Hash Collision

Problem: Can you log into our application and retrieve the flag?

Given: nc 127.0.0.1 30003; logic file

Hint: There might be a collision in our logic.

Steps)

1) Understanding the given logic file:

-We see that there is a username and password required.

-username = book

-password = complexPasswordWhichContainsManyCharactersWithRandomSuffixeghjrjg

-More importantly we see that the password is encrypted with a hash function called pbkdf2sync. This is a function for computing a shasum of the password given. If you research this function you will find out that it is vulnerable because it can also take a already hashed version of the same password that is computed by another shasum function and pass the password checking logic.

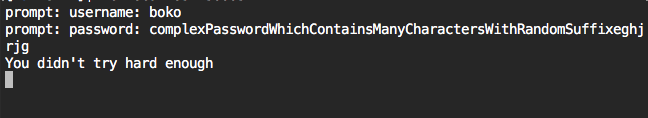
What this means if when this functions encrypted the password:

complexPasswordWhichContainsManyCharactersWithRandomSuffixeghjrjg

It computes a shasum of this password.

2) Connect to the server and try and log in.

- Let’s try the username and password that we can see in the logic first.



-This looks like we hit the if password if statement and the logic in there told us we did not try hard enough.

3) Generate hash and try to create hash collision

-From step 1 let’s recall that the function for the encryption is using a shasum. Let’s explore this in more detail.

There are also other ways to compute a shasum of a password. Such as the following:



The command:

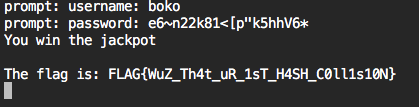
echo -n 'complexPasswordWhichContainsManyCharactersWithRandomSuffixeghjrjg' | shasum | xxd -r –p

Returns:

e6~n22k81<[p"k5hhV6\*

-We now have a hashed version of the password we want to use to log in. If we feed this hash string, we will create a hash collision inside the pbkdf2Sync function. To learn more about hash collision visit this [link](https://en.wikipedia.org/wiki/Collision_(computer_science)).

-Lets now try and log in and create the hash collision



-We now created a hash collision and were able to exploit the pbkdf2Sync function and log into the application and get the flag.