Lab 5: Ciphers and Crypto Fundamentals

Bill, Richard, Charley

Aim:

The aim of this lab is to give an introduction to ciphers, basic encoding/decoding techniques and frequency analysis, as to provide some fundamental understanding. Activities include decoding a range of ciphers and simple calculations.

Time to Complete:

4 hours (two supervised hours in the lab, and two additional unsupervised hours).

Learning activities:

At the end of this lab, you should understand:

- How to decode a range of ciphers.
- How to recognise certain encodings, such as Base-64, hex, and binary.
- How to write a Python script to crack PFX certificates. The PKCS#12 (PFX or P12) format is the binary format in which certificates are stored on a server. They are protected by a password and contain the public and private key (the key pair).
- How to perform bitwise calculations.
- How to perform frequency analysis.

Reflective statements (end-of-exercise):

- What do you think will happen when you do a frequency analysis of the bytes in an encrypted or compressed file?

- If we use a password to protect our key pairs on the server, what is the main weakness of this?

Lab 5: Ciphers and Crypto Fundamentals

Use your desktop computer to complete the following:

A Introduction

No	Description	Result
1	Go to: **http://asecuritysite.com/Challenges** Click on the "Start Challenge" button, and see if you can score over 30 points.	Your score:
2	Using: http://asecuritysite.com/Encryption/testprime Test for the following prime numbers:	91: [Yes] [No] 421: [Yes] [No] 1449: [Yes] [No]
3	Using: http://asecuritysite.com/Encryption/gcd Determine the GCD for the following:	88, 46: 105, 35:
4	Using: http://asecuritysite.com/coding/ascii Determine the Base 64 and Hex values for the following strings:	Hello: hello: HELLO:
5	Using: http://asecuritysite.com/coding/ascii Determine the following ASCII strings for these encoded formats:	bGxveWRz 6E6170696572 01000001 01101110 01101011 01101100 01100101 00110001 00110010 00110011
6	Using: http://asecuritysite.com/Coding/exor Determine the EX-OR of "hello" ex-ORed with the letter 't'	Hex: Base 64: Is the result printable in ASCII? [Yes][No]

7	What is the result of 53,431 mod 453?	
8	Generate a random number from: http://asecuritysite.com/Encryption/j	How many hex characters does the result have?
9	Try and crack some certificates from: http://asecuritysite.com/Encryption/certcrack What are the passwords for 'bill09.pfx', 'bill18.pfx', and 'country04.pfx'?	bill09.pfx: bill18.pfx: country04.pfx:

10. We can also create a short **Python script** to try to crack the same certificates.

Boot up your Kali VM, and download the following archive:

http://asecuritysite.com/public/certs.zip

Extract the certificates into the /root folder, and then move into that folder. Now use openssl to try a password:

```
openssl pkcs12 -nokeys -in bill01.pfx -passin pass:orange
```

Did you manage to run the script? What password is correct for bill01.pfx?

Now implement the Python script in Program 1.

```
from OpenSSL import crypto
words=[]
words.append("coconut")
words.append("mango")
words.append("apples")
words.append("apple")
words.append("oranges")
words.append("orange")
words.append("ankle")
words.append("password")
words.append("bill")
words.append("battery")

for passwd in words:

	try:
		p12 = crypto.load_pkcs12(open("fredpfx.pfx", 'rb').read(), passwd)

		certificate =p12.get_certificate()
		p12.get_privatekey()

	      print certificate.get_serial_number()
	      print certificate.get_issuer().get_components()
	      print certificate.get_signature_algorithm()
	      print ("Success: "+passwd)

	except Exception as ex:
		print (".")
```

Can adapt this script to crack some of the other certificates contained in the archive you have downloaded. Bill01.pdf to bill18.pdf are based on fruits (in lowercase), country01.pdf to country06.pdf are based on countries.

Outline the passwords of the certificates:

Can you modify the code so that it shows other details from the certificate, such as its public key, subject, version and “notBefore”, and “notAfter”.

Ref: https://pyopenssl.org/en/0.15.1/api/crypto.html#x509name-objects

B Frequency Analysis

Now see if you can crack the **five minute cracking challenge** for:

http://asecuritysite.com/challenges/scramb

C Character mapping

Complete the following table for each of the characters:

Char	Decimal	Binary	Hex	Oct	HTML
(Space)					
a					
}					
Ã					
ÿ					

D Test

1. Crack some Caesar codes at: http://asecuritysite.com/tests/tests?sortBy=caesar
2. Determine some hex conversions at: http://asecuritysite.com/tests/tests?sortBy=hex01
3. Determine some Base64 conversions: http://asecuritysite.com/tests/tests?sortBy=ascii01
4 Now complete the test at: http://asecuritysite.com/tests/tests?sortBy=crypto01