Secure Communications

Simple Hash Chains

Mark Cummins, ITB

OK for this lab you need to use python code to solve a simple hash chain or blockchain problem.

So getting started with python will likely be the biggest obstacle for most people. You can setup python on your own system, use kali/ubuntu, which already include python or use an online compiler and editor such as https://www.jdoodle.com/ or any of the other online free options.

Scenario:
You've registered for an online service that uses hash chains.

You've registered as user 'nOOB' and have been given the hash chain seed 654e1c2ac6312d8c6441282f155c8ce9

Use the given information to figure out how to authenticate as the user 'ECSC' for the given challenge hash c89aa2ffb9edcc6604005196b5f0e0e4 i.e. Find the hash that hashes to this - This hash will be your solution.

Procedure:

1. Get your python environment set up and run a hello world sample to get yourself started.

   Goto https://www.jdoodle.com/ and type the following code and execute it.
   print "Hello World!"

2. Next we want to see how to calculate hashes in python.
   You should be able to use the following bit of code to calculate the MD5 of the the string "Hello World!"

   ```
   import hashlib
   some_string = "Hello World!"

   hash = hashlib.md5()
   hash.update(some_string)
   print (hash.hexdigest())
   ```

3. Our next step is to understand what a hash chain is.

   So normally a hash chain is a hash value that we hash again, and again etc to produce a chain.

So
A = MD5('seed')
B = MD5(A)
C = MD5(B)
Etc.

Or MD5(MD5(MD5('seed'))

This produces a chain of hashes, and we keep going until the hash that we get is equal to the hash we are looking for (The challenge hash), so in our example we need to find 'c89aa2ffb9edcc6604005196b5f0e0e4'

4. The final piece of the jigsaw is to understand the 'seed' value. The seed value is the initial starting point of the hash chain. It's the original string that we hash and might normally be a user's password or similar.

   For our example this last step has a bit of a trick in it.. I suggest googling the seed given for the user nOOB and try to figure out how the seed value for this user is generated. Once you figure this out you need to calculate the seed for the user ECSC and then work out the hash chain until you reach the challenge hash.

5. The solution is the hash that actually maps to the challenge hash, so the second last hash in our chain as such.

Good luck.

You should submit your final hash and a detailed write up explaining your steps and logic and your final python code to solve the challenge.