



THE UNIVERSITY OF THE WEST INDIES

INFO3155
INFORMATION ASSURANCE AND SECURITY
ASSIGNMENT 1

STUDENT NAME AND ID: KYLE STERLING - 620118420

FACULTY: SCIENCE AND TECHNOLOGY

CAMPUS: UWI MONA

INTRODUCTION

As an information security specialist working with a super-secret shady government organization (SSSGO) our overall aim is to stop a potential terror attack that government spies believe is set to happen in February. The mission is split into three different parts: mission 1, mission 2 and mission 3. You are given three files, two of which are password protected and each mission unlocks the subsequent one after it. So, lets head into saving some lives. All sources will be added to the reference document.

MISSION 1

Analysts from the super-secret shady government organization (SSSGO) intercepted some chatter on the dark web, which is believed to be some sort of encrypted code, used to access the files you received. The numbers appear in groups of threes. We strongly believe that the first number is the actual cyphertext (c), while the second and third make up the public key for each group. The mission here is to use the knowledge learnt in class to get the private key given the public key and to use it to decrypt the ciphertext which was given. After all the cypher is decrypted, the objective was to use the plaintext to figure out a password for the next mission (Mission 2).

The security concepts being used with RSA are confidentiality, authenticity, and accountability. RSA preserves authorized restrictions from unauthorized users, information is kept confidential between sender and receiver. Due to the nature of RSA, there exist only one private key which can decrypt a message that is encoded with the subsequent public key which matches that private designated to one set person. Thus, this provides accountability being that if the message or a document is signed with a person's private key then we know it is them for sure. This also provides authenticity being that is two persons, say Alice and Bob are sending a message back and forth which are encrypted in each other's public key, then Bob can be sure that it is Alice and vice versa seeing that only she could decrypt the messages using her public key.

Through completing the mission given the aspect of the CIA Triad violated was confidentiality. Since I was able to figure out the receiver's private key and used it to get unauthorized access to the information, I violated the two parties' confidentiality.

To method I took was a step-by-step approach. Realizing that I had the public key (e, n), I used python code to get the values of p and q which are numbers relatively prime to n. Based on research referenced from (RSA Algorithm - known n how to get p & q, n.d.), I found out

that to limit my search for values I could go from 1 to $n^{1/2}$ which is the same as the square root. If in that search I found a value which could mod n and give zero I would use it as my p, divide n by that q to give me q then go on to find phi using the formula given in class. The formula to find d is put in a loop to find the values for k and d. The structure is given in this document below and the code is also well commented to explain functionality. (How to get numbers after decimal point?, n.d.) was research was done to figure out how to get the only relevant figure for d rather than print all values.

1) Cipher Text = 1407, e = 17, n = 2173

$$p = 41, q = 53$$

$$\begin{aligned}\phi(n) &= (p - 1) * (q - 1) \\ &= (41 - 1) * (53 - 1) \\ &= 2080\end{aligned}$$

$$\begin{aligned}d &= (k * \phi(n) + 1)/e \\ &= (14 * 2080 + 1)/17 \\ &= 1713\end{aligned}$$

$$\begin{aligned}M &= C^d \bmod n \\ &= 1407^{1713} \bmod 2173 \\ &= 112\end{aligned}$$

2) Cipher Text = 129, e = 31, n = 377

$$p = 13, q = 29$$

$$\begin{aligned}\phi(n) &= (p - 1) * (q - 1) \\ &= (13 - 1) * (29 - 1) \\ &= 336\end{aligned}$$

$$\begin{aligned}d &= (k * \phi(n) + 1)/e \\ &= (25 * 336 + 1)/31 \\ &= 271\end{aligned}$$

$$\begin{aligned}M &= C^d \bmod n \\ &= 129^{271} \bmod 377 \\ &= 64\end{aligned}$$

3) Cipher Text = 196, e = 61, n = 1067

$$p = 11, q = 97$$

$$\phi(n) = (p - 1) * (q - 1)$$

$$\begin{aligned}
&= (11 - 1) * (97 - 1) \\
&= 960 \\
d &= (k * \phi(n) + 1)/e \\
&= (42 * 960 + 1)/61 \\
&= 661 \\
M &= C^d \bmod n \\
&= 196^{661} \bmod 1067 \\
&= 53
\end{aligned}$$

4) Cipher Text = 1648, e = 43, n = 2117

$$\begin{aligned}
p &= 29, q = 73 \\
\phi(n) &= (p - 1) * (q - 1) \\
&= (29 - 1) * (73 - 1) \\
&= 2044 \\
d &= (k * \phi(n) + 1)/e \\
&= (26 * 2044 + 1)/43 \\
&= 1219 \\
M &= C^d \bmod n \\
&= 1648^{1219} \bmod 2117 \\
&= 53
\end{aligned}$$

After decrypting the values, they were converted to the phrase
“p@55” using the ASCII table.

MISSION 2

The objective for mission two was to write python code to create a rainbow table by concatenating each salt of the salts given to the end of every password then hashing the combination. The table is then to be used to compare to the hash in the file with each hash in the rainbow table. When you find a match check to see which combination gave you the match and use that password to gain entry to the other password protected file. In summary, the goal was to use popular password and salt combinations to recreate the hash given to decode what it was.

Due to the irreversible nature of hashing, it provides the security concepts of integrity. Since the hash cannot be reversed it cannot be modified and each hash is unique based on the input used hence it cannot be duplicated. Based on this feature of not being able to be duplicated without knowing the plaintext, if any salt is used or even what the salt is used or the hashing scheme. Hashing also provides authenticity.

While completing the mission the aspects of the CIA triad which were violated are integrity and confidentiality. Since we were able to gain access to the information, which was restricted from us, confidentiality was broken. Also, being that the password and salt were able to be recreated without knowing the original plaintext, this duplication of the process used caused an infringement and the aspect of integrity was lost.

To understand the requirements of the mission I referred to the text (Computer Security Principles and Practice Fourth Edition) to gain a better understating of what a rainbow table was. After I gained a reasonable understanding of what it was I then when on to think of how to create it. My thoughts first brought me to consider creating an actual table of hashes and then checking if the hash given in mission two was among them if so, I would print the password and the salt used to create it. However, this was deemed to be inefficient, because it would require storing the hash as well as the password and salt used to create it. I

then concluded that it would be easier to loop through all the passwords and then within a second loop concatenate the salt at the end of each password and hash them. The resulted hash would then be compared and if it matched, I would then print the password. It was a challenge understanding how to use the hashlib python library, as such I referred to the python online documentation, (python.org, n.d.) and an example video on YouTube (YouTube, n.d.). After gaining an understanding created the loops to access use the password and salt directly from the files given, subsequently, I ran into an error regarding the information from the file being interpreted as byte code and not string. So, the passwords were all saved in a list before being used. I then processed to concatenate and hash the two variables and was met with difficulty in hashing the string correctly. This was quickly solved after referenced the text, (Computer Security Principles and Practice Fourth Edition) once again. Due to complications with accessing the file with the hash to be compared against, I inserted it directly into the code. My code was then able to recreate the process and produce the password and salt used.

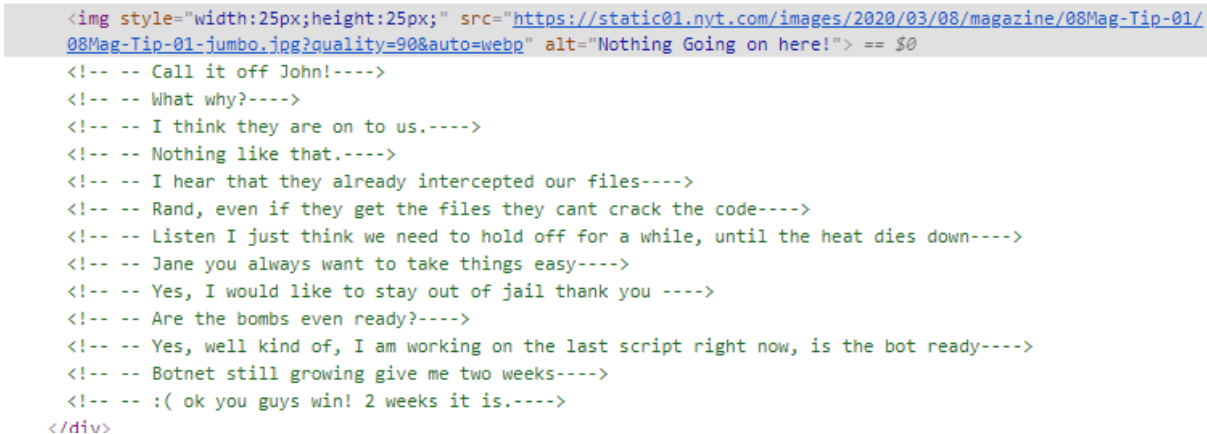
My code first accesses both the files of the popular passwords, the popular hashes and creates a passcode variable, saving the string "Passcode not found ... Mission Failed". The code goes on to creating two lists and using a for loop for each respective file, stripes each line using the python strip function and then stores the word on each line in a list. After the lists have been populated, the files are closed, and the lists are then traversed. The password list is traversed within a loop and then the salt list is traversed within a loop as well, inside the password list loop. This allows all the salts to be to loops with every password so that they can be concatenated together and hashed. The concatenation then takes place an using the sha256 function from the imported hashlib library the concatenated string is hashed with an ASCII encoding. The newly hashed is then converted to a string using hexdigest function and compared to the hash which was given in the mission if the hash matches the password and

salt used is stored in the passcode variable which was previously created, otherwise, the loop would continue and at the end would print the appropriate message of a pass or fail based on if the string was still the same as it was initialized to be at the start of the code.

MISSION 3

For mission 3, the URL to a website was given to be inspected to deduce any clues to the group which was planning the attack and exactly they were planning so countermeasures could be taken against them.

The information discovered was a secret chat between the terrorist who planned to make the speculated attack. I was able to deduce the number of persons involved, their first names or at least alias which they went by in the chat and the type of attack they were planning.



```
 == $0
<!-- -- Call it off John!---->
<!-- -- What why?---->
<!-- -- I think they are on to us.---->
<!-- -- Nothing like that.---->
<!-- -- I hear that they already intercepted our files---->
<!-- -- Rand, even if they get the files they cant crack the code---->
<!-- -- Listen I just think we need to hold off for a while, until the heat dies down---->
<!-- -- Jane you always want to take things easy---->
<!-- -- Yes, I would like to stay out of jail thank you ---->
<!-- -- Are the bombs even ready?---->
<!-- -- Yes, well kind of, I am working on the last script right now, is the bot ready---->
<!-- -- Botnet still growing give me two weeks---->
<!-- -- :( ok you guys win! 2 weeks it is.---->
</div>
```

The image above is a screenshot of the hidden chat which was discovered. There are three persons involved in the attack, John, Rand and Jane and the attack will take place in two weeks. The attack which they plan on committing is likely a botnet attack which can take the form of a denial-of-service attack. I speculate this since they are seen to be growing a botnet. Using a botnet, they could target an online server and overwhelm it, this could have many adverse effects based on their target or targets.

To discover the information displayed above my thought process was to inspect every element of the three webpages, this was the conclusion after I realized there was no information on the surface of the webpage. Though each verse was read, and a deep thought process was undertaken to see if the words on the site had any hidden meaning. When I saw

the bomb icon on the home page it looked suspicious, almost like a calling card so I inspected the icon and was able to find the chat. Reading the chat, I found the information I highlighted earlier.

My suggestion is to have the spies look into persons affiliated with setting up the website and who set up the domain name. This might lead to getting more information on the individuals involved since last names nor any other distinctive information about the suspects was retrieved. Further research and analysis are needed. Since we know the type of attack we can prepare against, if we can figure out the exact target then we need only have them reinforce their servers. The countermeasures we can implement are:

- Attack prevention and preemption – These help the target to endure attack attempts without denying service to legitimate clients by using policies for resource consumption and providing backup resources available on-demand, (Computer Security Principles and Practice Fourth Edition).
- Attack detection and filtering - To detect the attack as it begins and respond immediately by looking for suspicious patterns of behavior and filtering out packets likely to be part of the attack, (Computer Security Principles and Practice Fourth Edition).

REFERENCES

(n.d.). In L. B. William Stallings, *Computer Security Principles and Practice Fourth Edition* (pp. 97 - 98).

Stack Overflow(n.d.). Retrieved from python.org:

<https://docs.python.org/3.5/library/hashlib.html>

Stack Overflow(n.d.). Retrieved from YouTube:

https://www.youtube.com/watch?v=tk4Jl7L6hr0&t=223s&ab_channel=SachinShukla

How to get numbers after decimal point? (n.d.). Retrieved from Stack Overflow:

<https://stackoverflow.com/questions/3886402/how-to-get-numbers-after-decimal-point?page=1&tab=votes#tab-top>

RSA Algorithm - known n how to get p & q. (n.d.). Retrieved from Stack Overflow:

<https://stackoverflow.com/questions/16531958/rsa-algorithm-known-n-how-to-get-p-q>