**SIMPLE HASH CHAINS**

**By**

**Valery Aime Nundoya (B0099110)**

***School of Informatics***

***Department of Informatics and Engineering***

***Institute of Technology, Blanchardstown***

***Dublin 15.***

**Actual Words**  **Bachelor of Science Computing in**

**Information Technology**

**21/04/2019**



SECURE COMMUNICATION

Table of Contents

[Introduction 2](#_Toc6774376)

[Scenario 2](#_Toc6774377)

[Task 2](#_Toc6774378)

[Procedure 2](#_Toc6774379)

[Hash found 4](#_Toc6774380)

[Conclusion 5](#_Toc6774381)

[References 6](#_Toc6774382)

[Figure 1 Hello World 3](#_Toc6775041)

[Figure 2 Noob Hash 4](#_Toc6775042)

[Figure 3 Found challenge hash 5](#_Toc6775043)

# Introduction

The hash Chains lab is a lab to use python code to solve a simple hash chain. Hash chain is a successive application of a cryptographic hash function to a piece of data. According to (Hash Chain, n.d.) It’s a computerized security method to produce many one-time keys from a single key or passwords. Hash chain and blockchain are similar which are both utilize a cryptographic which has function for creating a link between two nodes.

Furthermore, more on this hash chain lab, it is going to help us understand how to calculate hashes using md5. We have to understand that, hash chain is a hash value that we hash again and again until the main challenge hash is found.

# Scenario

In this lab, a scenario is given as: an online service that uses hash chains has been made. ‘NooB’ is the user seed of 654e1c2ac6312d8c6441282f155c8ce9

# Task

The task then is to use the above given information to figure out how to authenticate as the user ‘ECSC’ with the given challenge hash 'c89aa2ffb9edcc6604005196b5f0e0e4'

Procedure

1 – The first part of this lab is to print “Hello World as you can see below which in this case I used on jdoodle for python (JDOODLE, n.d.)

Figure 1 Hello World

2- For the hash procedures, I imported the python hash library: (hashlid)

I then tried to figure how the nOOB seed 654e1c2ac6312d8c6441282f155c8ce9

Was gotten.

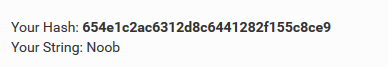


Figure 2 Noob Hash

It was generated from (md5hashgenerator, n.d.)

So next, using the example of the given hash, I then change the username ‘ECSC’ to lower case to be hashed to arrive at seed.

Challengehash as : 'c89aa2ffb9edcc6604005196b5f0e0e4'

the hash to use md5lib to hash the given ECSC String

So, for the md5 hashing algorithm to run on a string variable it needs to convert the string to bit using the utf-8 encoding

Hash will use the utf-8 to hash the seed using md5

At this stage a declared Variable is called seen to initialize to false, and md5 made readable to print the challenge has found.

While seen the challenge hash is not found, an if statement is passed here as shown in the codes below for the rest until the output:

**import** hashlib *#imported python hashing library  
  
# seed=('a2c83976c0adb482d280c6b10a042be3')*seed=**'ECSC'**.lower() *# using the example given, username 'ECSC' changed to lower case to be hashed to arrive at seed  
# seed.lower()*challengehash=**'c89aa2ffb9edcc6604005196b5f0e0e4'** *# the given challenge hash for 'ECSC'*hash = hashlib.md5() *#The md5 hash to hash the given ECSC string  
  
# for the md5 hashing algorithm to run on a string variable it needs to convert the string to byte using the utf-8 encording*hash.update(seed.encode(**'utf-8'**)) *# Now hash the seed using md5b*seen = **False** *# declare a variable called seed & initialize it to false*myseedhash=(hash.hexdigest()) *# md5 hash value made readable*print(myseedhash) *# print challenge hash found***while** (seen== **False**): *# while challenge hash is not found* **if**(myseedhash==challengehash): *#if challenge hash is found, print found challenge hash* print(**"found challengehash"**) *# print found challenge hash* seen = **True** *# make seed to become true* **break** *# break out of loop* **else**: *# challenge hash not found then continue to hash* myseedhash = hashlib.md5(myseedhash.encode(**'utf-8'**)).hexdigest() *# if challenge hash is* print(myseedhash) *# printing hashes as it loops through*

Hash found

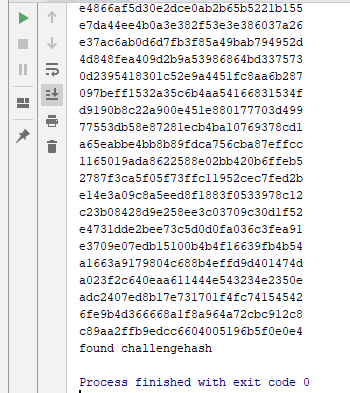


Figure 3 Found challenge hash

Conclusion

Going through this lab of hash chain was not an easy task, but it has given me knowledge on how hash chain works. I’ve learnt how to write a simple hello world with python, generate hashes and how to check them. I think python is a straight forward langusge.

# References

*Hash Chain*. (n.d.). Retrieved from https://en.wikipedia.org: https://en.wikipedia.org/wiki/Hash\_chain

*JDOODLE*. (n.d.). Retrieved from https://www.jdoodle.com: https://www.jdoodle.com/python3-programming-online

*md5hashgenerator*. (n.d.). Retrieved from https://www.md5hashgenerator.com: https://www.md5hashgenerator.com/