**CTF 01 Salt, Capability Report**

2018312567 조명하

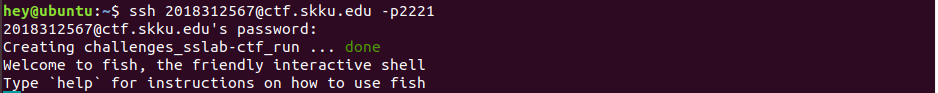
**0. Environment**

OS: Ubuntu 18.04. LST on VMware

Language: Python 3.8.5



**1. Connecting to ctf.skku.edu**



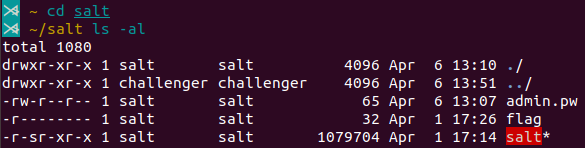
**2. Checking Directory Structure**

C:\Users\adminuser\AppData\Local\Microsoft\Windows\INetCache\Content.Word\디렉토리 처음.PNG

There are three directories.

**3. Solving *Salt* Problem**

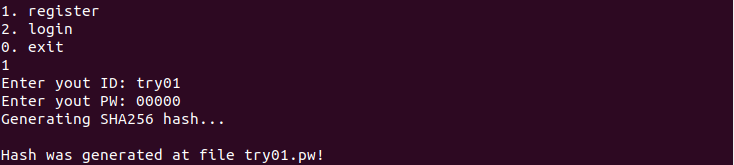
Using command ***cd***, go to ‘**salt/**’ and check its structure



I can only execute **salt**, read **admin.pw**. No right about **flag**.

**C:\Users\adminuser\AppData\Local\Microsoft\Windows\INetCache\Content.Word\실행.PNG**

Execute **salt.** After entering the command, it prints menu.

I choose **1. register** and set ID, PW as **try01, 00000**, then it makes **try01.pw**

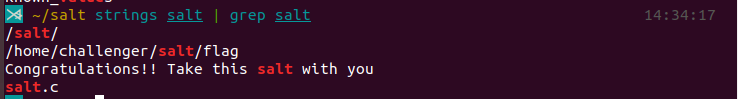
I guess the file **admin.pw** has a hash value of the combination of admin’s password and salt.

C:\Users\adminuser\AppData\Local\Microsoft\Windows\INetCache\Content.Word\비번해시.PNG

So, using command ***strings***, I read **admin.pw**. It seems that it contains the result of h(admin’s password, salt).



Back to the program salt, when I login as try01 and enter correct password, it prints **‘Helllo, try01.pw**!’ Maybe if I login as admin normally, it prints out the file flag or at least give me some useful information.



And I found a suspicious string ***“Congratulations!! Take this salt with you”***. Maybe I could get the salt value if I login as admin.

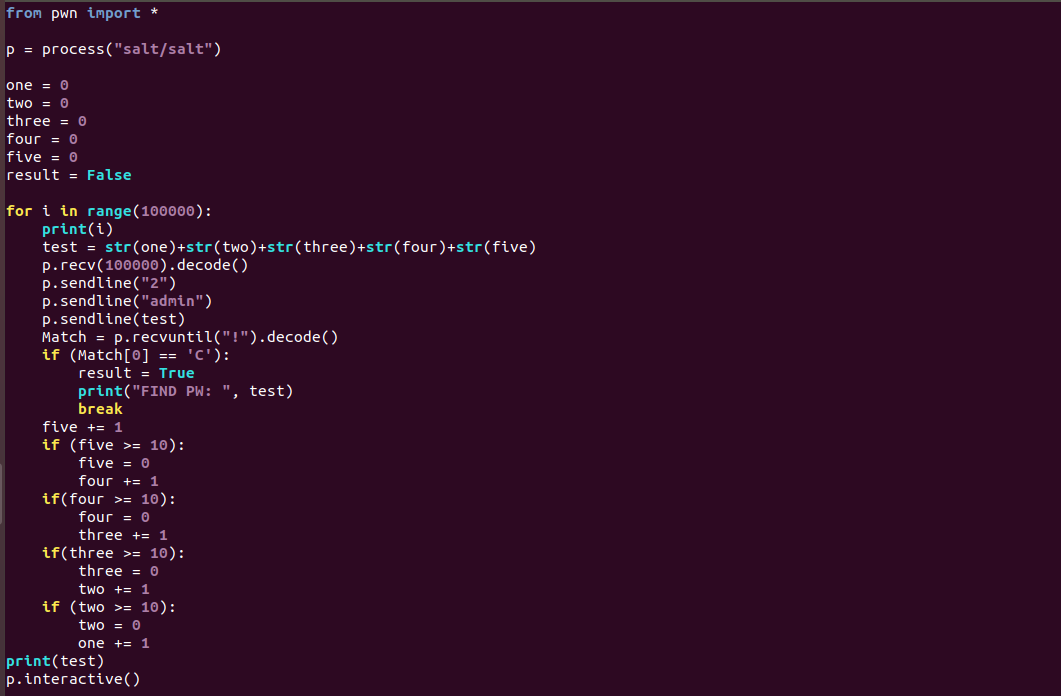
Since there are only two function, register and login in the program **salt**, I have only two options. **To register** admin’s new password, or **to login** as admin.



But when I try to register as admin, it prohibits my action. So, I have to find admin’s password, and then login normally as admin.

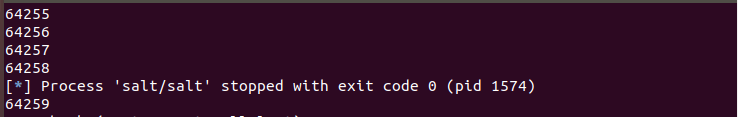
There are two hints. The password hashing algorithm uses a 4-byte salt. Admin's password is a 5-digit number.

This system saves hash value of password plus salt, and it is hard to decrypt hash value into plaintext in salt system. So, I have to try all possible passwords. To do that, I made a automation program **tryall.py** in the root directory(because I have no right to write on salt directory.)

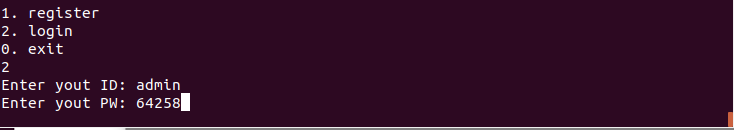


It automatically try to login as admin 100000 times, changing password from 00000 to 99999 until it finds correct password. Because I found a string starts with “Congratulations~”, I guess if it finds a correct password, the program would print that string, so it stops when the program prints a string starts with “C”. And print out the password tested on that time.

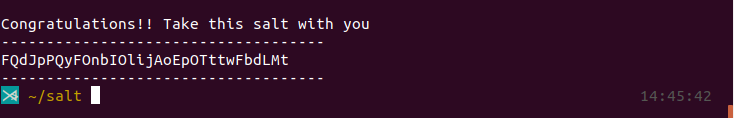




I execute it, and it stopped after trying **64258.**



So I login as **admin, 64258**

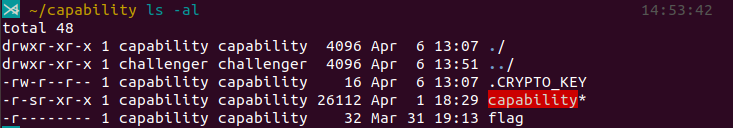


And it shows the salt value!

**4. Solving Capability Problem**

Go to directory capability and check its structure





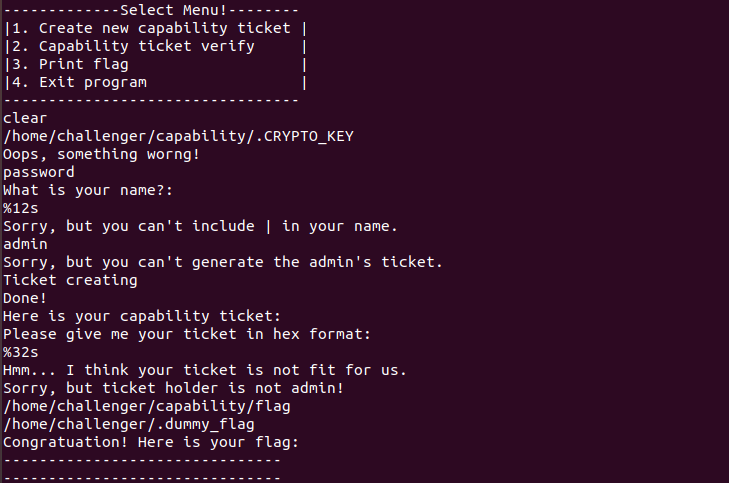
I can read and execute **capability,** can only read **.CRYPTO\_KEY,** have no right on **flag**

Read **.CRYPTO\_KEY** using ***strings***, but cannot understand



Read **capability** using ***strings,*** and found following contents





I guess that there are four functions. It takes input string as name, and it cannot include ‘|’ and cannot be ‘admin’. If I entering normal name, it seems to create capability ticket. But if the ticket holder is not **admin**, it prohibits certain actions. So, I need to figure out how admin’s ticket looks like, and verify my ticket is same as admin’s ticket to get flag.

I find that ‘admia’, ‘ndmia’, ‘ndmin’’s capability ticket share some parts. For example, ‘admia’ and ‘ndmia’’s capability ticket are exactly same except the first byte ‘00’ and ‘0f’. This means that ‘admia’ first character, ‘a’ turns into ‘00’.





And ‘ndmin’ would be exactly same except the first byte ‘0f’, and I can get ‘admin’s capability ticket by changing ‘0f’ into ‘00’. So ‘admin’s ticket is **00152b21194d040f2e2a29292e0a2a55**

So in menu 2, verifying this ticket and check my right is R



In menu 3, I enter admin’s ticket and get my flag.

