Hacky Easter 2015 – Write-up

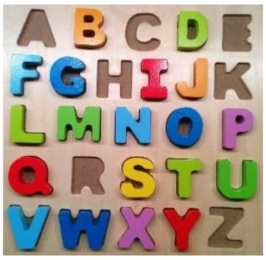
by Eash# - eduardo.sh@gmail.com

Challenge 01

Puzzword

Warming up with a simple one...

Search out the password in the image. Then jump up twice and grab the egg in the Egg-O-Matic below.



Answer:

Using missing letters “A,C,E,H,K,R,Z” of the board, you need to solve the anagram.

Used http://anagram-solver.net/ that output the correct answer “hackerz”

Challenge 02

It's in the Media

New York Times, March 12 2015

An Easter Egg of the famous Hacky Easter white-hat hacking competition, was leaked last Tuesday by the famous hacker group "Bunnonymous". Experts confirmed its authenticity, but could not crack the code hidden it in yet.

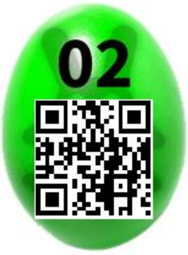
Can you do better?

Answer:

After initial analysis, I understood that the QRCode was produced using <style> tag.

```
<style>
.page { background-color: white !important;} .h { display:none;} .i3 { height: 4%; width: 4%; background: #fff;}
.o2 { height: 4%; width: 4%; background: #000;}.l1 { height: 4%; width: 4%; background: #000;}
.x5 { height: 4%; width: 4%; background: #fff;}@media print {body {-webkit-print-color-adjust: exact;}
.h { display:block;}.l1 { height: 4%; width: 4%; background: #000;}.x5 { height: 4%; width: 4%; background: #000;}}
</style>
```

Saving the page and replacing .x5 by .o2 on the html code solved the puzzle.



Challenge 03

Lego Stego

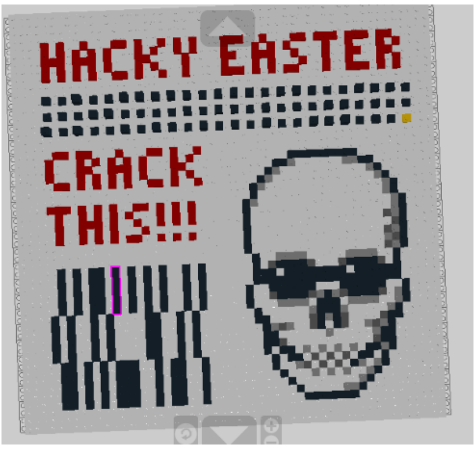
You intercepted a message sent by a nerd from the office nearby. On first sight, there's nothing suspicious in it. However, you are almost sure that something secret must be hidden in it. Can you find out what?

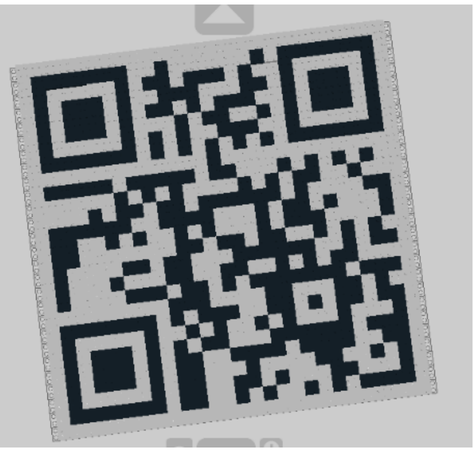
Answer:

This puzzle drove me crazy.

The QrCode was hidden on the low layer of the Lego Model.

Using LDD tool and removing the bricks on the top layer the QRCode is reveled.





Challenge 04

Twisted Num63rs

Your teacher was right when he said that math is useful in your whole life.
Calculate/convert the following values, and sort then in ascending order.

Answer:

Value	Sorted Value
sqrt 1296	36
PI^pi	36
ZmlmdHk=	50
Middle C[Hz]	262
10101111000	1400
303240 _ 8	100000
2^20	1048576
13MiB	13631488
Speed of light	299792458
127.0.0.1 Integer	2130706433
java.lang.Integer.MAX_VALUE	2147483648
8YiB	9,7E+24

Challenge 05

Phone Fumbling

In this challenge, you need to play with your phone a bit. Try to find out what controls the four bars, and make then reach the full width(all at the same time).

Answer:

Yes, I played a lot on my iphone ☺

Challenge 06

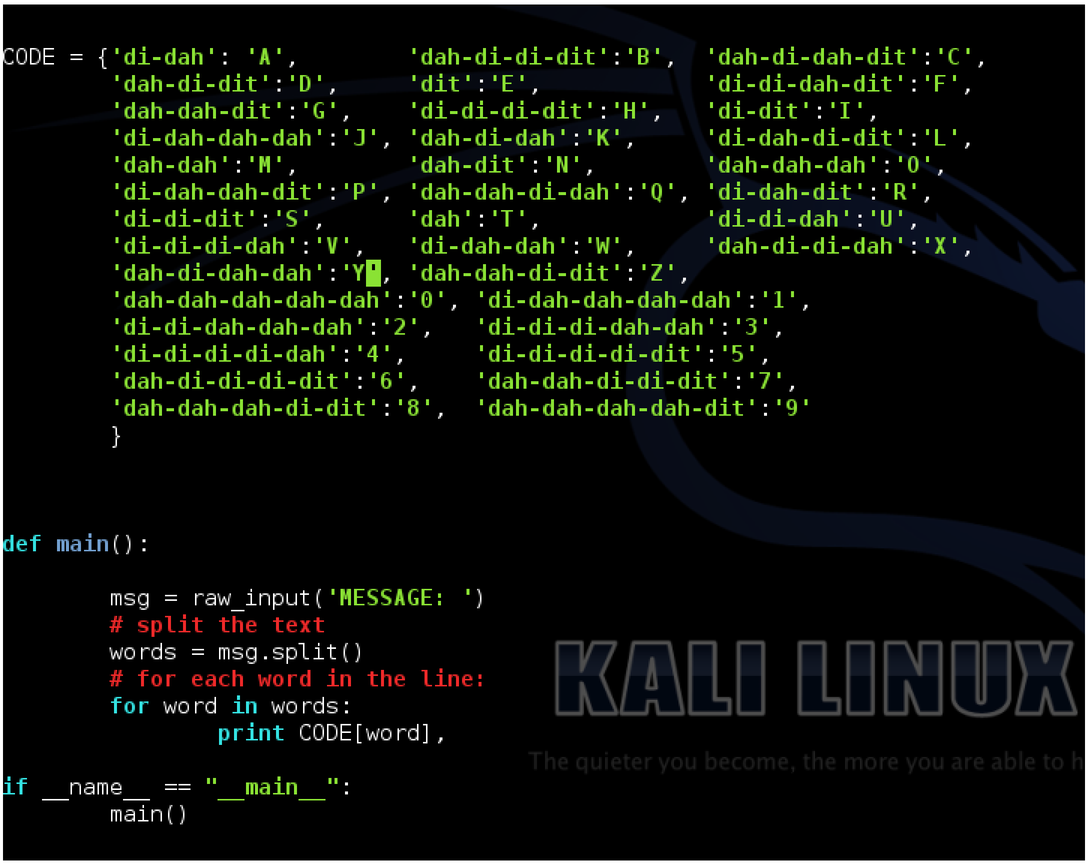
Hack to the Future

The Doc's in trouble again, and you must come to his rescue! As you jump into his time machine, you realize that a password is needed to start it. Just in that moment of despair, you receive an audio message from the Doc, through space and time:

dah-dah-dit dit dah-dah-dah di-dah-dit dah-dah-dit dit dah-dah dah-di-dah-dit di-di-dah-dit di-dah-di-dit dah-di-dah-dah

Answer: Part1

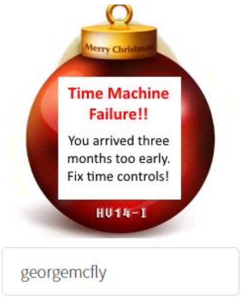
I developed a script in python to translate the Morse code.



```
CODE = {'di-dah': 'A',        'dah-di-di-dit':'B',  'dah-di-dah-dit':'C',
        'dah-di-dit':'D',     'dit':'E',            'di-di-dah-dit':'F',
        'dah-dah-dit':'G',    'di-di-di-dit':'H',   'di-dit':'I',
        'di-dah-dah-dah':'J', 'dah-di-dah':'K',     'di-dah-di-dit':'L',
        'dah-dah':'M',        'dah-dit':'N',        'dah-dah-dah':'O',
        'di-dah-dah-dit':'P', 'dah-dah-di-dah':'Q', 'di-dah-dit':'R',
        'di-di-dit':'S',      'dah':'T',            'di-di-dah':'U',
        'di-di-di-dah':'V',   'di-dah-dah':'W',     'dah-di-di-dah':'X',
        'dah-di-dah-dah':'Y', 'dah-dah-di-dit':'Z',
        'dah-dah-dah-dah-dah':'0', 'di-dah-dah-dah-dah':'1',
        'di-di-dah-dah-dah':'2',   'di-di-di-dah-dah':'3',
        'di-di-di-di-dah':'4',     'di-di-di-di-dit':'5',
        'dah-di-di-di-dit':'6',    'dah-dah-di-di-dit':'7',
        'dah-dah-dah-di-dit':'8',  'dah-dah-dah-dah-dit':'9'
        }



def main():

        msg = raw_input('MESSAGE: ')
        # split the text
        words = msg.split()
        # for each word in the line:
        for word in words:
                print CODE[word],

if __name__ == "__main__":
        main()
```

```
MESSAGE: dah-dah-dit dit dah-dah-dah di-dah-dit dah-dah-dit dit dah-dah dah-di-dah-dit di-di-dah-dit di-dah-di-dit dah-di-dah-dah

G E O R G E M C F L Y
```

Answer: Part2



Merry Christmas

Time Machine Failure!!

You arrived three months too early. Fix time controls!

HV14-I

georgemcfly

The simple date change in the computer (3 months ahead) revel the Easter egg 06.

Challenge 07

Vista de la calle

This egg is hidden in a street-view like viewer. Peek around the area and fit it!

Answer: Part1

The barcode is in the sky, or opening the Android App in the file **quito2_u.jpg**

Answer: Part2

To read the barcode you need to fix the border line. I did using Gimp.

Challenge 08

Spread the Sheet

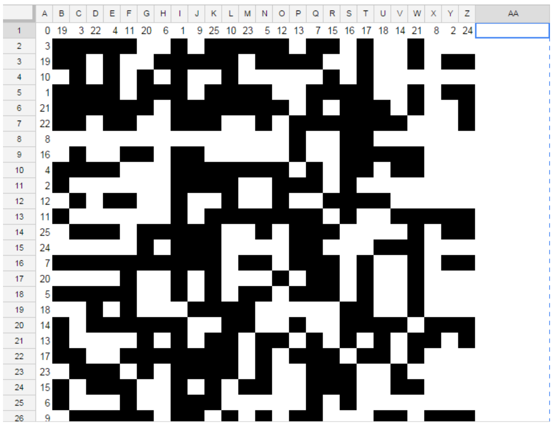
This egg is hidden within an online spreadsheet. Go find it's URL, and extract the egg out of it.

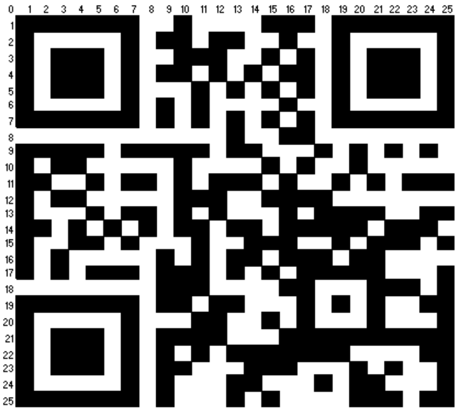
Spreadsheet ID:
1QPkfrnSVRAhQKL7AZx_HVXWrRXDvwCnVX2ih0jYp1CA

Answer:

Accessing online google spreadsheet on:

https://docs.google.com/spreadsheets/d/1QPkfrnSVRAhQKL7AZx_HVXWrRXDvwCnVX2ih0jYp1CA/edit#gid=0





To solve the puzzle the trick is ordering the lines and columns in ascended way.

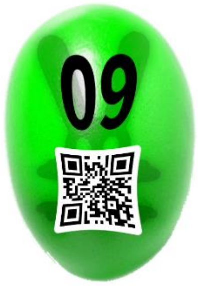
Challenge 9

Fish eye

Egg number nine is hidden here in the app. You've already seen it, haven't you? Go catch it and squint like a fish.

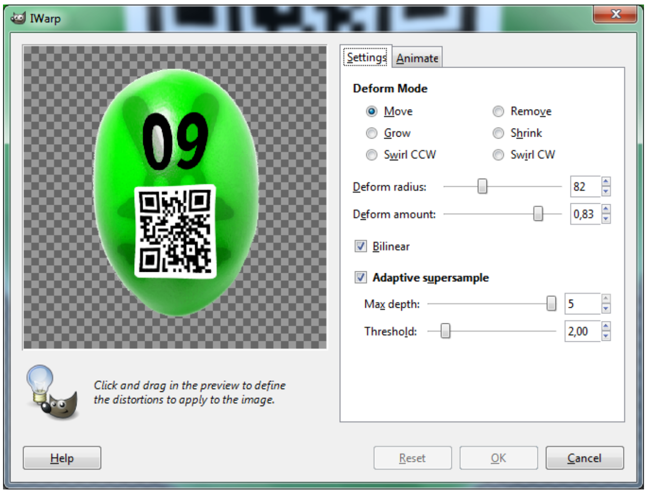
Answer: Part 1

You can figure out the egg when the app start up, or in the App with file name id_launcher.png.



Answer: Part 2

To scan the Egg nine you need to fix the "fish eye" effect. I did using Gimp>Filters>Distorts>IWarp.



Deform Mode

● Move ○ Remove
○ Grow ○ Shrink
○ Swirl CCW ○ Swirl CW

Deform radius: 82

Deform amount: 0,83

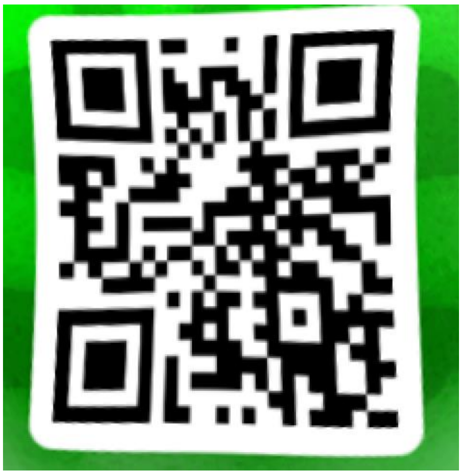
☑ Bilinear

☑ Adaptive supersample

Max depth: 5

Threshold: 2,00

Click and drag in the preview to define the distortions to apply to the image.



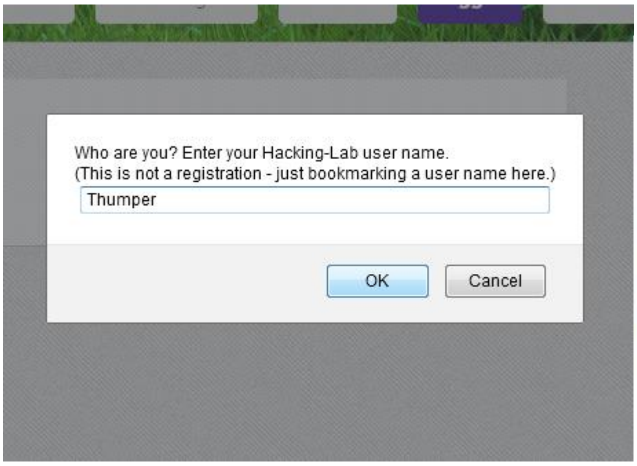
Challenge 10

Thumper's Den

In order to get this egg, you need to search on the web site. Rumors say that Thumper himself has bagged it.

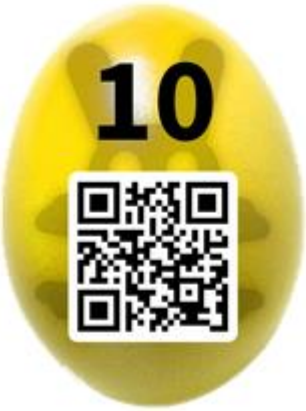
Answer:

Go to http://hackyeaster.hacking-lab.com/hackyeaster/eggs.html and change your name by Thumper and egg 10 is reveled.



Who are you? Enter your Hacking-Lab user name.
(This is not a registration - just bookmarking a user name here.)

Thumper



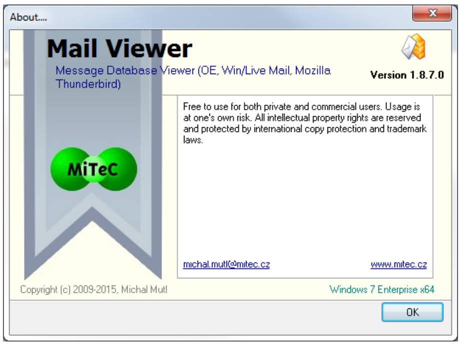
Challenge 11

You've got Mail

You caught a Thunderbird mailbox, which contains an easter egg. Go find it!

Answer:

Using Mail Viewer tool was possible carve the Egg eleven hidden on signature zip file.



Mail Viewer

Message Database Viewer (OE, Win/Live Mail, Mozilla Thunderbird)

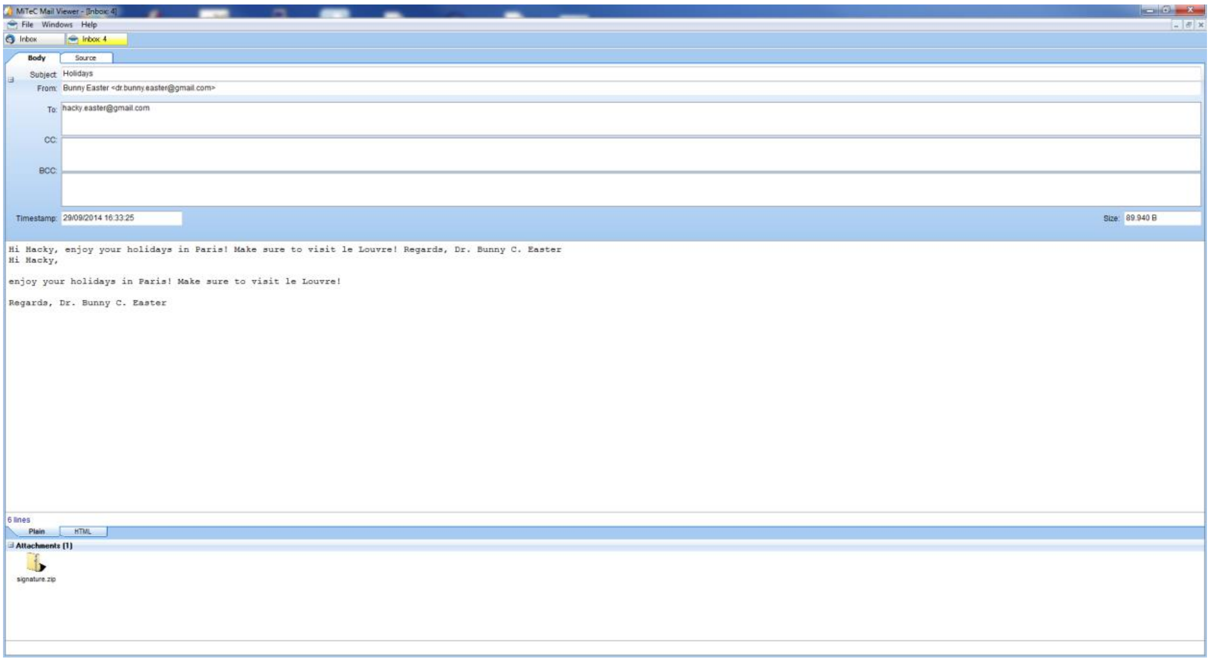
Version 1.8.7.0

Free to use for both private and commercial users. Usage is at one's own risk. All intellectual property rights are reserved and protected by international copy protection and trademark laws.

michal.mutl@mitec.cz www.mitec.cz

Copyright (c) 2009-2015, Michal Mutl

Windows 7 Enterprise x64



Subject: Holidays
From: Bunny Easter <dr.bunny.easter@gmail.com>
To: hacky.easter@gmail.com
CC:
BCC:
Timestamp: 29/09/2014 16:33:25
Size: 89.940 B

```
Hi Hacky, enjoy your holidays in Paris! Make sure to visit le Louvre! Regards, Dr. Bunny C. Easter
Hi Hacky,

enjoy your holidays in Paris! Make sure to visit le Louvre!

Regards, Dr. Bunny C. Easter
```

6 lines

Attachments (1)

signature.zip

Challenge 12

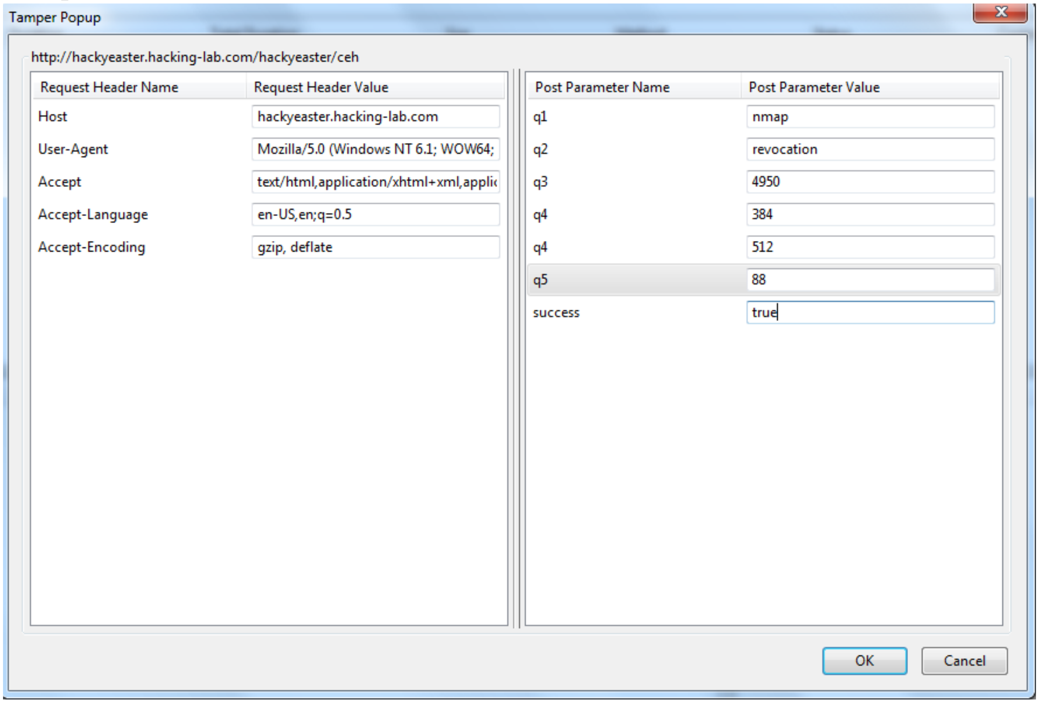
This is just a Test

This is your chance to become a Certified Easter Hacker (CEH)! Complete the following little test. Passing score is 100%.

Question 1
What is the name of the popular port scanner, implemented by Fyodor?
Question 2
In the context of PKI systems, the shorthand "CRL" stands for "certificate __________ list".
Question 3
A group of 100 people plans to use symmetric encryption for secure communication. How many keys are needed to let everybody communicate with each other?
Question 4
Which hash sizes are supported by the SHA2 family? Choose two!
Question 5
Which port number is used by Kerberos?

Answer:

Using Firefox tamper add-on to manipulate the data and put the correct values the Egg 12 was presented.



http://hackyeaster.hacking-lab.com/hackyeaster/ceh

Request Header Name	Request Header Value
Host	hackyeaster.hacking-lab.com
User-Agent	Mozilla/5.0 (Windows NT 6.1; WOW64;
Accept	text/html,application/xhtml+xml,applic
Accept-Language	en-US,en;q=0.5
Accept-Encoding	gzip, deflate

Post Parameter Name	Post Parameter Value
q1	nmap
q2	revocation
q3	4950
q4	384
q4	512
q5	88
success	true

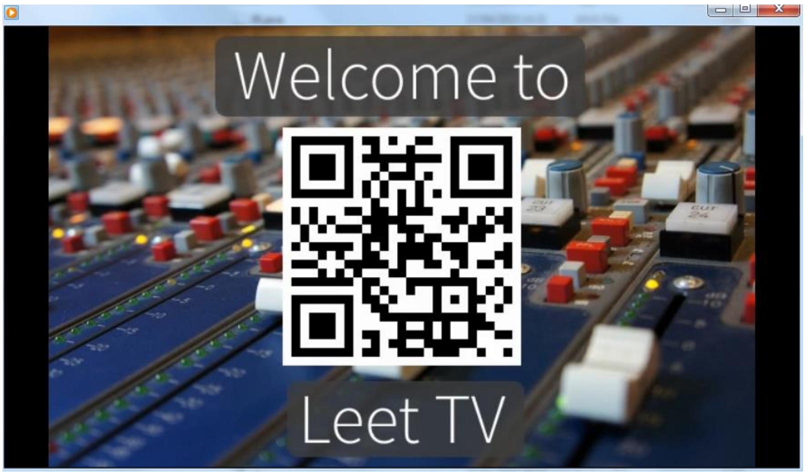
Challenge 13

Leet TV

Welcome to Leet TV! Click the image below to play our trailer.

Answer: Part 1

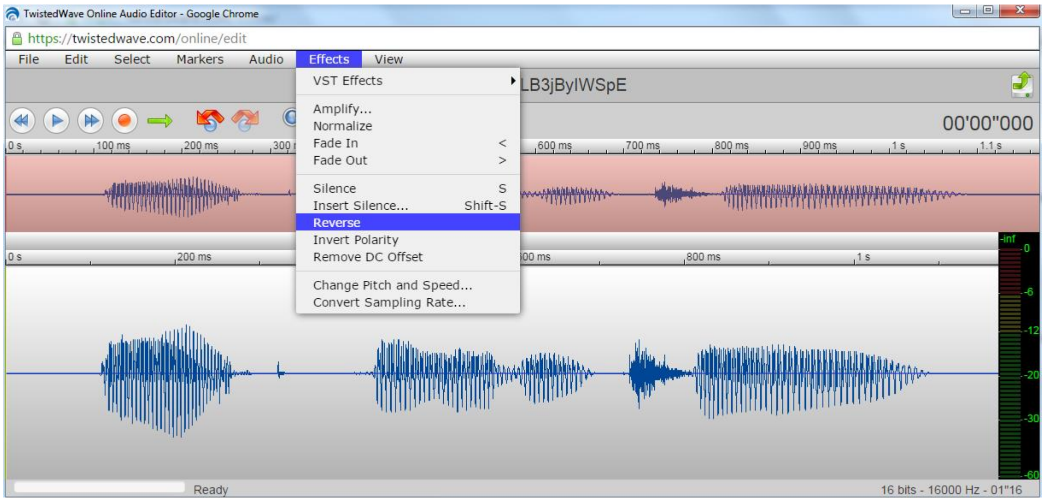
LEET = 1337 . Figured out the QRCode at 13:37 time movie.

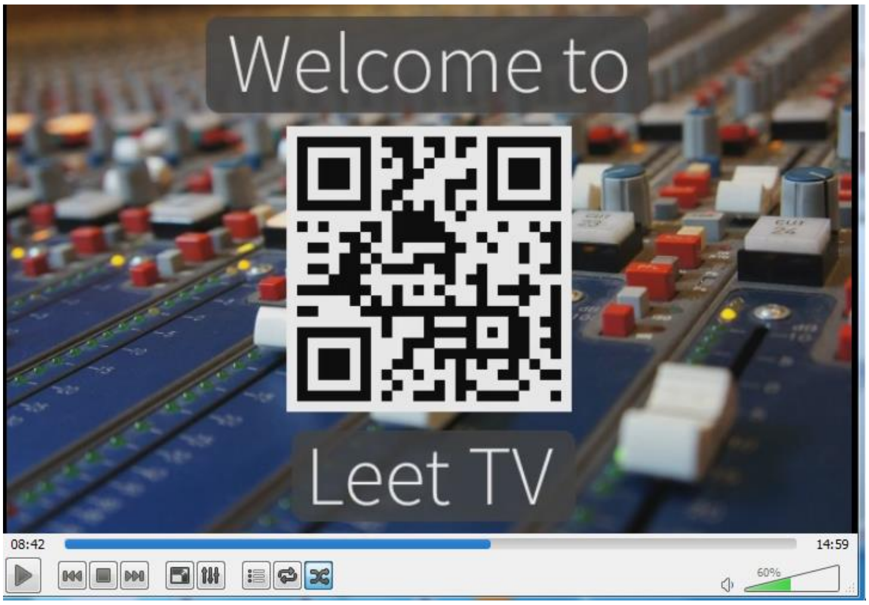


The QRCode points to url http://bit.ly/1BJENx8
http://hackyeaster.hacking-lab.com/hackyeaster/leettv_qbEtJZKLTLB3jByIWSpE.wav

Answer: Part 2

Reversing qbEtJZKLTLB3jByIWSpE.wav audio file points to QRcode on 08:42 time. The Egg thirteen was revealed.





Challenge 14

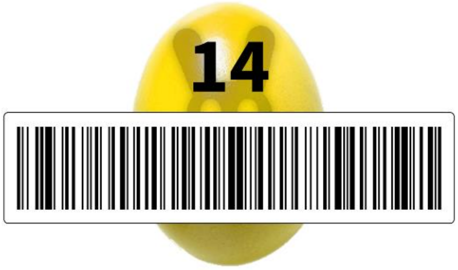
Wise Rabbit's Return

Wise Rabbit says:

An egg I give you for free, it's below, as you can see. But something got lost add a dimension you must!

Answer: Part 1

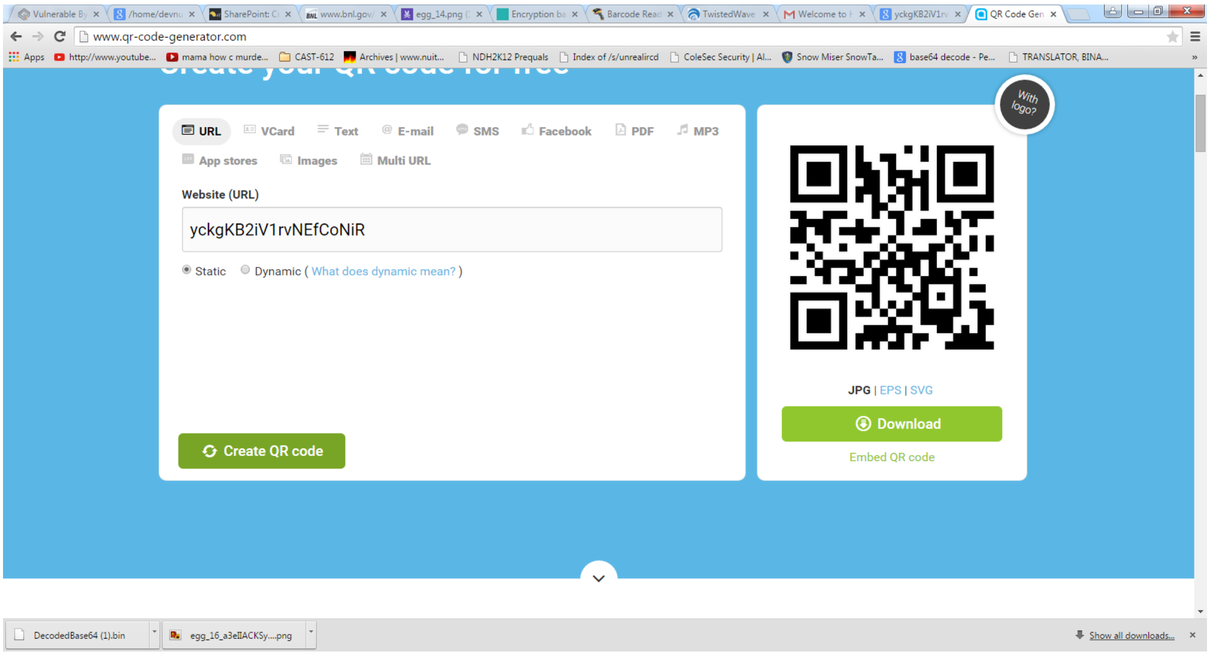
I read the barcode on the Egg 14 image.



Output: yckgKB2iV1rvNEfCoNiR

Answer: Part 2

Then converted the code “yckgKB2iV1rvNEfCoNiR” to QRCode format and voila.



Website (URL)

yckgKB2iV1rvNEfCoNiR

● Static ○ Dynamic (What does dynamic mean?)

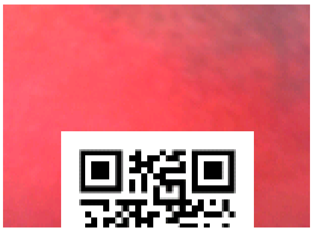
Challenge 15

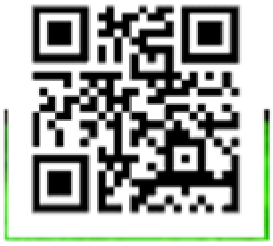
Photo Shooting

Your gallery needs some nice easter snapshots! What about a nice grasslands panorama, or a still life of a tomato?

Answer:

Following the instructions, I took pictures with “red” and “green” background, and merge the two half QRCode parts to get the Egg fifteen





Challenge 16

Ghost Room

Ghosts only come out when it's dark...

Answer: Part 1

Guessed the “dark” URL http://hackyeaster.hacking-lab.com/hackyeaster/challenge16-dark.html

Answer: Part 2

Decrypt the “GOST” ciphered message using the password “spooky” showed on “dark” URL image. I`ve used the online tool on http://www.tools4noobs.com/online_tools/decrypt/

```
d5++xytj6RiGwmqEecm63Kow7RZGAAHh
VFsksHFuj/Anap7pWHDZ1XQw8DAApUEN
R5ExOGUKTzGOtvSAlCHkHq6NneL6ZUTX
ej8Taxz+kHK9w9U8dxTOSksZ4HKS2YYD
```

Output:
http://hackyeaster.hacking-lab.com/hackyeaster/images/egg_16_a3eIIACKSy02sJ6LxXeh.png

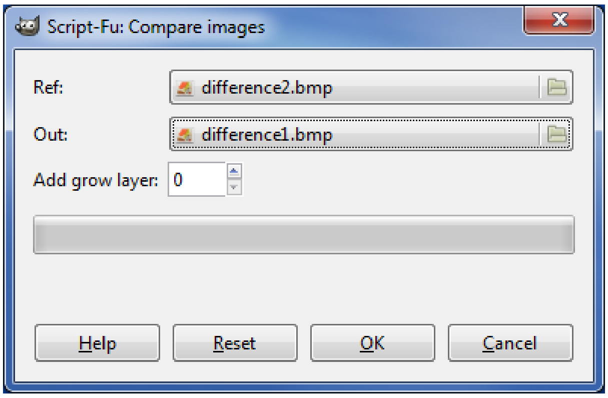
Challenge 17

Spot the Difference

Sharpen your eyes, and find the difference of the two images below!

Answer: Part1

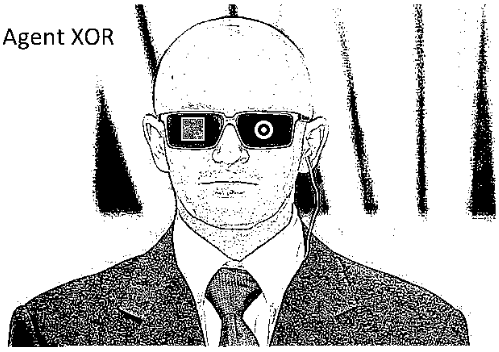
Using Gimp with Compare Script-Fu module to perform a XOR operation between the pictures difference1.bmp and difference2.bmp I’ve carved the hidden image “Agent XOR”.


Script-Fu: Compare images

Ref: difference2.bmp

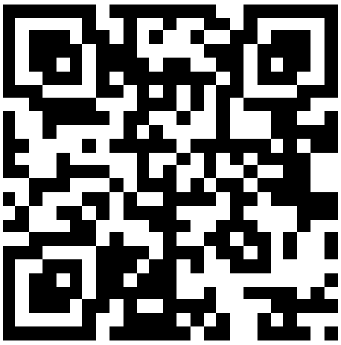
Out: difference1.bmp

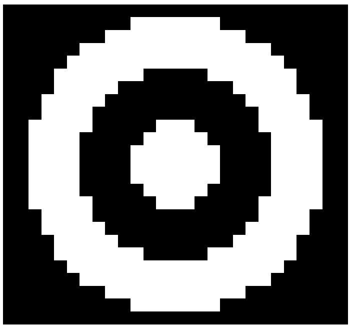
Add grow layer: 0

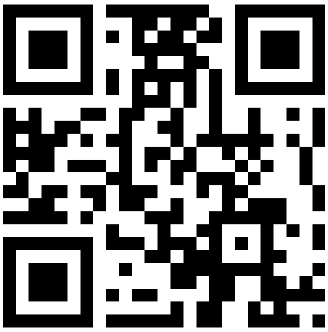


Answer: Part2

To solve the challenge one more XOR between the images on the shades lens was needed to revel the final QRcode.







Challenge 18

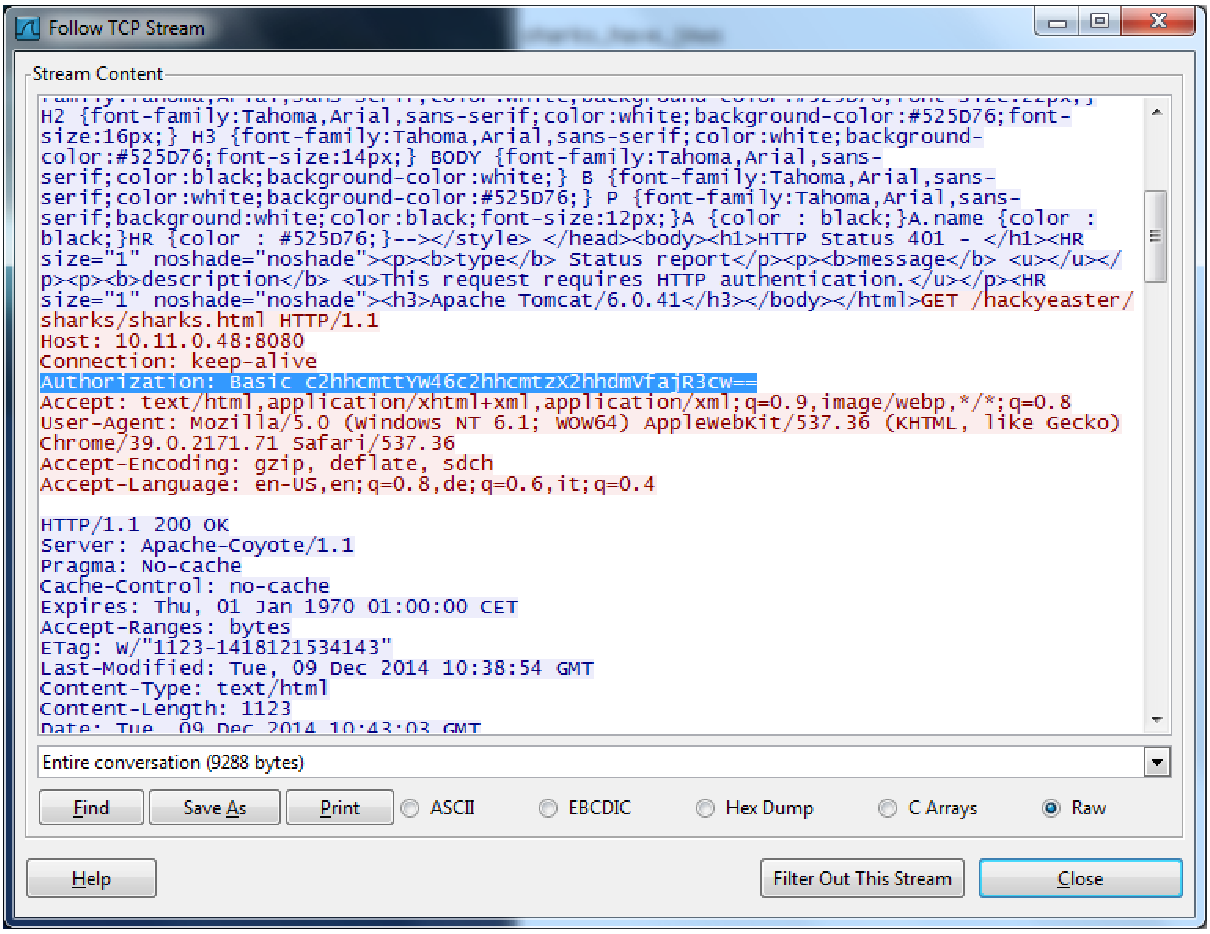
Sharks on Wire

In this challenge, you need to get access to a web site.

Answer: Part 1

I used Wireshark to analyze the pcap file. Converting the base64 Authorization code “c2hhcmttYW46c2hhcmtzX2hhdmVfajR3cw==” the first pair User:Password was reveled.

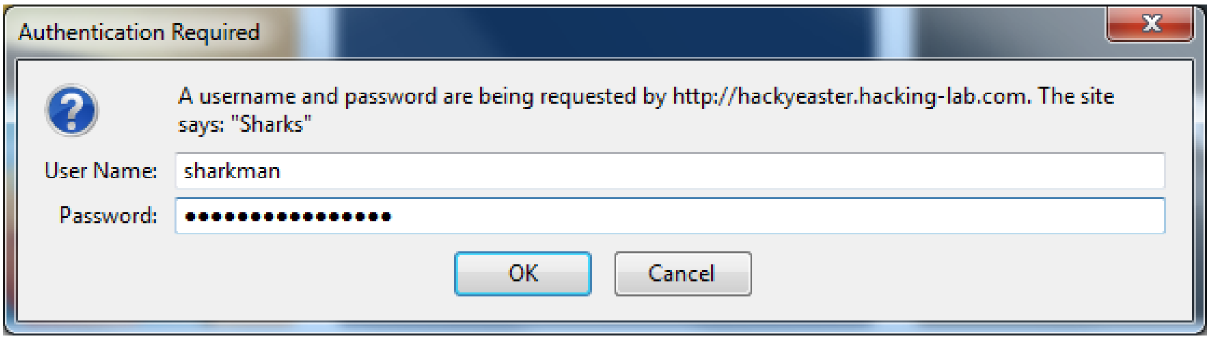
sharkman:sharks_have_j4ws



```
H2 {font-family:Tahoma,Arial,sans-serif;color:white;background-color:#525D76;font-
size:16px;} H3 {font-family:Tahoma,Arial,sans-serif;color:white;background-
color:#525D76;font-size:14px;} BODY {font-family:Tahoma,Arial,sans-
serif;color:black;background-color:white;} B {font-family:Tahoma,Arial,sans-
serif;color:white;background-color:#525D76;} P {font-family:Tahoma,Arial,sans-
serif;background:white;color:black;font-size:12px;}A {color : black;}A.name {color :
black;}HR {color : #525D76;}--></style> </head><body><h1>HTTP Status 401 - </h1><HR
size="1" noshade="noshade"><p><b>type</b> Status report</p><p><b>message</b> <u></u></
p><p><b>description</b> <u>This request requires HTTP authentication.</u></p><HR
size="1" noshade="noshade"><h3>Apache Tomcat/6.0.41</h3></body></html>GET /hackyeaster/
sharks/sharks.html HTTP/1.1
Host: 10.11.0.48:8080
Connection: keep-alive
Authorization: Basic c2hhcmttYW46c2hhcmtzX2hhdmVfajR3cw==
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/39.0.2171.71 Safari/537.36
Accept-Encoding: gzip, deflate, sdch
Accept-Language: en-US,en;q=0.8,de;q=0.6,it;q=0.4

HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Pragma: No-cache
Cache-Control: no-cache
Expires: Thu, 01 Jan 1970 01:00:00 CET
Accept-Ranges: bytes
ETag: W/"1123-1418121534143"
Last-Modified: Tue, 09 Dec 2014 10:38:54 GMT
Content-Type: text/html
Content-Length: 1123
Date: Tue, 09 Dec 2014 10:43:03 GMT
```

Entire conversation (9288 bytes)

Accessing http://hackyeaster.hacking-lab.com/hackyeaster/sharks/sharks.html with sharkman password.



A username and password are being requested by http://hackyeaster.hacking-lab.com. The site says: "Sharks"

User Name: sharkman

Password: ••••••••••••••••

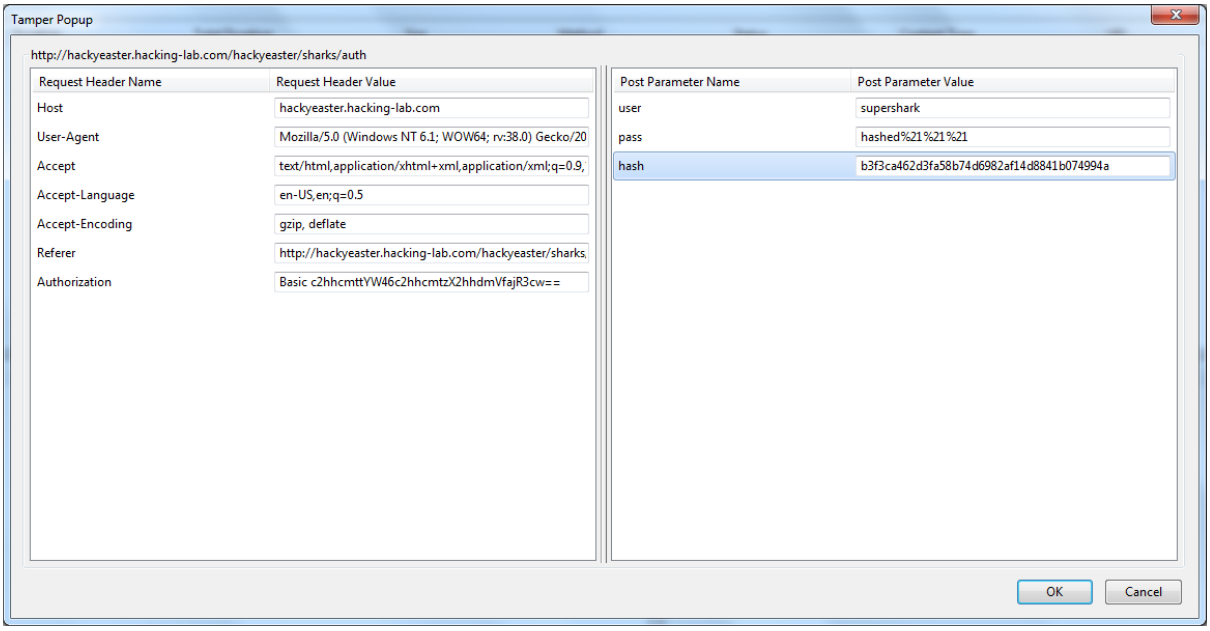
One more user:password pair is needed to move forward.

Answer: Part 2

Checking again the pcap file I've figured out a file named "Auth" with following content:

user=supershark&pass=hashed%21%21%21&hash=b3f3ca462d3fa58b74d6982af14d8841b074994a

Using Firefox tamper module to pass the hased password for supershark user, the URL with Egg eighteen was reveled.
http://hackyeaster.hacking-lab.com/hackyeaster/sharks/sharks_u83YOUgjSifjB8TDurz8.html



http://hackyeaster.hacking-lab.com/hackyeaster/sharks/auth

Request Header Name	Request Header Value
Host	hackyeaster.hacking-lab.com
User-Agent	Mozilla/5.0 (Windows NT 6.1; WOW64; rv:38.0) Gecko/20
Accept	text/html,application/xhtml+xml,application/xml;q=0.9,
Accept-Language	en-US,en;q=0.5
Accept-Encoding	gzip, deflate
Referer	http://hackyeaster.hacking-lab.com/hackyeaster/sharks,
Authorization	Basic c2hhcmttYW46c2hhcmtzX2hhdmVfajR3cw==

Post Parameter Name	Post Parameter Value
user	supershark
pass	hashed%21%21%21
hash	b3f3ca462d3fa58b74d6982af14d8841b074994a

OK
Cancel

18

Challenge 19

Cut'n'Place

Time for paper and scissors! The following PDF file contains some paper strips. Your task is to combine them in such a way that a passphrase appears. Once found, enter the passphrase in the Egg-O-Matic below.

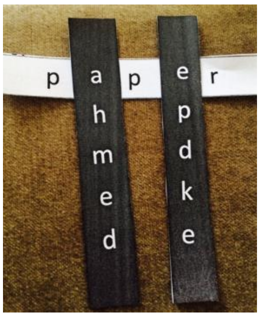
Hint: The passphrase does not use all characters available, and it has no spaces.

Answer:

This challenge almost drove me crazy. Congratulations to who ones have created it.

I`ve solved this challenge by guessing.  Three things called my attention, first the strips color, horizontal in white and vertical in black. I guess that the strips combination must create a chessboard. The second one was the stripe “p u p a r” where I`ve read the word “paper”. Third the symbols that must not be part of the solution. After was a trial-error method up to solve the challenge.

Solution: paperstripsmadebyshredder



Challenge 20

Lots of Bots

Robots have placed an egg on this web server. If you wanna find it, you need to think and act like a bot.

Answer: Part 1.

The hint on the challenge statement points to file robots.txt.
http://hackyeaster.hacking-lab.com/robots.txt

```
User-agent: EasterBot
Disallow: /
Allow: /hackyeaster/bots/bots.

User-agent: *
Disallow: /
```

```
I could see that only "EasterBot" user agent is allowed to access /bots/
directory. Then I used wget to download the bots.html page.
wget --user-agent="User-agent: EasterBot" -m http://hackyeaster.hacking-lab.com/hackyeaster/bots/bots.html
```

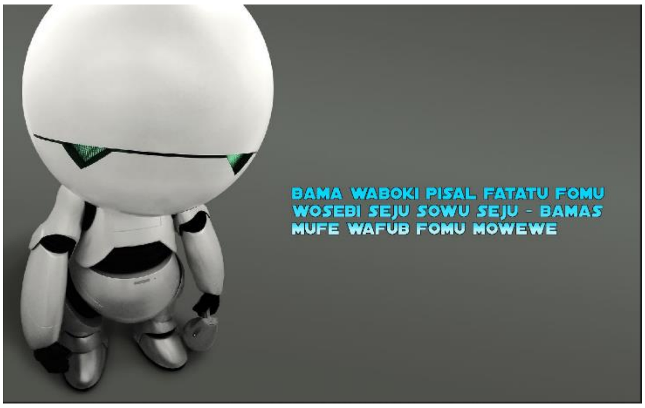
```
<html>
  <head>
            <title>Bots</title>
            <script type="text/javascript">
     eval(String.fromCharCode(105, 102, 32, 40, 33, 40, 110, 97, 118, 105, 103, 97, 116, 111, 114, 46, 117, 115, 101, 114, 65, 103, 101, 110, 116, 32, 61, 61, 61, 32, 39, 69, 97, 115, 116, 101, 114, 66, 111, 116, 39, 41, 41, 32, 123, 32, 108, 111, 99, 97, 116, 105, 111, 110, 46, 114, 101, 112, 108, 97, 99, 101, 40, 39, 104, 116, 116, 112, 58, 47, 47, 101, 110, 46, 119, 105, 107, 105, 112, 101, 100, 105, 97, 46, 111, 114, 103, 47, 119, 105, 107, 105, 47, 67, 45, 51, 80, 79, 39, 41, 59, 125));
    </script>
  </head>
  <body style="background: white; border: 20px solid white;">
    <div style="widht: 100%; height: 100%; background: url('./robotbg.jpg') no-repeat center center fixed; -webkit-background-size: contain; -moz-background-size: contain; -o-background-size: contain; background-size: contain;">&#160;</div>
  </body>

</html>
```

Coverting the decimal code to ascii you have:

```
if (!(navigator.userAgent === 'EasterBot')) { location.replace('http://en.wikipedia.org/wiki/C-3PO');}
```

Then, if your user Agent is “EsaterBot” the jpg background file is a allowed.

```
wget --user-agent="User-agent: EasterBot" -m http://hackyeaster.hacking-lab.com/hackyeaster/bots/robotbg.jpg
```



Answer: Part 2.

Here you need to translate the message from ROILA - Robot Interaction Language to English.

bama waboki pisal fatatu fomu wosebi   seju sowu seju - bamas mufe wafub fomu mowewe

= you must   make word   of   addition two and two -   this   be name of page
The hint points to **four.html** page.

Answer: Part 3.

Retrieving the page **four.html** using wget.

```
wget --user-agent="User-agent: EasterBot" -m http://hackyeaster.hacking-lab.com/hackyeaster/bots/four.html
```

The next hint is in the page source code. It is highlighted below.
The hint points to **ruof.html** page.

```
<html>
  <head>
        <title>Bots</title>
        <meta name="description" content="Robots talk in ROILA language: eman egap eht esrever tsum">
    <meta name="keywords" content="secret, page, robots, fun, hacky easter, blrt, five, beep">
        <script type="text/javascript">
      eval(String.fromCharCode(105, 102, 32, 40, 33, 40, 110, 97, 118, 105, 103, 97, 116, 111, 114, 46, 117, 115, 101, 114, 65, 103, 101, 110, 116, 32, 61, 61, 61, 32, 39, 69, 97, 115, 116, 101, 114, 66, 111, 116, 39, 41, 41, 32, 123, 32, 108, 111, 99, 97, 116, 105, 111, 110, 46, 114, 101, 112, 108, 97, 99, 101, 40, 39, 104, 116, 116, 112, 58, 47, 47, 101, 110, 46, 119, 105, 107, 105, 112, 101, 100, 105, 97, 46, 111, 114, 103, 47, 119, 105, 107, 105, 47, 67, 45, 51, 80, 79, 39, 41, 59, 125));
    </script>
  </head>
  <body style="background: white; border: 20px solid white;">
    <div style="widht: 100%; height: 100%; background: url('./robotbg2.jpg') no-repeat center center fixed; -webkit-background-size: contain; -moz-background-size: contain; -o-background-size: contain; background-size: contain;">&#160;</div>
  </body>
</html>
```

Answer: Part 4

Retrieving the page **ruof.html** using wget.

```
wget --user-agent="User-agent: EasterBot" -m http://hackyeaster.hacking-lab.com/hackyeaster/bots/ruof.html
```

```
<html>
  <head>
        <title>Bots</title>
        <script type="text/javascript">
     eval(String.fromCharCode(105, 102, 32, 40, 33, 40, 110, 97, 118, 105, 103, 97, 116, 111, 114, 46, 117, 115, 101, 114, 65,
103, 101, 110, 116, 32, 61, 61, 61, 32, 39, 69, 97, 115, 116, 101, 114, 66, 111, 116, 39, 41, 41, 32, 123, 32, 108, 111, 99, 97,
116, 105, 111, 110, 46, 114, 101, 112, 108, 97, 99, 101, 40, 39, 104, 116, 116, 112, 58, 47, 47, 101, 110, 46, 119, 105, 107,
105, 112, 101, 100, 105, 97, 46, 111, 114, 103, 47, 119, 105, 107, 105, 47, 67, 45, 51, 80, 79, 39, 41, 59, 125));
    </script>
  </head>
  <body style="background: white; border: 20px solid white;">
    <div style="position: absolute; left:50%; top: 50%; margin-left: -187px; margin-top: -187px; width: 375px; height: 375px;
background: url('./egg_20_j5fir8U6g8.png'); background-size: 375px 375px; background-repeat: no-repeat;">&#160;</div>
  </body>
</html>
```

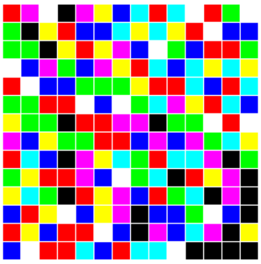
The Egg twenty is reveled in the ruof.html source code on URL:
Solution: http://hackyeaster.hacking-lab.com/hackyeaster/bots/egg_20_j5fir8U6g8.png

Challenge 21

Cony Code

Tired of boring QR codes, Dr. Bunny C. Easter developed an alternative. He's proudly introducing the "Cony Code" now! Crack the code in order to get another easter egg!

Hint: 110 is blue, the rest's up to you...



Answer:

Following the hint where 110 is blue is possible create the table below to other colors.

```
#ffffff    : '000',      #White
#ffff00  : '001',       #Yelow
#ff00ff  : '010',       #Magent
#ff0000  : '011',     #Red
#00ffff  : '100',       #Cyan
#00ff00  : '101',     #Green
#0000ff' : '110',     #Blue
#000000' : '111'    #Black
```

I did a python script to convert image to RGB, read the image in HEX, convert the HEX to BIN using the conversion table and convert BIN to ASCII.

The output is a URL that points to Egg twenty-one.

Solution: http://hackyeaster.hacking-lab.com/hackyeaster/images/egg_21_j7g67Z.png

Challenge 22

Hashes to Ashes

In this challenge, you need to prove your skills in hash cracking!

Standard algorithms (MD5, various SHA)
One iteration only, and no salting
For hashes 3 and 4, use the following word list

Hash 1: Numeric PIN (16 digits)
hint: numbers 1790
raLu6+eAmFelf2/uSy/67iTq57E=

Hash 2: Single word (lowercase only)
hint: Rick Astley
a791KNndKVmnr7N4mEJfZ1VfZ/Z3mHyufoYhCiyKDb38JI7C17JAEPRAutwiI7S1

Hash 3: Complex word (1 upper, 1 substitution, ending with punctuation + digit)
hint: http://xkcd.com/936/
uAgUxeDzhrBjcWP9iv6pKQ==

Hash 4: Multi-Word (lowercase only)
hint: http://xkcd.com/936/
l5HL4K6RmgMwmUota6Jrjww6HaFcc7zl/KOUlYgabJA=

Answer: Hash1

Viewing the hint image, I`ve guessed that the numbers used on the 16 long digits pin were 0179.
I`ve written a script to crack the hash.
The pin is **1199019170177790**

Answer: Hash2

Viewing the hint image, I`ve guessed that the words were related with Rick Astley records.
I`ve created a wordlist with all Rick`s records and written a script to crack the hash.
The word is **hopelessly**

Answer: Hash3

I`ve followed hint image from xkcd and the provided wordlist to written a script to crack the hash.
The word is **Disc0very.5**

Answer: Hash4

I`ve followed hint image from xkcd and the provided wordlist to written a script to crack the hash.
The word is **enginebulbgoatimportant**

Challenge 23

Beat the Nerd Master

Did you beat the Swordmaster in Monkey Island? Even if, it ain't gonna help you this time. Get to know the mighty **Nerd** Master!

Connect to port **1400** of hackyeaster.hacking-lab.com, and start the battle.

Here's an insult to start with: *Go 127.0.0.1 to your mummy.*

Answer:

First I`ve mapped all the insults and the answers. The second step was create a script in python to connect on the server and chat with it in automatic way. Due the short timeout is impossible answer the questions by hand. An attention point on the script is that the insults can't be used once only.

After completed the output point to Egg twenty-three URL.

http://hackyeaster.hacking-lab.com/hackyeaster/images/egg_23_j7vzfUzftszdf754fXDS.png

```
Connected to remote host. Start sending messages
Do you feel brave enough to challenge the mighty nerdmaster? (y|n)
y
OK, let's start, greenhorn. You won't have a chance. You may start!
---- YOUR TURN ----
Go 127.0.0.1 to your mummy.
Won't work. I only support IPv6.
---- MY TURN ----
I'll check you out - any last words?
svn:ignore
Arrgh! That's right.
Point for you! My health: 7, your health: 4
---- YOUR TURN ----
I bet you don't even understand binary.
Sure I do. Me and you, we are 10 different kind of persons.
---- MY TURN ----
You must be jealous when seeing my phone's display.
Not really - Your pixels are so big, some of them have their own region code!
Arrgh! That's right.
Point for you! My health: 6, your health: 4
---- YOUR TURN ----
format C:
Specified drive does not exist.
---- MY TURN ----
You'll be 0xdeadbeef soon.
Not as long as I have my 0xcafebabe.
Arrgh! That's right.
Point for you! My health: 5, your health: 4
---- YOUR TURN ----
This fight is like a hash function - it works in one direction only.
Too bad you picked LM hashing.
---- MY TURN ----
You're so slow, you must have been written in BASIC.
At least I don't have memory leaks like you.
Arrgh! That's right.
Point for you! My health: 4, your health: 4
---- YOUR TURN ----
1f u c4n r34d th1s u r s70p1d.
You better check your spelling. Stoopid has two 'o's.
---- MY TURN ----
Pna lbh ernq guvf?
EBG13 vf sbe ynzref.
Arrgh! That's right.
Point for you! My health: 3, your health: 4
---- YOUR TURN ----
Ping! Anybody there?
ICMP type 3, code 13: Communication Administratively Prohibited
---- MY TURN ----
Tell me your name, hobo. I need to check your records.
My name is bob'; DROP TABLE VALJ;--
Arrgh! That's right.
Point for you! My health: 2, your health: 4
---- YOUR TURN ----
I have more friends than you.
Yeah, but only until you update your Facebook profile with a real picture of you!
---- MY TURN ----
Af7ter th1s f1gh7, I w1ll pwn ur b0x3n.
Check your settings - you seem to have chosen the Klingon keyboard layout.
Arrgh! That's right.
Point for you! My health: 1, your health: 4
---- YOUR TURN ----
After loosing to me, your life won't be the same anymore.
A Life? Cool! Where can I download one of those?
---- MY TURN ----
You should leave your cave and socialize a bit.
I'm not anti-social. I'm just not user friendly.
Arrgh! That's right.
Respect! you've beaten the mighty nerd master! Here's your egg:
http://hackyeaster.hacking-lab.com/hackyeaster/images/egg_23_j7vzfUzftszdf754fXDS.png
```

Challenge 24

SHAM Hash

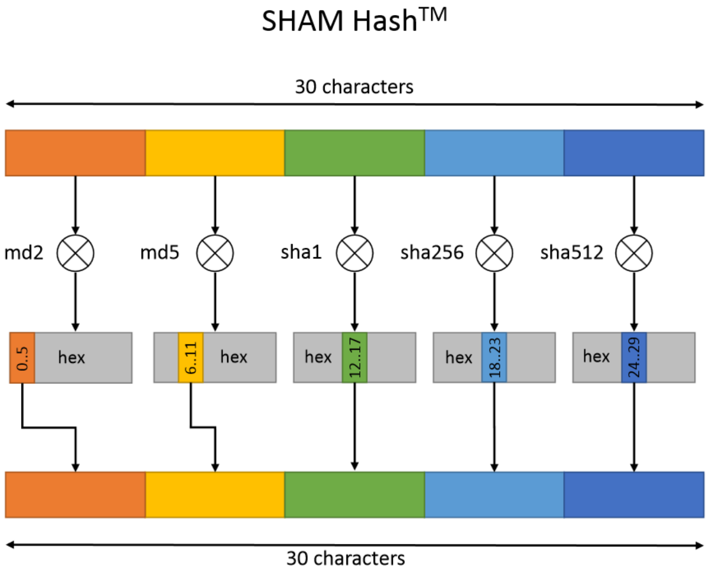
Crypto Chiefs Ltd. developed a new hash function, which takes a 'divide and conquer' approach and combines several well-known hash functions ("Split, Hash, And Merge"). The inventors claim that with this approach, their function becomes more secure. Can you prove they are wrong?

Create a string which produces the following hash:
757c479895d6845b2b0530cd9a2b11

Answer:

I've analyzed the specification on the image below and written a script to create a string that produced the hash: 757c479895d6845b2b0530cd9a2b11.

String: afpmqtaaqidtaww8ntangecaaf9pt3



f('hackyeaster2015isforeveryone!!') = 'd7d95bdc8b8c46e6b4a9217f7764d8'

Challenge 25

Jad & Ida

Jad and Ida are such a nice couple, don't you think?

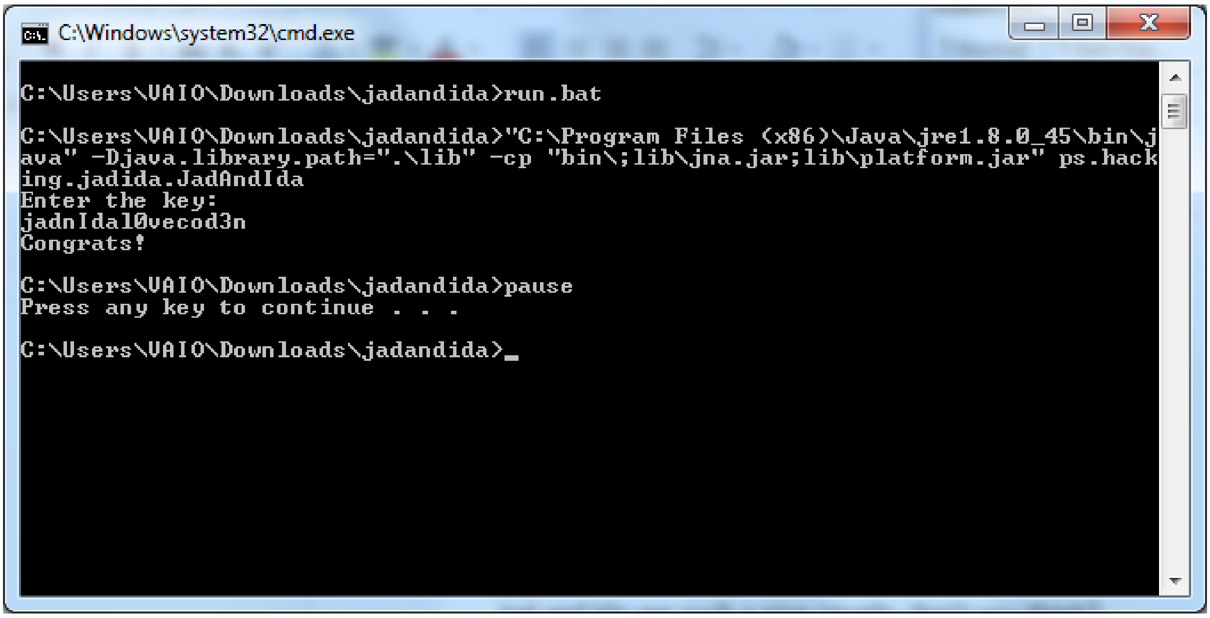
I`ve reversed four functions, two in Java and other two in ASM. I exported reversed disassembled dlls code using IDA Pro and java classes using JAD.

The goal was figured out k, such that "v3O] pmWm<Y(0=21".equals(h).

I’ve analyzed the disassembled code and written a python script do decode/decipher h = fizzle(rizzle(shizzle(bizzle(h))));

k = jadnIdal0vecod3n

Running the run.bat function with k password to decrypt s3cr3t.bin file and eggizzle_25.png was reveled.



```
C:\Users\VAIO\Downloads\jadandida>run.bat

C:\Users\VAIO\Downloads\jadandida>"C:\Program Files (x86)\Java\jre1.8.0_45\bin\j
ava" -Djava.library.path=".\lib" -cp "bin\;lib\jna.jar;lib\platform.jar" ps.hack
ing.jadida.JadAndIda
Enter the key:
jadnIdal0vecod3n
Congrats!

C:\Users\VAIO\Downloads\jadandida>pause
Press any key to continue . . .

C:\Users\VAIO\Downloads\jadandida>_
```

Challenge 26

Clumsy Cloud

Welcome to Clumsy Cloud ™ !
If your files are the eggs, then we are the hen ™.

We encrypt all your files, with a strong passphrase. The passphrase is kept securely in this app, protected by a PIN.

```
{
  "name" : "Clumsy Cloud Backup",
  "comment" : "Backup of your passphrase, protected with your secret PIN.",
  "params" : {
      "s" : "ovaederecumsale",
      "h" : "1.3.14.3.2.26",
      "i" : 10000,
      "k" : 128,
      "e" : "2.16.840.1.101.3.4.1.1",
      "p" : "8QeNdEdkspV6+1I77SEEEF4aWs5dl/auahJ46MMufkg="
    }
}
```

Answer:

I`ve analyzed the passphrase_backup.txt to understand the logic and use a java script to crack the correct PIN number.
Pin number is **7113** and the passphrase is **wirestarter54321.**

Challenge 27

Too Many Time Pad

You intercepted messages exchanged by evil Dr. Hopper and his agents. They used a One Time Pad for achieving perfect secrecy. Lucky for you, they have miserably failed, since the same key was used multiple times.

Check out the cipher texts, and try to decrypt them. Hint: The plain texts consist of lowercase letters and spaces only.

60c46964f83879618e2878de539f6f4a6271d716
63c37a6ca177792092602cc553c9684b
68d82c6bf4767f79dd617f9642d768057f63c1

6c8a7b6ce06a3161dd6a60d755d42d4d6d67
71c26929e96931698e2865d816d3624b687cd6
6cda6d6df87764709c6c7bd357d361556d77

Answer: Part 1

I solved challenge twenty-seven using the information of those two sites, a bit of good luck(guessing) and hard work.

http://travisdazell.blogspot.in/2012/11/many-time-pad-attack-crib-drag.html
http://www.mobilefish.com/services/one_time_pad/one_time_pad.php

Step 1:

I did XOR between two-ciphered texts.

68d82c6bf4767f79dd617f9642d768057f63c1
XOR
71c26929e96931698e2865d816d3624b687cd6
= 191a45421d1f4e1053491a4e54040a4e171f17

Step 2:
I did XOR between the output of XOR1 and “the” in hexadecimal value 746865

191a45421d1f4e1053491a4e54040a4e171f17
XOR
74686574686574686574686574686574686574
= mr 6uz:x6=r+ lo:⍰zc

At this point I guessed that the message was “mr bunny??????”

Step 3:
Kept guessing the phrases up to decipher the last one ciphered text with the secret “ipadyoupadweallpad” that reveled the Egg twenty-seven.

60c46964f83879618e2878de539f6f4a6271d716 -> enemy has the bonbo?
63c37a6ca177792092602cc553c9684b -> five oh oh seven
68d82c6bf4767f79dd617f9642d768057f63c1 -> mr bunny is the spy
6c8a7b6ce06a3161dd6a60d755d42d4d6d67 ->   i wear a black hat
71c26929e96931698e2865d816d3624b687cd6 -> the hq is in London
6cda6d6df87764709c6c7bd357d361556d77 ->   ipadyoupadweallpad

APPENDIX – SOURCE CODE

Challenge 21

```
from PIL import Image
import binascii

reps = {'#ffffff' : '000',     #White
    '#ffff00' : '001',     #Yelow
    '#ff00ff' : '010',     #Magent
    '#ff0000' : '011',     #Red
    '#00ffff' : '100',     #Cyan
    '#00ff00' : '101',     #Green
    '#0000ff' : '110',     #Blue
    '#000000' : '111'      #Black
}

def rgb2hex(r,g,b):
  return  '#{:02x}{:02x}{:02x}'.format(r,g,b)


def replace_all(text, dic):
  for i, j in dic.iteritems():
    text = text.replace(i, j)
  return text

#Opening Image code
img = Image.open("conycode.png")

#Converting image  to RGBA format
pixels = img.convert('RGBA').load()
width, height = img.size

#Lines vars
line16 = []
line32 = []
line48 = []
line64 = []
line80 = []
line96 = []
line112 = []
line128 = []
line144 = []
line160 = []
line176 = []
line192 = []
line208 = []
line224 = []

#Starting line reading lines in Hex value
for x in range(0,width,16):
    for y in range(0,height,16):
      r, g , b, a = pixels[x,y]
      if y <> 0:
        if x <> 0:
          if y == 16:
            line16.append(rgb2hex(r,g,b))
          elif y == 32:
            line32.append(rgb2hex(r,g,b))
          elif y == 48:
            line48.append(rgb2hex(r,g,b))
          elif y == 64:
            line64.append(rgb2hex(r,g,b))
          elif y == 80:
            line80.append(rgb2hex(r,g,b))
          elif y == 96:
            line96.append(rgb2hex(r,g,b))
          elif y == 112:
            line112.append(rgb2hex(r,g,b))
          elif y == 128:
            line128.append(rgb2hex(r,g,b))
          elif y == 144:
            line144.append(rgb2hex(r,g,b))
          elif y == 160:
            line160.append(rgb2hex(r,g,b))
          elif y == 176:
            line176.append(rgb2hex(r,g,b))
          elif y == 192:
            line192.append(rgb2hex(r,g,b))
          elif y == 208:
            line208.append(rgb2hex(r,g,b))
          elif y == 224:
            line224.append(rgb2hex(r,g,b))

#All list Concatenation
all_lists = sum([line16, line32, line48, line64, line80, line96, line112, line128, line144, line160, line176, line192, line208, line224], [])

#Formating all_lists
text = ''.join(all_lists)

#Replace HEX to BIN
bin = replace_all(text,reps)

#Need to Remove last 12 Chars to fit in 8-bit size
bin = bin[:-12]

#Print Result in ASCii
m = ''.join(chr(int(bin[i:i+8], 2)) for i in xrange(0, len(bin), 8))
print 'Solution: ' + m
```

Challenge 22

Hash 1

```
from datetime import datetime
import itertools
from Crypto.Hash import SHA1
import re
import base64
import hashlib

#HASH1
# ???
#HASH: raLu6+eAmFelf2/uSy/67iTq57E=
#SHA1: ada2eeebe7809857a57f6fee4b2ffaee24eae7b1
#HINT
charset = '0179'
hash1 = "raLu6+eAmFelf2/uSy/67iTq57E="
start_time = datetime.now()
sha1hasher = SHA1.new()

for i in itertools.product(charset, repeat=16):
# SHA1
    decoded_string = base64.b64decode(hash1).encode('hex')
    strSHA1 = SHA1.new(''.join(i)).hexdigest()
    if strSHA1 ==  decoded_string:
      print 'HASH1=' + ''.join(i)
      end_time = datetime.now()
      print('Duration: {}'.format(end_time - start_time))
      break
```

Hash 2

```
from datetime import datetime
import itertools
from Crypto.Hash import SHA384
import re
import base64
import hashlib

#HASH2 - a791KNndKVmnr7N4mEJfZ1VfZ/Z3mHyufoYhCiyKDb38JI7C17JAEPRAutwiI7S1
#SHA384: 6bbf7528d9dd2959a7afb37898425f67555f67f677987cae7e86210a2c8a0dbdfc248ec2d7b24010f440badc2223b4b5

#Hint Rick Asthley
hash2 = 'a791KNndKVmnr7N4mEJfZ1VfZ/Z3mHyufoYhCiyKDb38JI7C17JAEPRAutwiI7S1'
start_time = datetime.now()
sha384hasher = SHA384.new()

#SHA384
decoded_string = base64.b64decode(hash2).encode('hex')
file=open('rickroll.txt', 'r')
words=file.read().split("\n")
for line in words:
    strSHA384 = SHA384.new(line).hexdigest()
    if strSHA384 == decoded_string:
        print 'SHA384=' + line
        end_time = datetime.now()
        print('Duration: {}'.format(end_time - start_time))
        break
```

Hash 3

```
from datetime import datetime
from string import rstrip
import itertools
import string
from Crypto.Hash import MD4,MD5
import re
import base64
import hashlib

hash3 = 'uAgUxeDzhrBjcWP9iv6pKQ=='
leet = string.maketrans('abegiloprstz', '463611092572')
md4hasher = MD4.new()
md5hasher = MD5.new()
start_time = datetime.now()
charset = string.punctuation + '0123456789'

decoded_string = base64.b64decode(hash3).encode('hex')
file=open("wordlist.txt","r");
names=file.read().split("\n")
for line in names:
    upper = line.title()
    word = list(upper)
    for index, j in enumerate(word):
        s =  j.translate(leet)
        word1 = list(upper)
        word1[index] = s
        word2 = ''.join(word1)
        for i in itertools.product(charset, repeat=2):
            append = str(''.join(i))
            word3 =  word2 + append
            strMD5 = MD5.new(word3).hexdigest()
            if strMD5 == decoded_string:
                print 'HASh3 MD5=' + word3
                end_time = datetime.now()
                print('Duration: {}'.format(end_time - start_time))
                break
            strMD4 = MD4.new(word3).hexdigest()
            if strMD4 == decoded_string:
                print 'HASh3 MD4=' + word3
                end_time = datetime.now()
                print('Duration: {}'.format(end_time - start_time))
                break
```

Hash 4

```
from datetime import datetime
from string import rstrip
import itertools
import string
from Crypto.Hash import SHA256
import re
import base64
import hashlib

hash4 = 'l5HL4K6RmgMwmUota6Jrjww6HaFcc7zl/KOUlYgabJA='
start_time = datetime.now()
sha256hasher = SHA256.new()

decoded_string = base64.b64decode(hash4).encode('hex')
file=open("wordlist.txt","r");
names=file.read().split("\n")
for line in names:
    for i in itertools.product(names, repeat=4):
      word = str(''.join(i))
      #print str(''.join(i))
      #append = str(''.join(i))
      #word3 =  word2 + append
      strSHA256 = SHA256.new(word).hexdigest()
      #print strSHA256
      if strSHA256 == decoded_string:
        print 'HASh4 SHA256=' + word
        end_time = datetime.now()
        print('Duration: {}'.format(end_time - start_time))
        break
```

Challenge 23

```
# telnet program example
import socket, select, string, sys

def prompt() :
  sys.stdout.write('<You> ')
  sys.stdout.flush()



#main function
if __name__ == "__main__":

  if(len(sys.argv) < 3) :
    print 'Usage : python telnet.py hostname port'
    sys.exit()

  host = sys.argv[1]
  port = int(sys.argv[2])

  s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
  s.settimeout(15)


  # connect to remote host
  try :
    s.connect((host, port))
```

```
except :
    print 'Unable to connect'
    sys.exit()

print 'Connected to remote host. Start sending messages'
#prompt()

def readlines(sock, recv_buffer=4096, delim='\n'):
               buffer = ''
               data = True
               while data:
                                   data = sock.recv(recv_buffer)
                                   buffer += data

                                   while buffer.find(delim) != -1:
                                                      line, buffer = buffer.split('\n', 1)
                                                      yield line
               return


while 1:
    socket_list = [sys.stdin, s]

    # Get the list sockets which are readable
    read_sockets, write_sockets, error_sockets = select.select(socket_list , [], [])

    for sock in read_sockets:
        #incoming message from remote server
        if sock == s:
            buffer = ''
            data = True
            delim ='\n'
            i = 0
            a = 0
            b = 0
            c = 0
            d = 0
            e = 0
            f = 0
            g = 0
            h = 0
            j = 0
            k = 0
            l = 0
            m = 0
            n = 0
            o = 0
            p = 0
            while data:
                    data = sock.recv(4096)
                    buffer += data
                    while buffer.find(delim) != -1:
                            line, buffer = buffer.split('\n', 1)
                            print line
                            if line == ("Do you feel brave enough to challenge the mighty nerdmaster? (y|n)"):
                                    s.send("y" + "\n")
                                    print "y"
                            elif line == ("---- YOUR TURN ----"):
                                    if i == 0:
                                           print "Go 127.0.0.1 to your mummy." + "\n"
                                           s.send("Go 127.0.0.1 to your mummy." + "\n")
                                           i += 1
                                    elif i == 1:
                                           if a == 0:
                                                print "I bet you don't even understand binary." + "\n"
                                                s.send("I bet you don't even understand binary." + "\n")
                                                i += 1
                                           else:
                                                i += 1
                                                next
                                    elif i == 2:
```

```
    if b == 0:
        print "format C:" + "\n"
        s.send("format C:" + "\n")
        i += 1
    else:
        i += 1
elif i == 3:
    if c == 0:
        print "This fight is like a hash function - it works in one direction only." + "\n"
        s.send("This fight is like a hash function - it works in one direction only." + "\n")
        i += 1
    else:
        i += 1
elif i == 4:
    if d == 0:
        print "1f u c4n r34d th1s u r s70p1d." + "\n"
        s.send("1f u c4n r34d th1s u r s70p1d." + "\n")
        i += 1
    else:
        i += 1
elif i == 5:
    if e == 0:
        print "Ping! Anybody there?" + "\n"
        s.send("Ping! Anybody there?" + "\n")
        i += 1
    else:
        i += 1
elif i == 6:
    if f == 0:
        print "I have more friends than you." + "\n"
        s.send("I have more friends than you." + "\n")
        i += 1
    else:
        i += 1
elif i == 7:
    if g == 0:
        print "After loosing to me, your life won't be the same anymore." + "\n"
        s.send("After loosing to me, your life won't be the same anymore." + "\n")
        i += 1
    else:
        i += 1
elif i == 8:
    if h == 0:
        print "You're so slow, you must have been written in BASIC." + "\n"
        s.send("You're so slow, you must have been written in BASIC." + "\n")
        i += 1
    else:
        i += 1
elif i == 9:
    if j == 0:
        print "Af7ter th1s f1gh7, I w1ll pwn ur b0x3n." + "\n"
        s.send("Af7ter th1s f1gh7, I w1ll pwn ur b0x3n." + "\n")
        i += 1
    else:
        i += 1
elif i == 10:
    if k == 0:
        print "You'll be 0xdeadbeef soon." + "\n"
        s.send("You'll be 0xdeadbeef soon." + "\n")
        i += 1
    else:
        i += 1
elif i == 11:
    if l == 0:
        print "Pna lbh ernq guvf?" + "\n"
        s.send("Pna lbh ernq guvf?" + "\n")
        i += 1
    else:
        i += 1
elif i == 12:
    if m == 0:
            print "Tell me your name, hobo. I need to check your records." + "\n"
            s.send("Tell me your name, hobo. I need to check your records." + "\n")
            i += 1
        else:
            i += 1
    elif i == 13:
        if n == 0:
            print "You must be jealous when seeing my phone's display." + "\n"
            s.send("You must be jealous when seeing my phone's display." + "\n")
            i += 1
        else:
            i += 1
    elif i == 14:
        if o == 0:
            print "I'll check you out - any last words?" + "\n"
            s.send("I'll check you out - any last words?" + "\n")
            i += 1
        else:
            i += 1
    elif i == 15:
        if p == 0:
            print "You should leave your cave and socialize a bit." + "\n"
            s.send("You should leave your cave and socialize a bit." + "\n")
            i += 1
        else:
            i += 1

elif line ==("I bet you don't even understand binary."):
    print "Sure I do. Me and you, we are 10 different kind of persons." + "\n"
    s.send("Sure I do. Me and you, we are 10 different kind of persons." + "\n")
    a = 1
elif line ==("format C:"):
    print "Specified drive does not exist." + "\n"
    s.send("Specified drive does not exist." + "\n")
    b = 1
elif line ==("This fight is like a hash function - it works in one direction only."):
    print "Too bad you picked LM hashing." + "\n"
    s.send("Too bad you picked LM hashing." + "\n")
    c = 1
elif line ==("1f u c4n r34d th1s u r s70p1d."):
    print "You better check your spelling. Stoopid has two 'o's." + "\n"
    s.send("You better check your spelling. Stoopid has two 'o's." + "\n")
    d = 1
elif line ==("Ping! Anybody there?"):
    print "ICMP type 3, code 13: Communication Administratively Prohibited" + "\n"
    s.send("ICMP type 3, code 13: Communication Administratively Prohibited" + "\n")
    e = 1
elif line ==("I have more friends than you."):
    print "Yeah, but only until you update your Facebook profile with a real picture of you!" + "\n"
    s.send("Yeah, but only until you update your Facebook profile with a real picture of you!" + "\n")
    f = 1
elif line ==("You should leave your cave and socialize a bit."):
    print "I'm not anti-social. I'm just not user friendly." + "\n"
    s.send("I'm not anti-social. I'm just not user friendly." + "\n")
    g = 1
elif line ==("I'll check you out - any last words?"):
    print "svn:ignore" + "\n"
    s.send("svn:ignore" + "\n")
    h = 1
elif line ==("You must be jealous when seeing my phone's display."):
    print "Not really - Your pixels are so big, some of them have their own region code!" + "\n"
    s.send("Not really - Your pixels are so big, some of them have their own region code!" + "\n")
    j = 1
elif line ==("Tell me your name, hobo. I need to check your records." ):
    print "My name is bob'; DROP TABLE VALJ;--" + "\n"
    s.send("My name is bob'; DROP TABLE VALJ;--" + "\n")
    k = 1
elif line ==("Pna lbh ernq guvf?" ):
    print "EBG13 vf sbe ynzref." + "\n"
    s.send("EBG13 vf sbe ynzref." + "\n")
    l = 1
```

```
                elif line ==("You'll be 0xdeadbeef soon."):
                    print "Not as long as I have my 0xcafebabe." + "\n"
                    s.send("Not as long as I have my 0xcafebabe." + "\n")
                    m = 1
                elif line ==("Af7ter th1s f1gh7, I w1ll pwn ur b0x3n."):
                    print "Check your settings - you seem to have chosen the Klingon keyboard layout." + "\n"
                    s.send("Check your settings - you seem to have chosen the Klingon keyboard layout." + "\n")
                    n = 1
                elif line ==("You're so slow, you must have been written in BASIC."):
                    print "At least I don't have memory leaks like you." + "\n"
                    s.send("At least I don't have memory leaks like you." + "\n")
                    o = 1
                elif line ==("After loosing to me, your life won't be the same anymore."):
                    print "A Life? Cool! Where can I download one of those?" + "\n"
                    s.send("A Life? Cool! Where can I download one of those?" + "\n")
                    p = 1

        if not data :
            print '\nDisconnected from chat server'
            sys.exit()

    #user entered a message
    else :
        sys.stdout.write('y \n')
```

Challenge 24

```
import itertools
from Crypto.Hash import MD2,MD5,SHA1,SHA256,SHA512
import re
from datetime import datetime

charset = 'abcdefghijklmnopqrstuvwxyz0123456789'
md2hasher = MD2.new()
md5hasher = MD5.new()
sha1hasher = SHA1.new()
sha256hasher = SHA256.new()
sha512hasher = SHA512.new()
md2ok = 0
md5ok = 0
sha1ok  = 0
sha256ok = 0
sha512ok = 0

start_time = datetime.now()

for i in itertools.product(charset, repeat=6):
#MD2
    if md2ok == 0:
        strMD2 = MD2.new(''.join(i)).hexdigest()
        if strMD2.startswith('757c47'):
            md2 = ''.join(i)
            md2ok = 1
            print 'MD2=' + ''.join(i)
# MD5
    if md5ok == 0:
        strMD5 = MD5.new(''.join(i)).hexdigest()
        s = strMD5
        two = s[6:12]
        if two == '9895d6':
            md5 = ''.join(i)
            md5ok = 1
            print 'MD5=' + ''.join(i)

# SHA1
    if sha1ok == 0:
        strSHA1 = SHA1.new(''.join(i)).hexdigest()
        s = strSHA1
        two = s[12:18]
        if two == '845b2b':
            sha1 = ''.join(i)
            sha1ok =  1
            print 'SHA1=' + ''.join(i)

#SHA256
    if sha256ok == 0:
        strSHA256 = SHA256.new(''.join(i)).hexdigest()
        s = strSHA256
        two = s[18:24]
        if two == '0530cd':
            sha256 = ''.join(i)
            sha256ok = 1
            print 'SHA256=' + ''.join(i)

#SHA512
    if sha512ok == 0:
        strSHA512 = SHA512.new(''.join(i)).hexdigest()
        s = strSHA512
        two = s[24:30]
        if two == '9a2b11':
            sha512 = ''.join(i)
            sha512ok = 1
            print 'SHA512=' + ''.join(i)
    if (md2ok == 1)  and (md5ok == 1) and  (sha1ok == 1) and (sha256ok == 1) and (sha512ok == 1) :
        print md2 + md5 + sha1 + sha256 + sha512
        end_time = datetime.now()
        print('Duration: {}'.format(end_time - start_time))
        break
```

Challenge 25

```
import sys
import string

key = "v3O] pmWm<Y(0=21"

def func(x, y):
          x = ord(x)
          x -= 32
          n = 0
          while True:
                    xx = x + n * 91 + (27 - (y * y))
                    if 32 <= xx <= 122: break
                    n += 1
          x = xx
          x = chr(x)
          return x

def fizzle(k):
    kk = list(k)
    kk[:8], kk[8:] = kk[8:], kk[:8]
    for i in xrange(15,-1,-1):
        ch = kk[i]
        if ord(ch) > ord("z"): continue
        else: kk[i] = func(ch, i)
    return "".join(kk)

def rizzle(k):
    ret = []
    for c in k:
        if c in string.ascii_letters:
            if c.isupper(): ret.append(c.lower())
            elif c.islower(): ret.append(c.upper())
        else: ret.append(c)
    return "".join(ret)
```

```
def shizzle(k): return k[::-1]

def bizzle(k):
        ret = []
        for c in k:
                added = False
                for lst in [string.ascii_lowercase, string.ascii_uppercase]:
                        if c in lst:
                                ret.append(lst[(lst.index(c)-1+len(lst))%len(lst)])
                                added = True
                                break
                if not added: ret.append(c)
        return "".join(ret)

for i in xrange(10):
        key = bizzle(shizzle(rizzle(fizzle(key))))
print "Password:", key
```

Challenge 26

```
import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;
import javax.xml.bind.DatatypeConverter;
import javax.crypto.Cipher;
import javax.crypto.spec.SecretKeySpec;

class egg26
{
    private static String crypt(final String PIN)
    {
        try
        {
            SecretKeySpec localSecretKeySpec =
                new SecretKeySpec(hash(PIN, "ovaederecumsale", 10000), "AES");
            Cipher localCipher = Cipher.getInstance("AES");
            localCipher.init(2, localSecretKeySpec);
            String encodedString = "8QeNdEdkspV6+1I77SEEEF4aWs5dl/auahJ46MMufkg=";
            byte[] dec64 = DatatypeConverter.parseBase64Binary(encodedString);
            String str = new String(localCipher.doFinal(dec64));
            return str;
        }
        catch (Exception localException)
        {
            return null;
        }
    }

    public static byte[] hash(String paramString1, String paramString2, int paramInt) throws NoSuchAlgorithmException
    {
        MessageDigest localMessageDigest = MessageDigest.getInstance("SHA-1");
        byte[] arrayOfByte1 = (paramString2 + paramString1).getBytes();
        for(int n = 0;; n++)
        {
            if(n >= paramInt)
            {
                byte[] arrayOfByte2 = new byte[16];
                System.arraycopy(arrayOfByte1, 0, arrayOfByte2, 0, 15);
                return arrayOfByte2;
            }
            arrayOfByte1 = localMessageDigest.digest(arrayOfByte1);
        }
    }

    public static void main(String[] args)
    {
        for(int i=0; i<9999; i++)
        {
            String PIN = String.format("%04d", i);
            String ciphertxt = crypt(PIN);
            if (ciphertxt != null
            && ciphertxt.matches("^\\p{ASCII}*$"))
            {
                System.out.println(PIN+": "+ciphertxt);
                break;
            }
        }
    }
}
```