**BlackBox1 – Solution**

1. Create an empty .json file named **“config.json”** so the server will be able to be executed.

You can create one by entering: echo > config.json

1. Execute **“server.exe”** and afterwards **“secret.exe”** files on separated **cmd** interfaces.
2. Enter any appropriate input to the **secret.exe** execution, for example:

Enter your username: abcd

Enter your password: 1234

1. Afterwards a packet will be sent to the server, so to track it open **Wireshark** and sniff the packet:

From here we can understand that the client (in red) sends a json format messages to the server with the **“username”** flag that given and some other flags like: **“action”, “token”** and **“hash”**. In addition, to the **config.json** file this data was added:

{"users": [ {"username": "abcd", "token": "MTIzNA=="}]}

Which means that the config file saves users’ details.

1. After writing this script in python:

import base64

def decrypt(encrypted\_text):

    # The function gets a string.

    # The function returns the given string after decrypting it.

    return base64.b64decode(encrypted\_text).decode()

print(decrypt("MTIzNA=="))

I found that the value “**MTIzNA==**“ which was found on **config.json** and the packet itself under the flag “**token**” is equals “**1234**” which is the password I entered on secret.txt(look at section number 2), so from this it can be concluded that the flag **token** saves passwords encrypted.

1. Now, I have tried to enter a username that already exist from my previous try “**abcd**” but with wrong password. Therefore, the **secret.exe** returned me this message: **“Error: Invalid token for existing user”** so from this it can be concluded that **secret.exe** check for already exist users and doesn’t create new one with a different password.
2. Afterwards, I tried to login as “**admin**” with the password “**1234**”. Surprisingly, the secret.exe tells me that “**Error: Bad Password**” which means that the username “admin” probably exist but with another password.
3. Then, I created an artificial user of the admin with the password “**admin123**” in **config.json** buy adding this to the “**users**” square brackets:

{

"username": "admin",

    "token": "YWRtaW4xMjM="

}

and try login to this admin user. The secret.exe sent me again the error message of bad password so it doesn’t seem to be helpful.

1. I wrote a script in python to edit a packet of a “**get-actions**” action with **admin** username and a random encrypted password as a token and send it to the server. This is the script:

import requests

URL = "http://localhost:5555"

# Get-actions request

message\_body = {"action": "get-actions", "type": "admin", "token": "MTIzNA=="}

respond = requests.post(url=URL, json=message\_body)

respond\_data = respond.json()

if respond.status\_code == 200:

    print(respond\_data)

else:

    print(f"Error{respond.status\_code}")

The server returned to me this message “**get-users”**. It seems to be an action mode so I changed the script’s “**message\_body”** parameter with this action mode:

message\_body = {"action": "get-users", "type": "admin", "token": "MTIzNA=="}

After running the script again and sniff it the returned message was: **“ {"users":[{"date registered":"XX/XX/2019","token":"Y3liZXI=","username":"admin"}]} ”.**

1. It can be concluded from this message that the **admin**’s user password is encrypted to **“Y3liZXI=”** and after decrypting it the password found to be “**cyber**”. In addition, the action **get-users** probably return information about specific users.
2. Next, I tried to login with the username and the password of the **admin** user via the **secret.exe** but it returned me a new message “**Error: Good password, wrong hash**”, so now the problem is the hash.