** dan **binwalk** tetapi tidak menemukan sesuatu yang mencurigakan. Lalu kami lihat string yang ada pada file pdf (**file1.pdf**) dengan menggunakan **strings.**

```
hrdn@hacking:~/workspace/Gemastik8/Files $ strings file1.pdf | less
```

Hasilnya:



Terlihat file tersebut terkompresi karena ada **`/Filter /FlateDecode`**, dan pdf tersebut terembed file **data1.txt**. Cara selanjutnya yaitu **uncompress pdf**, yang pada kali ini kami menggunakan tools **qpdf.**

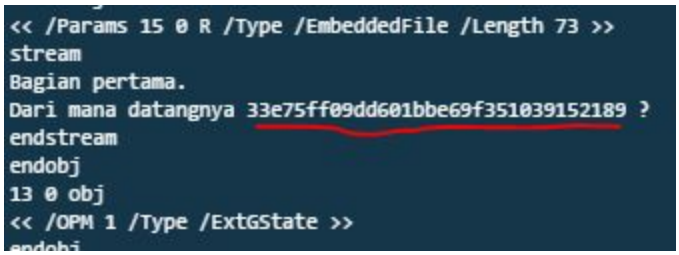
```
hrdn@hacking:~/workspace/Gemastik8/Files $ qpdf --stream-data=uncompress file1.pdf
file1.uncompressed.pdf
hrdn@hacking:~/workspace/Gemastik8/Files $ qpdf --stream-data=uncompress file2.pdf
file2.uncompressed.pdf
hrdn@hacking:~/workspace/Gemastik8/Files $ qpdf --stream-data=uncompress file3.pdf
file3.uncompressed.pdf
hrdn@hacking:~/workspace/Gemastik8/Files $ qpdf --stream-data=uncompress file4.pdf
file4.uncompressed.pdf
hrdn@hacking:~/workspace/Gemastik8/Files $ qpdf --stream-data=uncompress file5.pdf
file5.uncompressed.pdf
hrdn@hacking:~/workspace/Gemastik8/Files $ ls -lah
total 3.5M
drwxr-xr-x 2 ubuntu ubuntu 4.0K Sep 16 10:41 ./
drwxr-xr-x 3 ubuntu ubuntu 4.0K Sep 16 10:27 ../
```

```
-rw-r--r-- 1 ubuntu ubuntu 355K Sep 15 17:16 file1.pdf
-rw-r--r-- 1 ubuntu ubuntu 355K Sep 16 10:40 file1.uncompressed.pdf
-rw-r--r-- 1 ubuntu ubuntu 355K Sep 15 17:18 file2.pdf
-rw-r--r-- 1 ubuntu ubuntu 355K Sep 16 10:40 file2.uncompressed.pdf
-rw-r--r-- 1 ubuntu ubuntu 355K Sep 15 17:19 file3.pdf
-rw-r--r-- 1 ubuntu ubuntu 355K Sep 16 10:40 file3.uncompressed.pdf
-rw-r--r-- 1 ubuntu ubuntu 355K Sep 15 17:20 file4.pdf
-rw-r--r-- 1 ubuntu ubuntu 355K Sep 16 10:40 file4.uncompressed.pdf
-rw-r--r-- 1 ubuntu ubuntu 355K Sep 15 17:16 file5.pdf
-rw-r--r-- 1 ubuntu ubuntu 355K Sep 16 10:41 file5.uncompressed.pdf
```

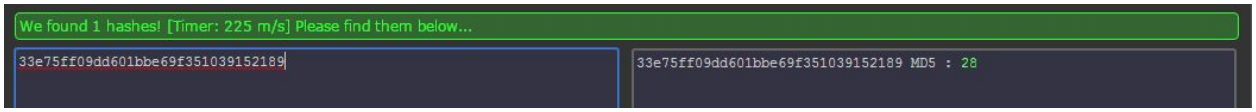
Setelah kami uncompress semua, kami analisa hasil uncompressnya dengan **strings.**

```
hrdn@hacking:~/workspace/Gemastik8/Files $ strings file1.uncompressed.pdf |
less
```

Terdapat suatu pesan dan suatu hash.



Panjangnya 32 karakter hexadecimal, sepertinya string hash **MD5.** Kami coba decrypt menggunakan **hashkiller.co.uk.**



Kami lanjutkan keempat file pdf uncompressed selanjutnya untuk mendapatkan pesan yang lain. Hasil kami pada tabel berikut.

No	Hash	Jenis	Tools	Decrypted
1	33e75ff09dd601bbe69f351039152189	MD5	http://hashkiller.co.uk/	28

2	ea0f3b9271eff45033bea3a6b1c36fa0000c3de7	SHA1	http://hashkiller.co.uk/	okt
3	a9bcf1e4d7b95a22e2975c812d938889	MD5	http://hashkiller.co.uk/	hari
4	1489eec1dd6f153d2da7d5a4b40bc078896fc006c8534dc449ce35e28436fd6a	SHA-256	http://md5decrypt.net/en/Sha256/	sumpah
5	44c582df70cc6e653adf2cf0df3f29fd1b6e16ca	SHA1	http://hashkiller.co.uk/	...?

Jika digabungkan hasil dekripsinya:

28 okt hari sumpah ...?

Jawabannya adalah **pemuda**

Maka selanjutnya buat file **soal1.py** dan melakukan submit jawaban.

```
soal1 = raw_input('')

print('pemuda')
```

[SOLVED]

Soal B : Tebak Gambar

Simpan gambar berikut. Jawaban problem B terdapat pada gambar tersebut.

Kumpulkan jawaban dalam sebuah source code Python berekstensi (.py). Dua baris kode adalah template yang akan ditambahkan dengan jawaban problem ini.

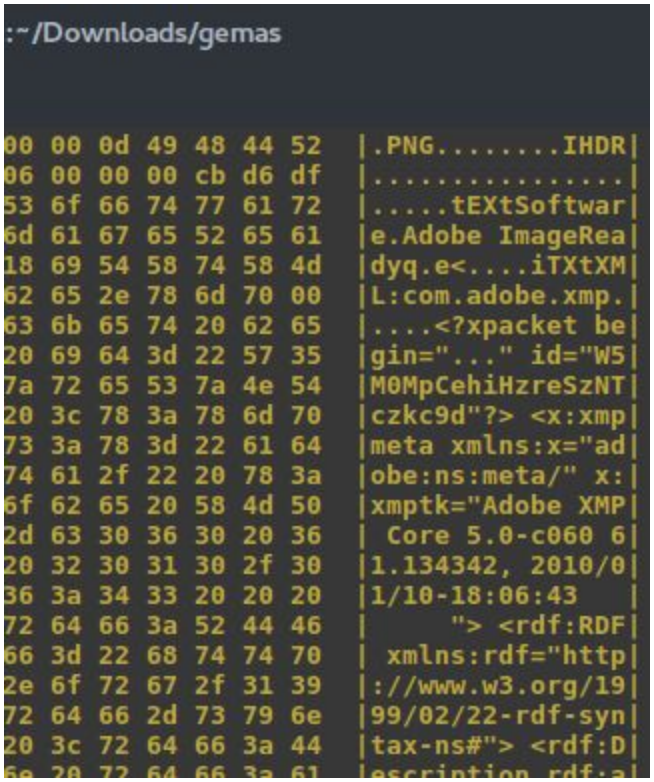
```
soal2 = raw_input('')

print('put your answer here')  #Tuliskan jawaban diantara tanda petik
```

File challenge:

Challenge

Analisa file

File unduhan tersebut berisi file gambar dengan ekstensi .png. Kemudian kami lakukan tools **hexdump** untuk mengetahui ekstensi sebenarnya dari gambar, ternyata dapat diketahui bahwa file tersebut memiliki ekstensi .zip, .txt, dan .png.

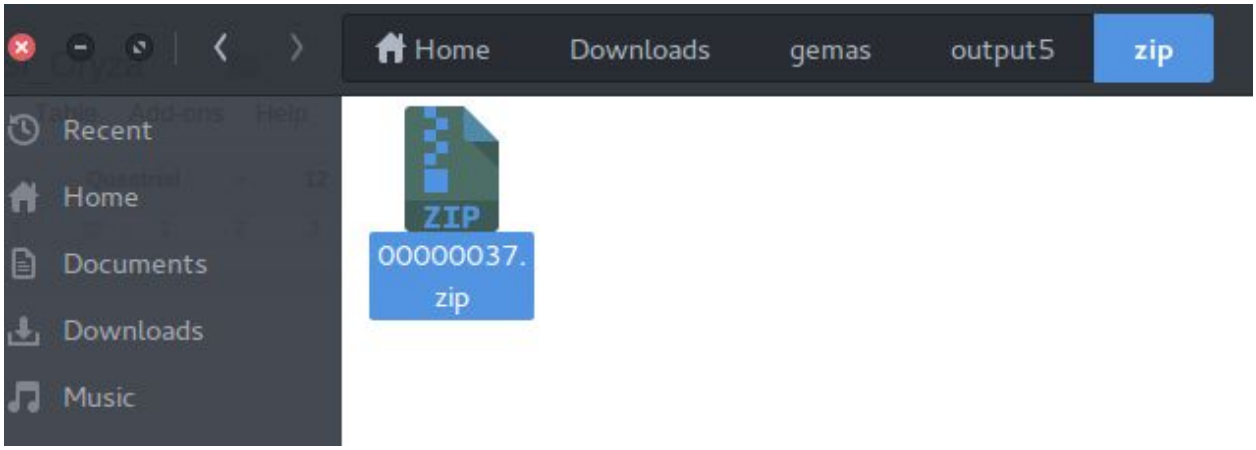


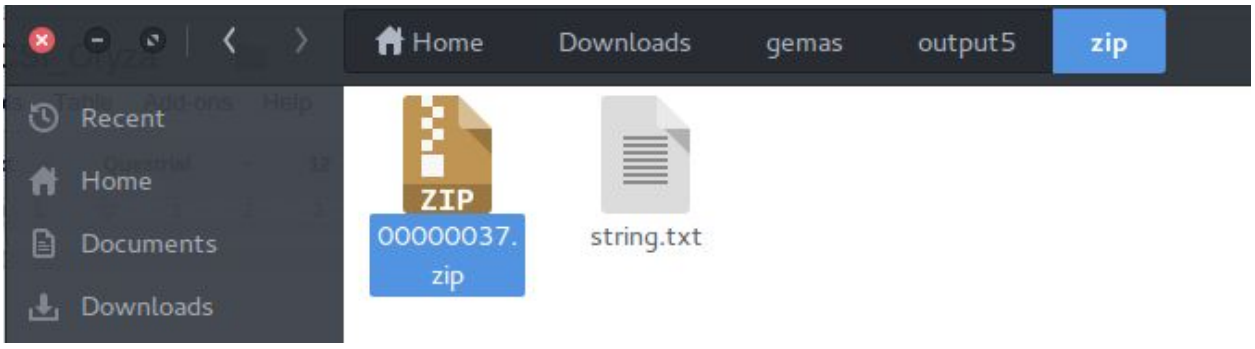
Setelah itu kami lakukan recovery file tersebut menggunakan tool **foremost.** Tool foremost merupakan tools forensik yang bisa melakukan analisa isi binary file tersebut. Ketika header suatu format file ditemukan dalam binary-nya, maka **foremost** langsung memotong bagian file tersebut dan mengekstraknya. Sehingga kami dapatkan file kompresi yang didalamnya berisi file string.txt.

```
[marlaw@gusti gemas] $ foremost -i hayo.png -o output5
```

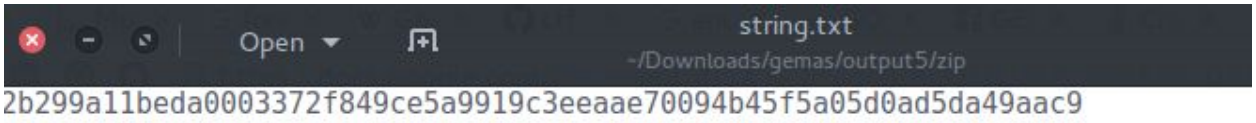
Kami lakukan foremost yang inputnya **hayo.png** dan outputnya di folder **output5.** Berikut ini potongan proses foremost.

Terlihat file string.txt yang ter-embed di hayo.png ketika dilakukan foremost. Di folder hasil foremost, terdapat file ZIP yang berisi file string.txt.



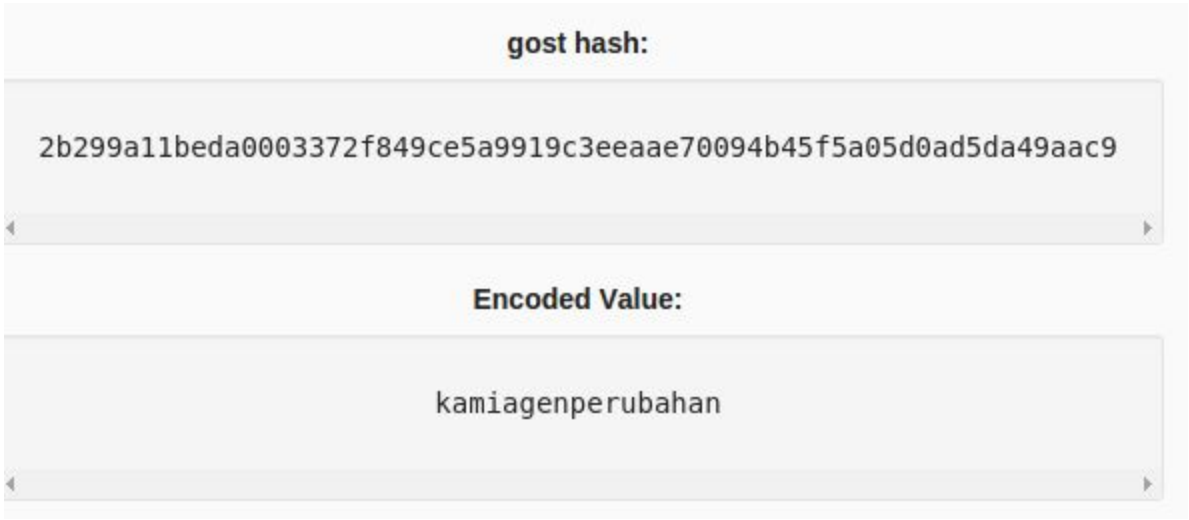


File string.txt berisi suatu hash:



Dari file string.txt tersebut kami dapatkan suatu hash yang panjangnya 64 karakter. Untuk mengetahui pesan sebenarnya kami lakukan percobaan untuk setiap metode hash, beberapa di antaranya adalah **sha256** dan **gost.** Kami mencoba melakukan decrypt string tersebut yang kami kira sha256 dengan hasil string "sumpah", tetapi ternyata wrong answer.
Lalu kami mencoba kemungkinan satunya lagi yaitu **gost**, lalu kami dapatkan hasil decrypt yaitu berupa pesan "kamiagenperubahan".

Tool yang kami gunakan untuk decrypt yaitu **md5hashing.net.**



Maka selanjutnya buat file **soal2.py** dan melakukan submit jawaban.

```
soal2 = raw_input('')

print('kamiagenperubahan')
```

[SOLVED]

Soal C : Dengarkan dan Sampaikan

Unduh file berikut dan kemudian selesaikan tantangan yang terdapat dalam file tersebut.

Challenge

Apakah Anda mendengar pesan yang disampaikan?
Kumpulkan jawaban dalam sebuah source code Python berekstensi (.py). Dua baris kode adalah template yang akan ditambahkan dengan jawaban problem ini.

```
soal3 = raw_input('')

print('put your answer here')  #Tuliskan jawaban diantara tanda petik
```

File challenge:

File challenge yang di-download bernama **soal3.tar**
File tersebut berupa archive sehingga perlu dilakukan dekompresi terlebih dahulu.

```
root@knight:~/Downloads/Gemastik8# tar -xf soal3.tar -v
soal3.pcap
```

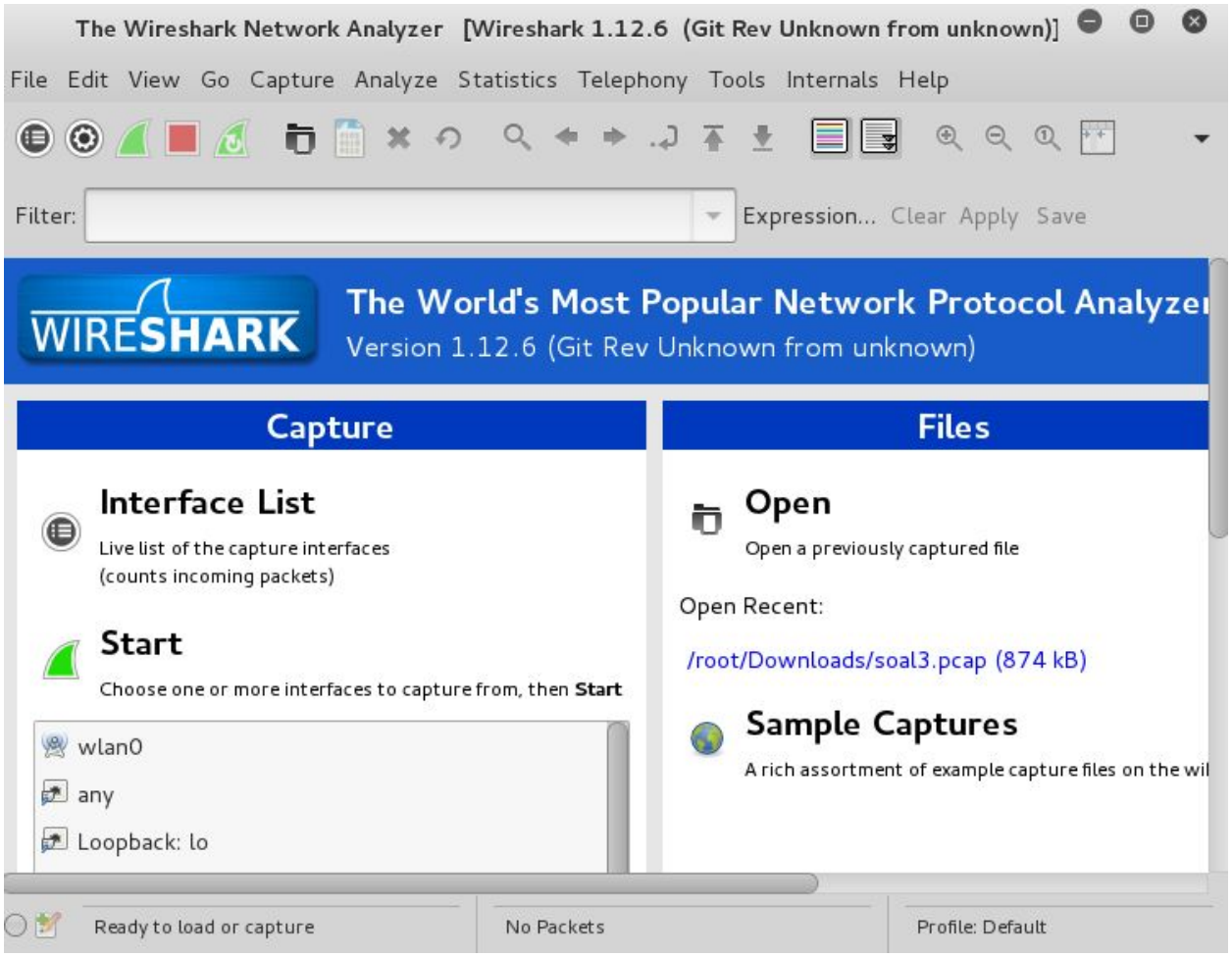
Di dalam file tersebut terdapat file "soal3.pcap". Berdasarkan ekstensi yang dimiliki, soal3.pcap merupakan file **packet capture**. File pcap berisikan paket-paket yang ditangkap melalui suatu jaringan tertentu.

Solution:

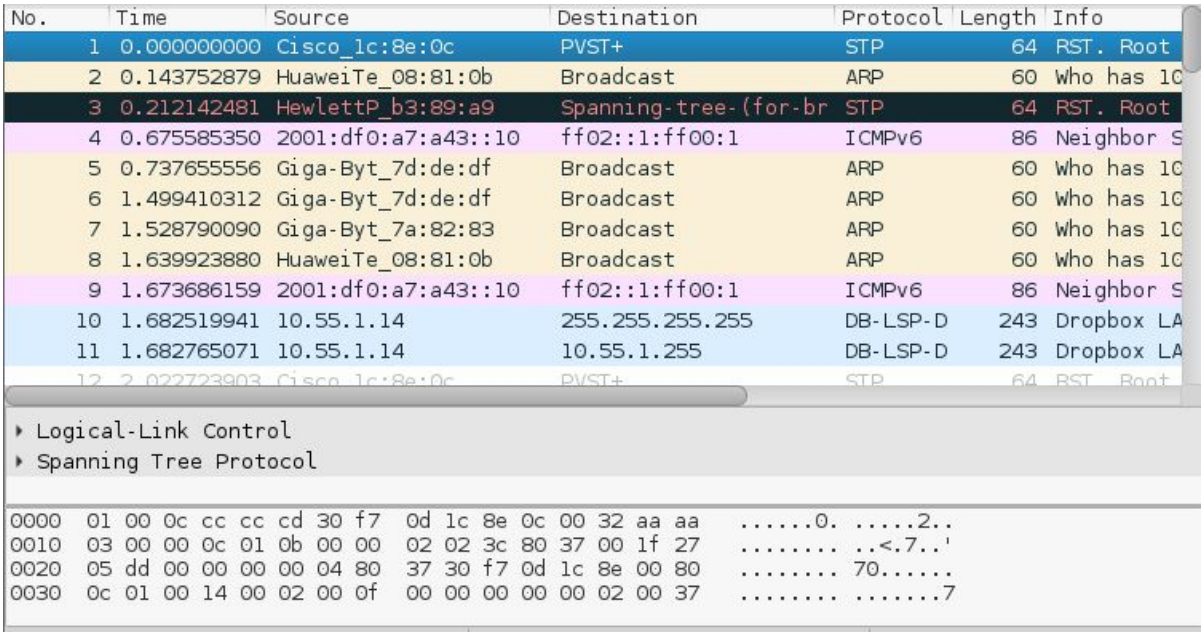
Tool yang dapat digunakan untuk menginvestigasi file berekstensi .pcap adalah **wireshark**.

- buka wireshark:

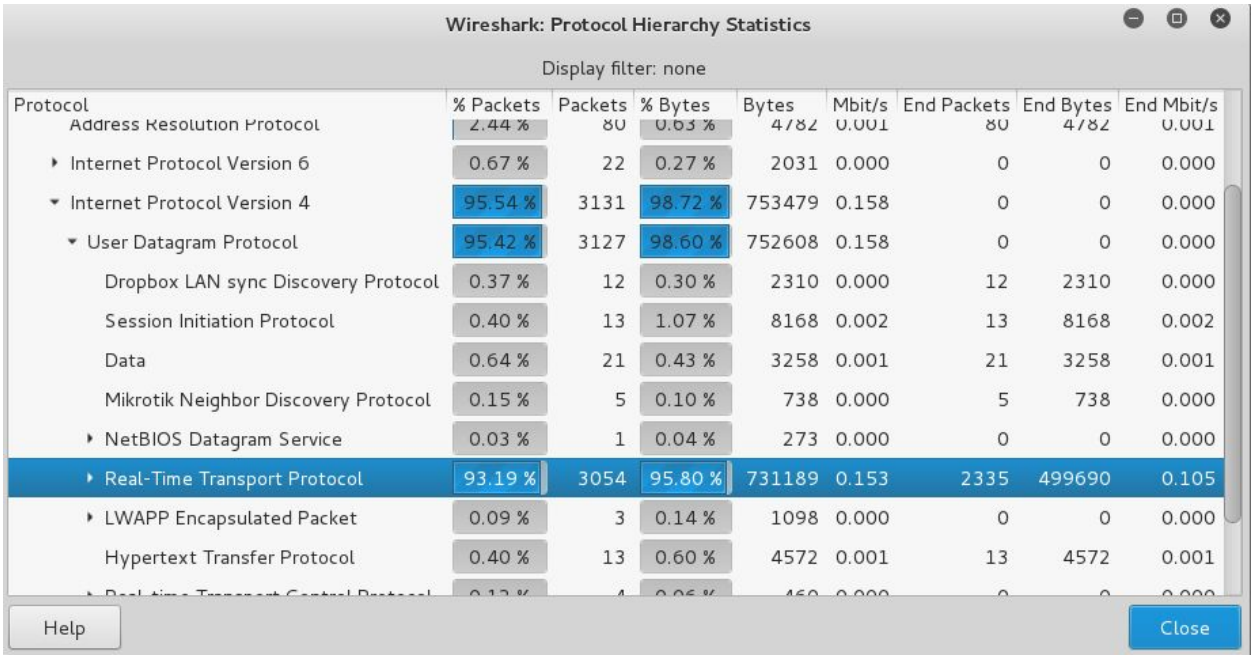
```
root@knight:~/Downloads/Gemastik8# wireshark
```



- Open file soal3.pcap



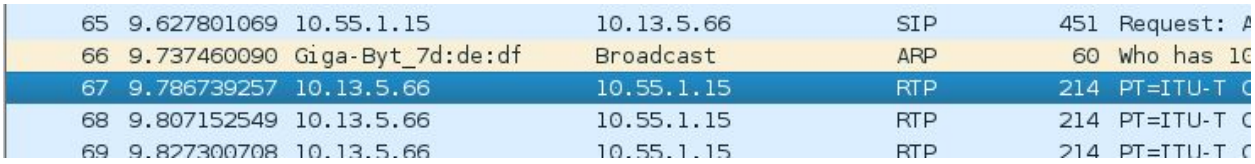
- Belum ditemukan hal yang menarik.
  Di dalam file tersebut terdapat banyak protokol. Untuk melihat statistik protokol yang ada buka **Statistics → Protocol Hierarchy**.



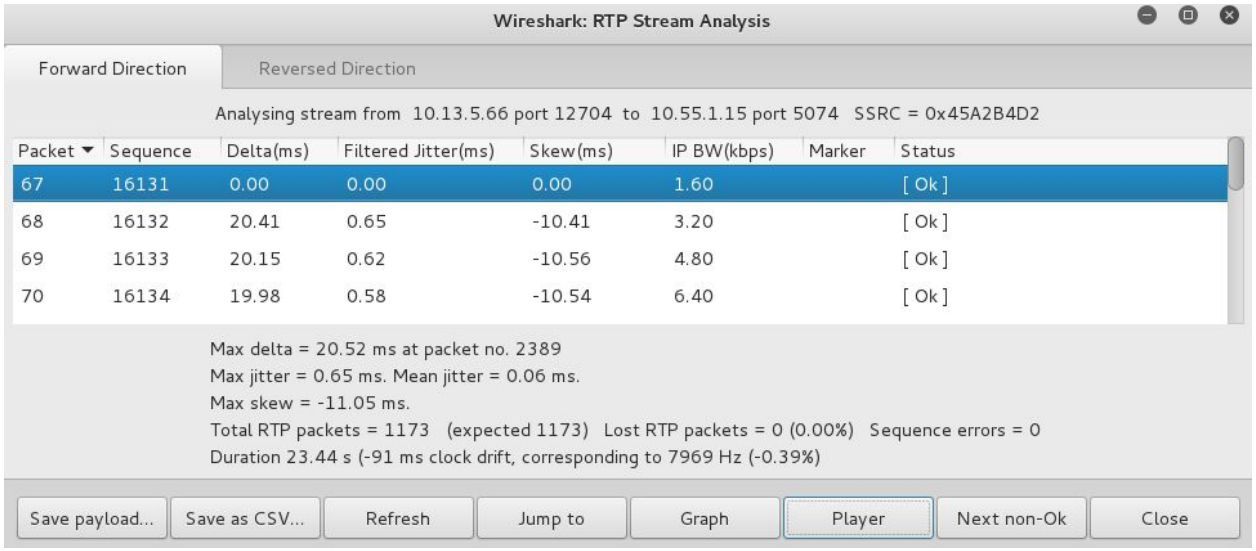
  - Dari hasil analisis, Terdapat protokol yang statistiknya menunjukkan penggunaan paling besar yaitu **RTP (Real-Time Transport Protocol)**. Berdasarkan definisinya, RTP dirancang untuk menyediakan fungsi transport jaringan ujung ke ujung untuk aplikasi yang mengirimkan data real time, misalnya audio atau video, melalui layanan jaringan multicast atau unicast.

    Dan dengan informasi soal yang berjudul "Dengarkan dan Sampaikan", maka dapat diambil kesimpulan ada sesuatu yang ditransfer melalui protokol RTP, dapat berupa file audio maupun video yang dapat mengeluarkan suara.

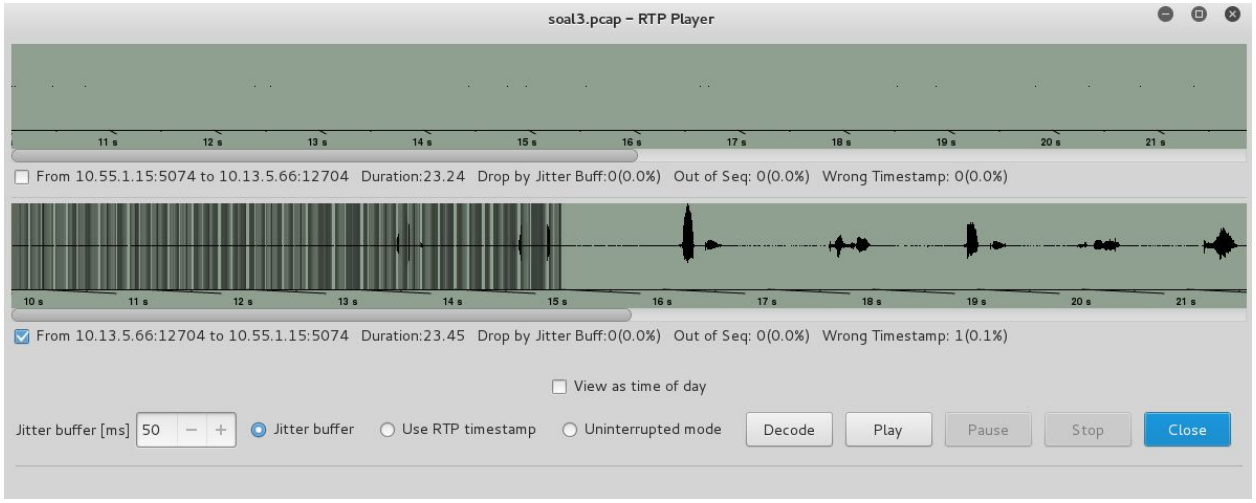
    Oleh karena itu, kami akan mencari ujung paket pada protokol RTP dan men-decodenya.



- Setelah ujung paket ditemukan, dilakukan decoding dengan memilih menu **Telephony → RTP → Stream analysis... → Player → Decode**



- Buka Player untuk melakukan replay terhadap isi stream paket.



- Setelah file audio di-decode, kemudian di-**Play**. akan muncul suara yang memberitahu deretan angka "**2810192828102015**" (kunci yang benar)

- Selanjutnya buat file **soal3.py** dan melakukan submit jawaban.

```
soal3 = raw_input('')

print('2810192828102015')
```

[UNSOLVED]

Soal D : WEB Login

Silahkan akses di alamat berikut. Carilah kata kunci administrator untuk di-submit

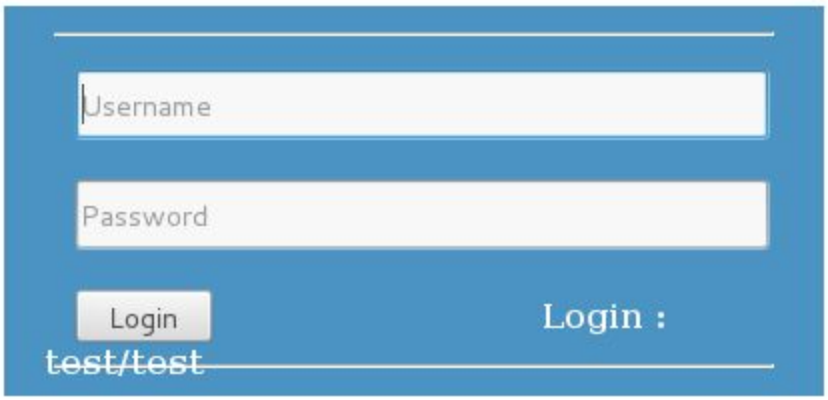
Challenge

Kumpulkan jawaban dalam sebuah source code Python berekstensi (.py). Dua baris kode adalah template yang akan ditambahkan dengan jawaban problem ini.

```
soal4 = raw_input('')

print('put your answer here')  #Tuliskan jawaban diantara tanda petik
```

File challenge:

Challange mengarah ke halaman login pada alamat http://175.111.88.222/login.php



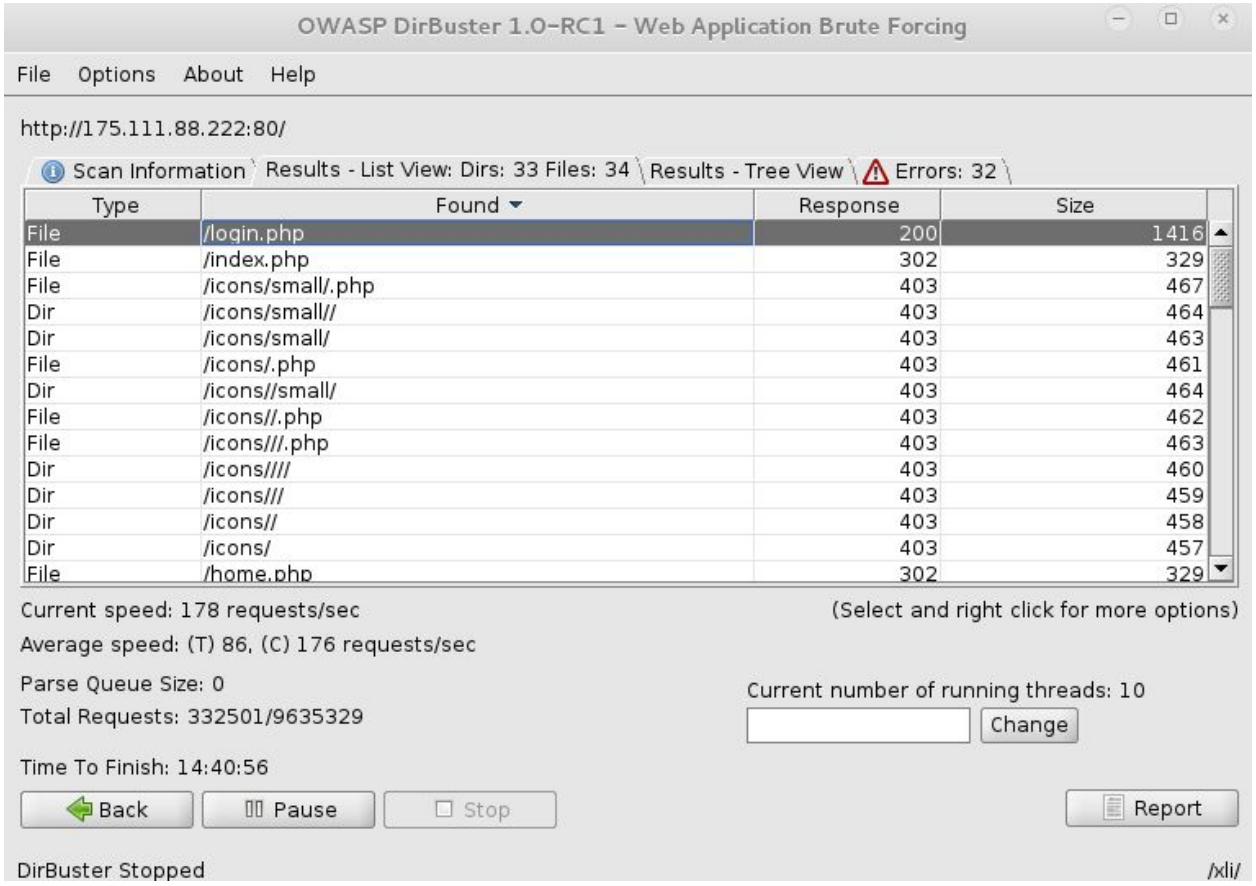
Solution:

- Analisa pertama form tersebut vulnerable terhadap sql injection. Sehingga kami coba mem-bypass form tersebut dengan payload **') OR '1'='1--** pada field Username.

Namun menghasilkan hasil negatif artinya field Username tidak dapat diinjeksi menggunakan payload tersebut.

- Kami kemudian mencoba melakukan hal yang sama terhadap field Password dengan payload yang sama yaitu **') OR '1'='1--**, namun kami tetap memperoleh hasil negatif.

- Setelah mencoba tehnik sql injection pada form login dengan method post yang memberikan hasil negatif, kami mencoba untuk melakukan brute-force pada direktori WEB dengan menggunakan tool **dir-buster**.



Hasil negatif tetap kami peroleh karena setelah melakukan brute-force dan mengecek file satu-persatu, tidak ditemukan flag maupun informasi penting di dalamnya.

- Kami kemudian mencoba melakukan brute-force terhadap password serta username pada halaman login, tetapi mekanisme name field username serta password yang di-random menyebabkan tehnik tersebut tidak dapat dilakukan

- Selanjutnya kami men-scan server menggunakan **nmap**, hal tersebut kami lakukan berharap menemukan service-service yang dapat membantu dalam menemukan flag. Port pertama yang kami scan adalah port 3306 letak service mysql bekerja.

```
hrdn@hacking:~/workspace $ nmap -p 3306 175.111.88.222

Starting Nmap 6.40 ( http://nmap.org ) at 2015-09-16 11:42 UTC
Nmap scan report for host-175-111-88-222.ugm.ac.id (175.111.88.222)
Host is up (0.082s latency).
PORT     STATE  SERVICE
3306/tcp closed mysql

Nmap done: 1 IP address (1 host up) scanned in 0.33 seconds
```

  ternyata port default mysql **3306 ditutup**, artinya kemungkinan web ini tidak menjalankan mysql, lalu kami mencoba mendeteksi nosql salah satunya **mongodb** menggunakan nmap

```
hrdn@hacking:~/workspace $ nmap -p 27017 --script mongodb-info 175.111.88.222

Starting Nmap 6.40 ( http://nmap.org ) at 2015-09-16 11:43 UTC
Nmap scan report for host-175-111-88-222.ugm.ac.id (175.111.88.222)
Host is up (0.083s latency).
PORT      STATE  SERVICE
27017/tcp closed unknown

Nmap done: 1 IP address (1 host up) scanned in 0.22 seconds
```

  hasilnya negatif, port default monggodb juga ditutup.

- Kami tidak dapat menemukan flag dari soal 4 hingga waktu pengerjaan problem selesai.

[SOLVED]

Soal E : Fun Cipher

Geser sedikit dong ...

```
Shpxgd lqgrqhvld bdqj ehuedkdjld. Nlwd dgdodk jhqhudvl shqhuxv edqjvd. Edjldq shuwdpd gdul nxqfl dgdodk "uxgqx", wdqsd wdqgd shwln.
```

Butuh kunci untuk membuka pesan.

```
lyek iuo 1000 btixa gwi, xcfeiiu nmix eheilog umwyew lklv csklaai. lyek iuo 10 cguexn pqcwnai kenp seahpkkhtmix xhpqk. vniqkh xgleu qczs ehpks uqctkb "zivknacx", duari duafi zygks
```

Bekerja keras untuk memenuhi kebutuhan keluarga (diperibahasakan)

```
Yztrzm gvizpsri wzir pfmxr zwzozs "hfivgfqzn" gzmkz gzmwz kvgrp.
```

Kumpulkan jawaban dalam sebuah source code Python berekstensi (.py). Dua baris kode adalah template yang akan ditambahkan dengan jawaban problem ini.

```
soal5 = raw_input('')

print('put your answer here') #Tuliskan jawaban diantara tanda petik
```

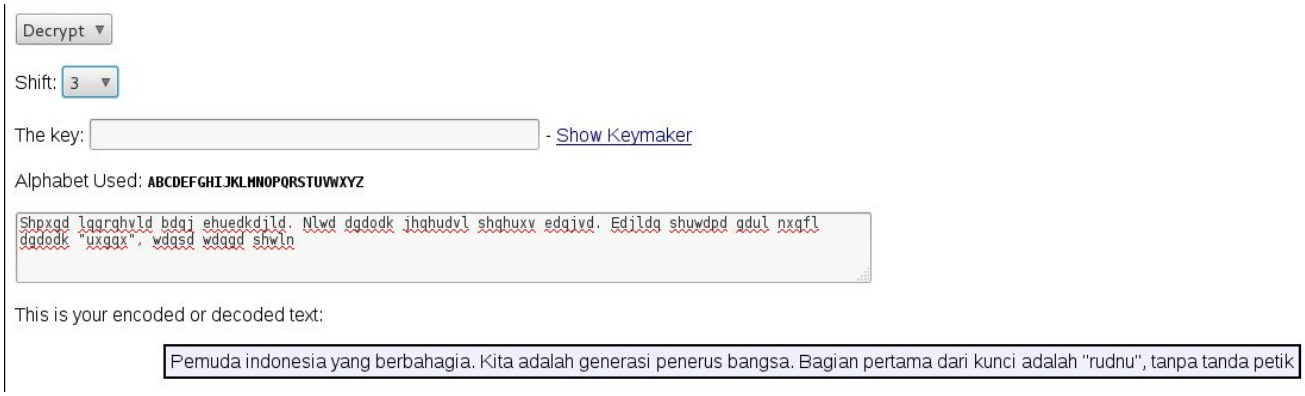
Analisis

Soal ini terdiri atas beberapa metode cipher yang menyembunyikan pesan.

Solusi:

- Untuk cipher pertama "**Shpxgd lqgrqhvld bdqj ehuedkdjld. Nlwd dgdodk jhqhudvl shqhuxv edqjvd. Edjldq shuwdpd gdul nxqfl dgdodk "uxgqx", wdqsd wdqgd shwln**" merupakan plaintext yang dienkripsi dengan algoritma caesar cipher karena di soal diberi clue "geser sedikit dong".

  Menggunakan tool dekriptor online http://rumkin.com/tools/cipher/caesar-keyed.php maka ditemukan plaintext berupa



Potongan kunci pertama adalah "**rudnu**".

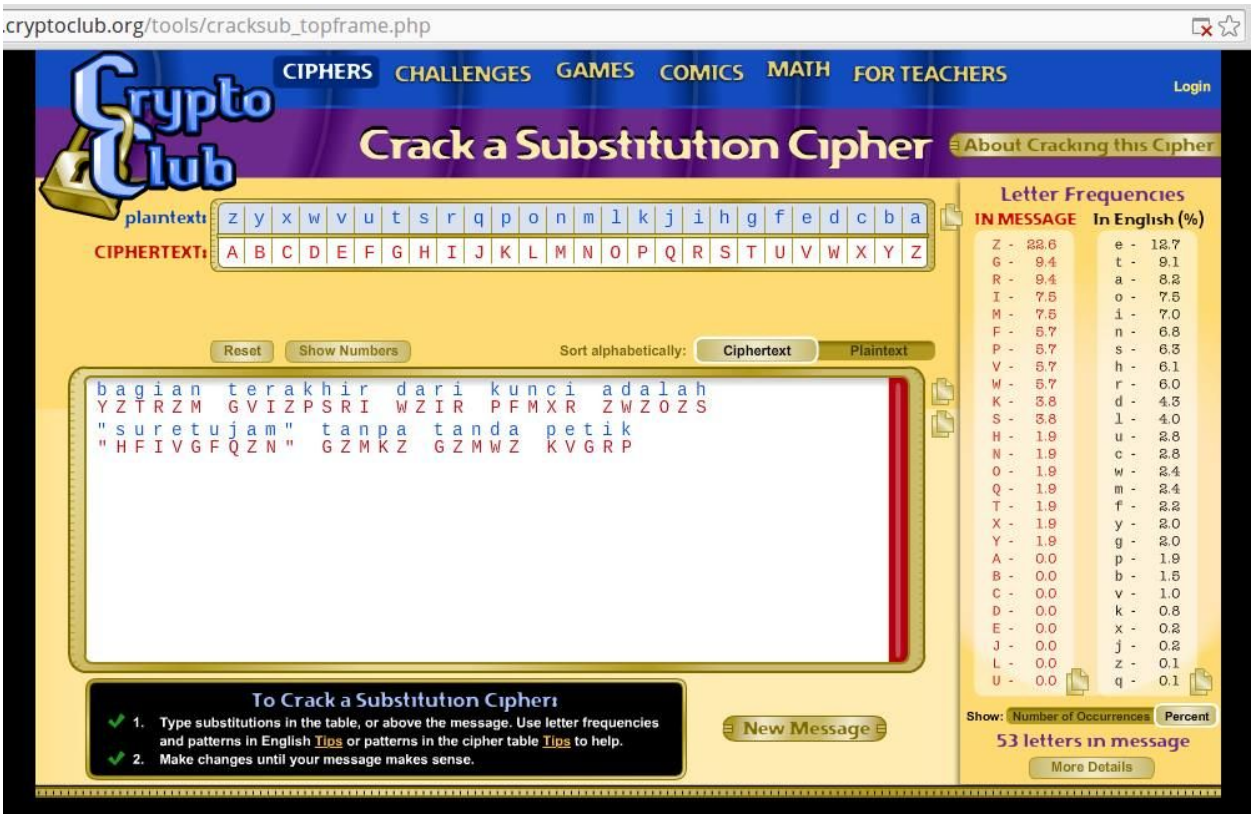
- Kami kemudian langsung melakukan dekripsi pada cipher text ketiga untuk memperoleh potongan kunci ketiga. Dengan memperoleh potongan terakhir, kemungkinan kami tidak perlu melakukan dekripsi terhadap cipher text kedua karena kemungkinan besar potongan flag tersebut memiliki makna yang berhubungan.

  Cipher text ketiga "**Yztrzm gvizpsri wzir pfmxr zwzozs "hfivgfqzn" gzmkz gzmwz kvgrp**" merupakan plaintext yang dienkripsi menggunakan algoritma subtitution cipher.

  Kami menganggap substitution cipher karena dari bagian pertama, ada kata "**adalah**", dan di ciphertext ini ada kata "**zwzozs**", yang dapat kita buat mapping:

  - `z = a`
  - `w = d`
  - `o = l`
  - `s = h`, dan sebaliknya

  Dengan menggunakan tool online dekriptor www.cryptoclub.org/tools/cracksub_topframe.php dan mencoba substitusi huruf yang memberikan pesan yang bermakna, diperoleh:



Bagian ketiga adalah "**suretujam**".

- Jika dihubungkan kedua bagian tersebut menghasilkan "**rudnu...suretujam**". Sesuai perkiraan kami, kunci tersebut bermakna setelah di-reverse menjadi "**majuterus...undur**". Maka kunci yang lengkap adalah "majuterus**pantangm**undur" dengan "**mgnatnap**" sebagai potongan kunci kedua. Dengan demikian kami tidak perlu melakukan kriptanalisis bagian kedua.

- Selanjutnya buat file **soal5.py** dan melakukan submit jawaban.

```
soal5 = raw_input('')

print('majuteruspantangmundur'[::-1]) # membalik string
```