First I dowloaded the ‘encrypted’ file and ZOR.py into my computer.

* ZOR.py <http://pastebin.com/R6KDDeCW>
* encrypted file <http://pastebin.com/DDryi10H>

\*I placed these 2 files in a folder called “zor”

Then I downloaded WinSCP(<http://winscp.net/eng/download.php>) to login to the shell server via SSH to upload the folder to the shell account PicoCTF provided for you when you visit the ‘Shell’ page.

I just placed the folder in the root directory of my account.

Once the folder is copied over to the shell server, go to the “Shell” page and login and stuff.

Type: cd zor (or whatever you named the folder)

This will change your current directory over to ~/zor. So now you will be inside the “zor” folder.

To run ZOR.py, type:

python ZOR.py arg1 arg2 arg3 arg4 \*don’t actually type these args\*

What this does is that it runs the shell server’s python program to execute ZOR.py with those 4 arguements.

If you don’t type it correctly, it’ll give you the correct usage syntax:

Usage: ZOR.py [encrypt/decrypt] [in\_file] [out\_file] [password]

Basically, you have the option to encrypt/decrypt the input file and it’ll produce an output file and also the password.

So in our case, we would want to decrypt the ‘encrypted’ file.

So we would type:

python ZOR.py decrypt encrypted decrypted\_file(Whatever name you want to give it) [theactualpassword]

I’ve decided to use a python script to bruteforce the encrypted file and came up with this:

[0X86]

tHISMESSAGEISFORdAEDALUScORPORATIONONLYURBLUEPRINTSFORTHE␌YBORGAREPROTECTEDWITHAPASSWORD├HATPASSWORDISFA

DBCEDEDAEFE

The key it used is 0x86 which is a hex value. Converted to binary, that will be 10000110. But when I tried decrypting through ZOR.py it still outputted a encrypted file…

\*\*\*So I noticed that the key was a little bit off. I added 20 to the current key since I noticed that the letters were switched (Lower case switched with Upper case) and the key was now 0xA6, which was the correct key which decrypted to:

[0XA6]

This message is for Daedalus Corporation only. Our blueprints for the Cyborg are protected with a passwor

d. That password is fa64d1b7c32561ede2d93a2e7fe801

[/0XA6]

**Make your own encrypted file for testing!**

1. Copy-Paste ‘encrypted’ file and change its name to whatever you want. In this case, the file will be named “test”.
2. Edit the file and erase the data inside and write a message like “Hello World” and save&close the file.
3. Copy that file to the “zor” folder in the shell server if you haven’t already.
4. In the shell command prompt (Assuming you’re already in the correct directory), type:

python ZOR.py encrypt test testEncrypted mypassword(whatever pass)

1. Once that command executes, you should see a “testEncrypted” file in the shell server inside the “zor” folder. (try refreshing if it’s not there)
2. If you open the file up, you’ll see that it’s similar to the text that ‘encrypted’ has.
3. To decrypt the file, go back into the shell command prompt and type:

python ZOR.py decrypt test testDecrypted mypassword(the pass you used to encrypt it)

1. Now if you look at your “zor” folder in the shell server, you should see a “testDecrypted” file inside.

If you edit it, you’ll see that it’s the same text that you originally typed. In this case, it’ll be “Hello World”.