

Project 1

賴文揚 0316025

robots.txt

- 從這個檔案中擷取到，這個Domain不想讓search engine preview的文件。
 - /backup.tar.gz
 - 可download，內容為網站php原始碼
 - Found password hash function is mysql323
 - /phpMyAdmin_NS_pRojEct_2017
 - Entering /phpMyAdmin_NS_pRojEct_2017/index.php
 - /blog/memorandum.txt
 - 404 NOT FOUND
 - But find out “.memorandum.txt.swp” vim temp file

PHP source code

- Get a lot of source code
 - functions.php (Most important)
 - Get hash function information (using mysql323)
 - index.php
 - show.php
 - views(folder)
 - Index.php
 - Show.php
 - element (folder)
 - Css and js files
 - ...etc

Base64 Online Decoder

- We can simply use frequency analysis to know “.memorandum.txt.swp” is using Base64 encode
- Using Online tool to recover it!
 - <https://www.base64decode.org/>

Decryption the .memorandum.txt.swp

- Guess it is using xor encryption
 - Using my own tool by command
 - \$./frequency_analyze num
 - “num” indicate the length of password using in xor encryption
 - And We can get the results when typing num equal to 7

```
[XD] % ./frequency_analyze 7
cipher length: 7
0th round
char: 80, frequency: 0.064641
char: a1, frequency: 0.064641
char: e2, frequency: 0.193924
char: 00, frequency: 1.098901
char: 01, frequency: 2.908856
char: 03, frequency: 4.783452
char: 04, frequency: 1.486749
char: 05, frequency: 1.422107
char: 06, frequency: 3.555269
char: 07, frequency: 9.502263
char: 09, frequency: 0.840336
char: 0a, frequency: 4.524887
char: 0b, frequency: 5.946994
char: 0c, frequency: 5.106658
char: 0d, frequency: 5.494505
char: 0e, frequency: 2.197802
char: 0f, frequency: 2.585650
char: 10, frequency: 4.654170
char: 11, frequency: 4.524887
char: 12, frequency: 1.616031
char: 14, frequency: 0.581771
char: 15, frequency: 1.098901
char: 16, frequency: 7.304461
```

```
1th round
char: 97, frequency: 0.064641
char: d0, frequency: 0.064641
char: e8, frequency: 0.064641
char: e9, frequency: 0.064641
char: ec, frequency: 0.064641
char: f5, frequency: 0.064641
char: 00, frequency: 1.292825
char: 01, frequency: 6.140918
char: 02, frequency: 0.646412
char: 03, frequency: 0.775695
char: 05, frequency: 2.068520
char: 06, frequency: 5.494505
char: 07, frequency: 5.106658
char: 0c, frequency: 1.486749
char: 0d, frequency: 0.193924
char: 0f, frequency: 0.193924
char: 10, frequency: 10.601164
char: 11, frequency: 2.973497
char: 12, frequency: 0.969619
char: 13, frequency: 1.745314
char: 14, frequency: 7.433743
char: 16, frequency: 3.296704
char: 17, frequency: 1.163542
char: 18, frequency: 1.874596
char: 19, frequency: 2.262444
char: 1a, frequency: 5.946994
char: 1b, frequency: 4.718811
char: 1c, frequency: 5.817712
```

Decryption the .memorandum.txt.swp (Cont.)

- First I determined the most high frequency character is 'e', but it's not true.
- Second, try the second high frequency character is 'e', and decrypt it successfully.
- The using tool is programmed by my self
 - \$./decrypt_xor

Get Hash Key

- There is myphpadmin login account and password. Log in to the phpmyadmin page get the hashed password value.

```
+ [master]
[XD] % cat result.txt
2016.01.13
phpMyAdmin Account & Password
Account: BobIsGod
Password: distentionspracticalssturdiness

2015.12.15
- Pencial
- Eraser
- Ruler

2013.11.30
I forget to bring my mo
And my mom was pretty a

2010.10.22
Go to the zoo with my p
Lion, sheep, dog, rabbi
f course the horses.

2014.03.15
```

編輯	複製	刪除	3	おだ のぶなが	My life is brilliant. My...	NULL
編輯	複製	刪除	4	山本五十六	Oda Nobunaga (織田 信長 About this sound Oda Nobunaga ...	NULL
編輯	複製	刪除	5	This is not what you're looking for...	山本 五十六 (やまもと いそろく、1884年 (明治17年) 4月4日 - 1943年 (昭和18年) 4月...	NULL
編輯	複製	刪除	6	Fake stay night saying	Not this post... Please try another post...	NULL
編輯	複製	刪除	7	My Lovely Girlfriend!!	People die if they are killed!!	NULL
編輯	複製	刪除			(p` collide with `3455b9824a3ac89b`	

Get target!



Summary – What I Learned

- **Robots.txt**

- 在這次作業以前，並不曉得在**web**上有這種機制去避免預覽網頁內容的方法，在這次作業中不但認識了這個機制，並了解了要使用這個機制必須避免一些問題的產生。

- **Base64**

- 這種編碼方式，能夠使得一些unprintable character以其他字元表示出來，似乎在**WEB**中受到廣泛的使用，

Summary – What I Learned

- Xor encryption decryption
 - 只要藉由去假設password length，當在英文語系下，資料樣本夠大的時候就可以藉由frequency analyze去破解此密碼。
- MySQL323 hash function
 - 由於此function前8byte output與後8byte的output，沒有任何的相依性，以及其簡易的運算導致此hash function unsafe，能夠網路上可以查詢到一些工具使得此hash function能夠簡單的被破解。

Summary – How to prevent the vulnerability in Bob's blog

1. 不要將 backup file 放在 public web server 的 document root 底下.
 - 等同將自己的 source code open 至網路上。
2. 記得移除所有的 tmp file，在 linux 上每次檢查時可以執行 `ls -al` 以及 `grep` 搭配 regular expression 的方式確認是否有未刪除乾淨的 tmp file.
3. 不要使用容易遭到破解的加密方式
 1. Xor encryption
 2. Mysql323 hash