



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:

```
%PDF-1.4
3 0 obj
/Type /EmbeddedFile
/Filter /FlateDecode
/Params 1 0 R
/Length 2 0 R
stream
-sa`#
endstream
endobj
2 0 obj 80
endobj
1 0 obj
/Size 73
endobj
4 0 obj
/Type /F
/F (data1.txt)
/EF
/F 3 0 R
/UF (
endobj
5 0 obj
/Names [(
t) 4 0 R]
endobj
6 0 obj
```

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.

```
<< /Params 15 0 R /Type /EmbeddedFile /Length 73 >>
stream
Bagian pertama.
Dari mana datangnya 33e75ff09dd601bbe69f351039152189 ?
endstream
endobj
13 0 obj
<< /OPM 1 /Type /ExtGState >>
endobj
```

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

```
We found 1 hashes! [Timer: 225 m/s] Please find them below...
33e75ff09dd601bbe69f351039152189 33e75ff09dd601bbe69f351039152189 MD5 : 28
```

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	ea0f3b9271eff450 33bea3a6b1c36fa0 000c3de7	SHA1	http://hashkiller.co.uk/	okt
3	a9bcf1e4d7b95a22 e2975c812d938889	MD5	http://hashkiller.co.uk/	hari
4	1489eec1dd6f153d 2da7d5a4b40bc078 896fc006c8534dc4 49ce35e28436fd6a	SHA-256	http://md5decrypt.net/en/Sha256/	sumpah
5	44c582df70cc6e65 3adf2cf0df3f29fd 1b6e16ca	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.

```

:~/Downloads/gemas
00 00 0d 49 48 44 52 |.PNG.....IHDR|
06 00 00 00 cb d6 df |.....|
53 6f 66 74 77 61 72 |.....tEXtSoftwar|
6d 61 67 65 52 65 61 |e.Adobe ImageRea|
18 69 54 58 74 58 4d |dyq.e<.....iTXtXM|
62 65 2e 78 6d 70 00 |l:com.adobe.xmp.|
63 6b 65 74 20 62 65 |....<?xpacket be|
20 69 64 3d 22 57 35 |gin="..." id="W5|
7a 72 65 53 7a 4e 54 |M0MpCehiHzreSzNT|
20 3c 78 3a 78 6d 70 |czkc9d"?> <x:xmp|
73 3a 78 3d 22 61 64 |meta xmlns:x="ad|
74 61 2f 22 20 78 3a |obe:ns:meta/" x:|
6f 62 65 20 58 4d 50 |xmptk="Adobe XMP|
2d 63 30 36 30 20 36 |Core 5.0-c060 6|
20 32 30 31 30 2f 30 |1.134342, 2010/0|
36 3a 34 33 20 20 20 |1/10-18:06:43 |
72 64 66 3a 52 44 46 |"> <rdf:RDF|
66 3d 22 68 74 74 70 | xmlns:rdf="http|
2e 6f 72 67 2f 31 39 |://www.w3.org/19|
72 64 66 2d 73 79 6e |99/02/22-rdf-syn|
20 3c 72 64 66 3a 44 |tax-ns#"> <rdf:D|
5e 20 72 64 66 3a 61 |escription rdf:a|

```

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.

```

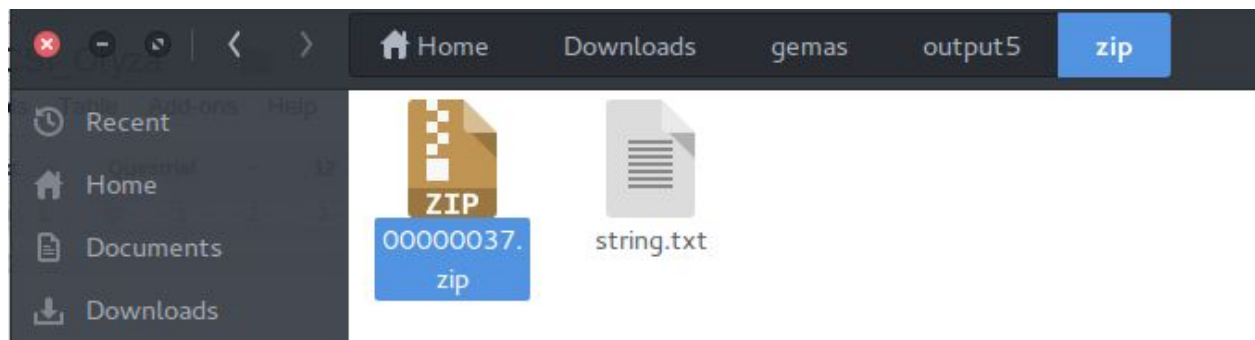
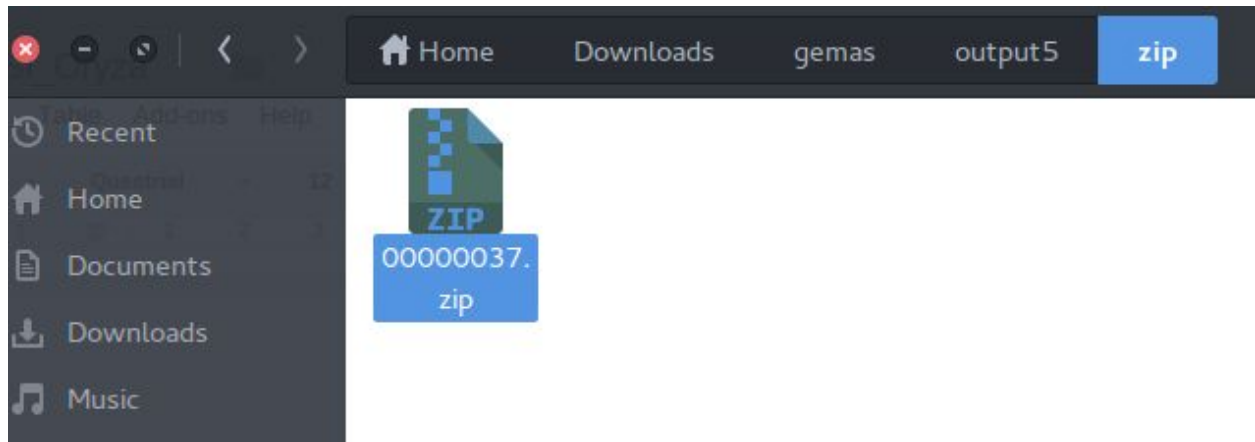
[marlaw@gusti gemas] $ foremost -i hayo.png -o output5

```

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


```
marlaw@gusti:~/Downloads/gemas
File Edit View Search Terminal Help
marlaw@gusti ~] $ foremost hayo.png output2
Processing: stdin
^C
marlaw@gusti ~] $ cd Do
cuments/ Downloads/
marlaw@gusti ~] $ cd Downloads/
marlaw@gusti Downloads] $ cd gemas
marlaw@gusti gemas] $ foremost hayo.png output5
ERROR: /home/marlaw/Downloads/gemas/output is not empty
Please specify another directory or run with -T.
marlaw@gusti gemas] $ foremost -i hayo.png -o output5
Processing: hayo.png
foundat=string.txt00000000L0K=0y00N0f0#
*
marlaw@gusti gemas] $
```

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**.

A screenshot of the md5hashing.net website interface. It features two main input/output sections. The first section is labeled 'gost hash:' and contains a text box with the hash '2b299a11beda0003372f849ce5a9919c3eeaae70094b45f5a05d0ad5da49aac9'. The second section is labeled 'Encoded Value:' and contains a text box with the decoded message 'kamiagenperubahan'.

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
```

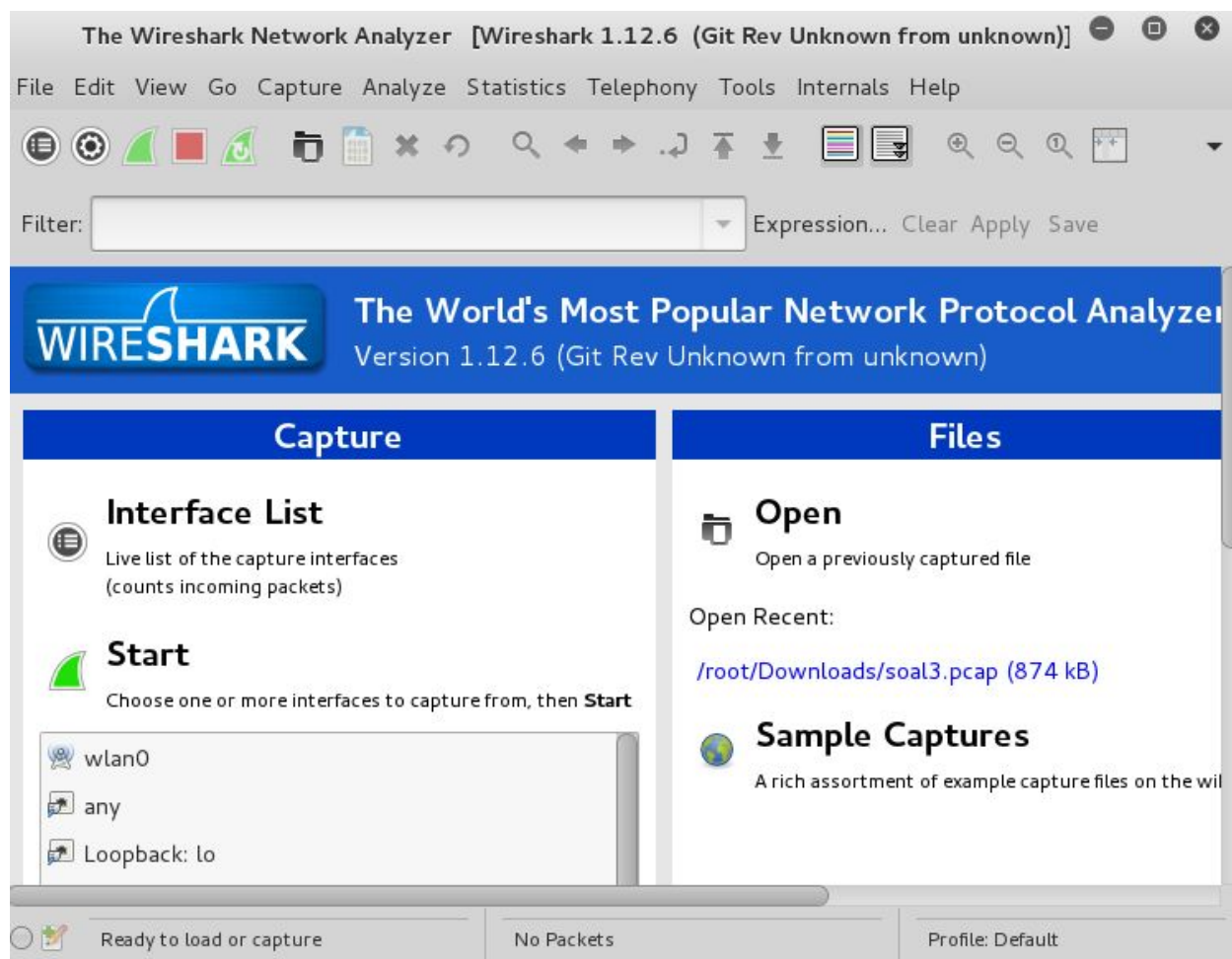
Di dalam file tersebut terdapat file "soal3.pcap". Berdasarkan ekstensi yang dimiliki, soal3.pcap merupakan file **packet capture**. File pcap berisikan paket-paket yang ditangkap melalui suatu jaringan tertentu.

Solution:

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

- buka wireshark:

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



- Open file soal3.pcap

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	Cisco_1c:8e:0c	PVST+	STP	64	RST. Root
2	0.143752879	HuaweiTe_08:81:0b	Broadcast	ARP	60	Who has 10
3	0.212142481	HewlettP_b3:89:a9	Spanning-tree-(for-br	STP	64	RST. Root
4	0.675585350	2001:df0:a7:a43::10	ff02::1:ff00:1	ICMPv6	86	Neighbor S
5	0.737655556	Giga-Byt_7d:de:df	Broadcast	ARP	60	Who has 10
6	1.499410312	Giga-Byt_7d:de:df	Broadcast	ARP	60	Who has 10
7	1.528790090	Giga-Byt_7a:82:83	Broadcast	ARP	60	Who has 10
8	1.639923880	HuaweiTe_08:81:0b	Broadcast	ARP	60	Who has 10
9	1.673686159	2001:df0:a7:a43::10	ff02::1:ff00:1	ICMPv6	86	Neighbor S
10	1.682519941	10.55.1.14	255.255.255.255	DB-LSP-D	243	Dropbox LA
11	1.682765071	10.55.1.14	10.55.1.255	DB-LSP-D	243	Dropbox LA
12	2.022723903	Cisco_1c:8e:0c	PVST+	STP	64	RST. Root

Logical-Link Control						
Spanning Tree Protocol						

0000	01 00 0c cc cc cd 30 f7	0d 1c 8e 0c 00 32 aa aa0.2..
0010	03 00 00 0c 01 0b 00 00	02 02 3c 80 37 00 1f 27<.7..'
0020	05 dd 00 00 00 00 04 80	37 30 f7 0d 1c 8e 00 80 70.....
0030	0c 01 00 14 00 02 00 0f	00 00 00 00 00 02 00 377

- Belum ditemukan hal yang menarik.

Di dalam file tersebut terdapat banyak protokol. Untuk melihat statistik protokol yang ada buka **Statistics → Protocol Hierarchy**.

Wireshark: Protocol Hierarchy Statistics

Display filter: none

Protocol	% Packets	Packets	% Bytes	Bytes	Mbit/s	End Packets	End Bytes	End Mbit/s
Address Resolution Protocol	2.44 %	80	0.63 %	4782	0.001	80	4782	0.001
Internet Protocol Version 6	0.67 %	22	0.27 %	2031	0.000	0	0	0.000
Internet Protocol Version 4	95.54 %	3131	98.72 %	753479	0.158	0	0	0.000
User Datagram Protocol	95.42 %	3127	98.60 %	752608	0.158	0	0	0.000
Dropbox LAN sync Discovery Protocol	0.37 %	12	0.30 %	2310	0.000	12	2310	0.000
Session Initiation Protocol	0.40 %	13	1.07 %	8168	0.002	13	8168	0.002
Data	0.64 %	21	0.43 %	3258	0.001	21	3258	0.001
Mikrotik Neighbor Discovery Protocol	0.15 %	5	0.10 %	738	0.000	5	738	0.000
NetBIOS Datagram Service	0.03 %	1	0.04 %	273	0.000	0	0	0.000
Real-Time Transport Protocol	93.19 %	3054	95.80 %	731189	0.153	2335	499690	0.105
LWAPP Encapsulated Packet	0.09 %	3	0.14 %	1098	0.000	0	0	0.000
Hypertext Transfer Protocol	0.40 %	13	0.60 %	4572	0.001	13	4572	0.001
Real-Time Transport Control Protocol	0.13 %	4	0.06 %	460	0.000	0	0	0.000

Help Close

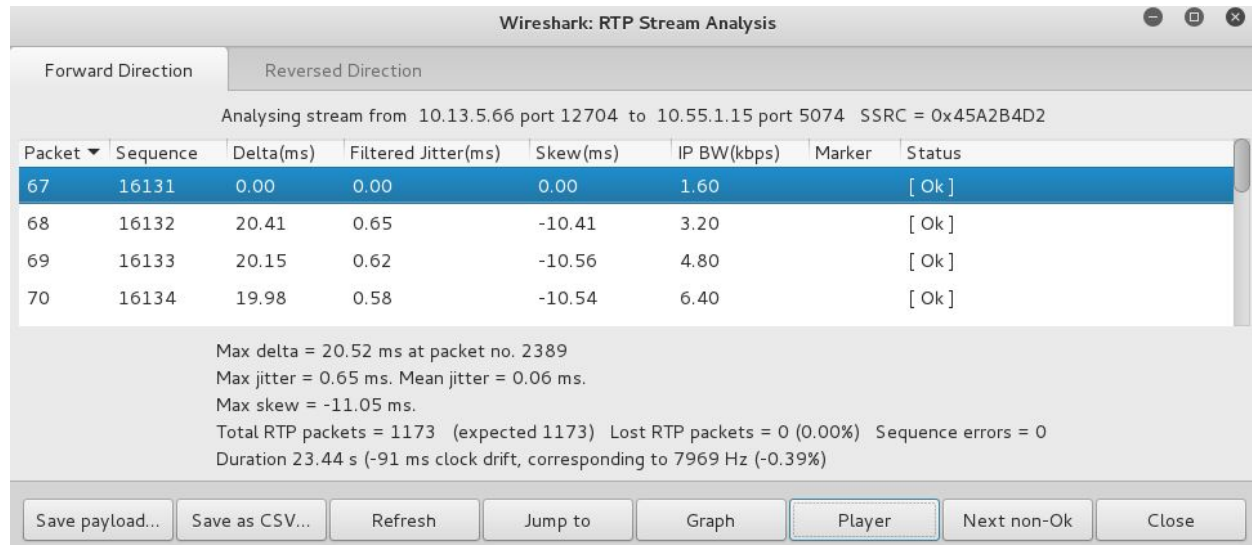
- Dari hasil analisis, Terdapat protokol yang statistiknya menunjukkan penggunaan paling besar yaitu **RTP (Real-Time Transport Protocol)**. Berdasarkan definisinya, RTP dirancang untuk menyediakan fungsi transport jaringan ujung ke ujung untuk aplikasi yang mengirimkan data real time, misalnya audio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

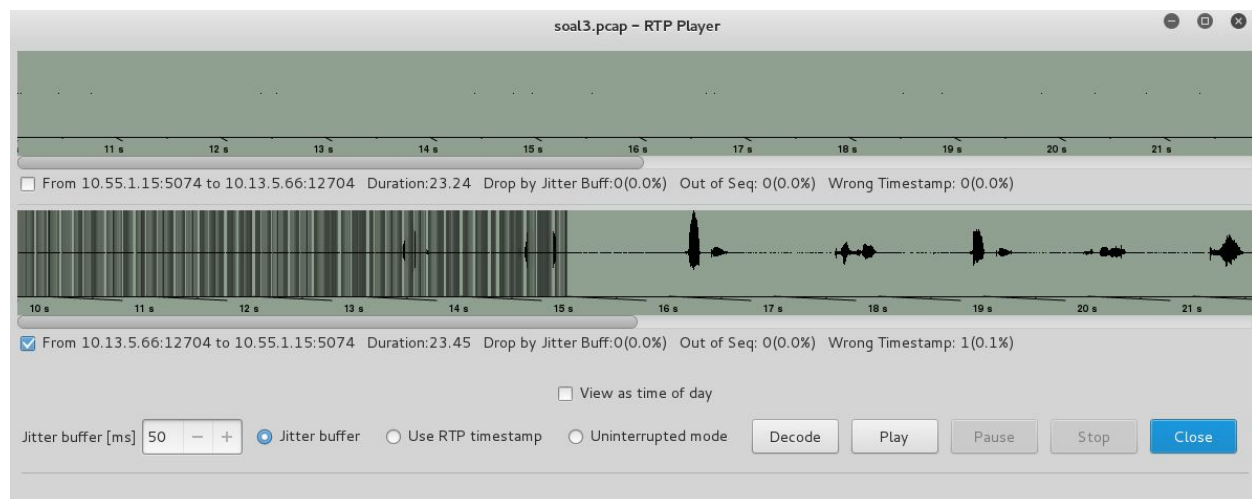
Oleh karena itu, kami akan mencari ujung paket pada protokol RTP dan men-decodenya.

65	9.627801069	10.55.1.15	10.13.5.66	SIP	451	Request: A
66	9.737460090	Giga-Byt_7d:de:df	Broadcast	ARP	60	who has 10
67	9.786739257	10.13.5.66	10.55.1.15	RTP	214	PT=ITU-T C
68	9.807152549	10.13.5.66	10.55.1.15	RTP	214	PT=ITU-T C
69	9.827300708	10.13.5.66	10.55.1.15	RTP	214	PT=ITU-T C

- 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:

Challenge mengarah ke halaman login pada alamat <http://175.111.88.222/login.php>



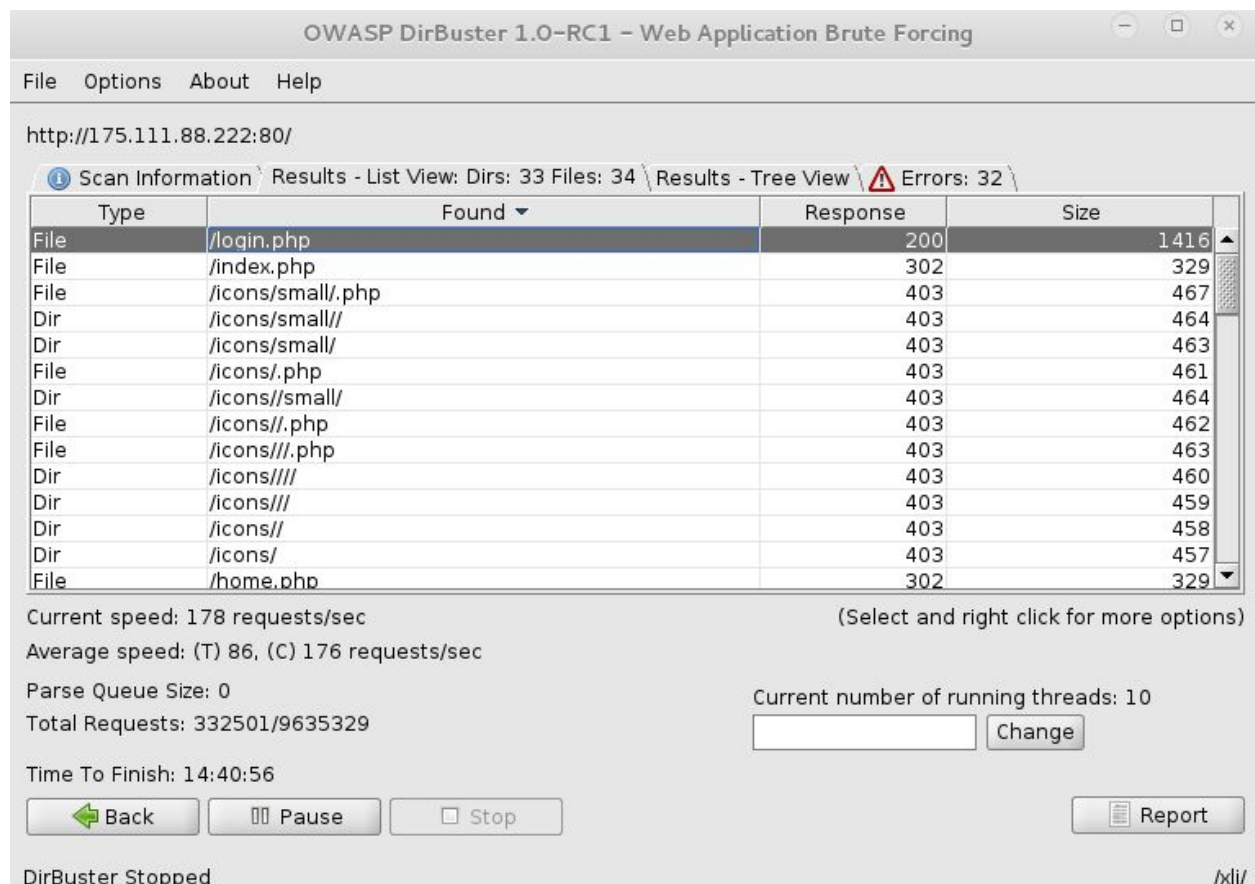
The image shows a web login interface with a blue background. It features two input fields: 'Username' and 'Password'. Below the 'Password' field is a 'Login' button. To the right of the button, the text 'Login :' is displayed. At the bottom left, the text 'test/test' is visible, likely representing the default credentials.

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 mongodb 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 eheillog umwyew lklv csklaai. lyek iuo 10 cguexn pqcwnai kenp seahpkkhtmix xhpqk. vniqkh xgleu qczs ehpkv uqctkb "zivknacx", duari duaafi zygks

Bekerja keras untuk memenuhi kebutuhan keluarga (diperibahasakan)

Yztrzm gvizpsri wzir pfmxr zwzozs "hfivgfqzn" gzmkg 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

Decrypt ▼

Shift: 3 ▼

The key: - [Show Keymaker](#)

Alphabet Used: ABCDEFGHIJKLMNOPQRSTUVWXYZ

Shpxgd laarghylvd bdaj ehuedkdild. Nlwd dadodk ihghudvl shghuxv edajvd. Edildg shuwddp adul nxafldadodk "uxgax", wdgqd wdgqd shwln

This is your encoded or decoded text:

Pemuda Indonesia yang berbahagia. Kita adalah generasi penerus bangsa. Bagian pertama dari kunci adalah "rudnu", tanpa tanda petik

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 dienkrpsi menggunakan algoritma substitution 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:

cryptoclub.org/tools/cracksub_topframe.php

Crypto Club CIPHERS CHALLENGES GAMES COMICS MATH FOR TEACHERS Login

Crack a Substitution Cipher

About Cracking this Cipher

plaintext: z y x w v u t s r q p o n m l k j i h g f e d c b a
 CIPHERTEXT: A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

Reset Show Numbers Sort alphabetically: Ciphertext Plaintext

bagian terakhir dari kunci adalah
 YZTRZM GVIZPSRI WZIR PFMXR ZWZOZS
 "suretujem" tanpa tanda petik
 "HFIVGFQZN" GZMKZ GZMWZ KVGRP

To Crack a Substitution Cipher:

1. Type substitutions in the table, or above the message. Use letter frequencies and patterns in English [Tips](#) or patterns in the cipher table [Tips](#) to help.
2. Make changes until your message makes sense.

New Message

Letter Frequencies

IN MESSAGE	In English (%)
Z - 22.6	e - 12.7
G - 9.4	t - 9.1
R - 9.4	a - 8.2
I - 7.5	o - 7.5
M - 7.5	i - 7.0
F - 5.7	n - 6.8
P - 5.7	s - 6.3
V - 5.7	h - 6.1
W - 5.7	r - 6.0
K - 3.8	d - 4.3
S - 3.8	l - 4.0
H - 1.9	u - 2.8
N - 1.9	c - 2.8
O - 1.9	w - 2.4
Q - 1.9	m - 2.4
T - 1.9	f - 2.2
X - 1.9	y - 2.0
Y - 1.9	g - 2.0
A - 0.0	p - 1.9
B - 0.0	b - 1.5
C - 0.0	v - 1.0
D - 0.0	k - 0.8
E - 0.0	x - 0.2
J - 0.0	j - 0.2
L - 0.0	z - 0.1
U - 0.0	q - 0.1

Show: Number of Occurrences Percent

53 letters in message

More Details

Bagian ketiga adalah "suretujem".

- Jika dihubungkan kedua bagian tersebut menghasilkan "rudnu...suretujem". Sesuai perkiraan kami, kunci tersebut bermakna setelah di-reverse menjadi "majuterus...undur". Maka kunci yang lengkap adalah "majuteruspantangmundur" 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
```