

Forum: [Hack Forums](#)

Thread: [Md5 Cracker Help](#)

XxClassifiedxX - May 30, 2017, 1:41 pm

I'm still very new to python and I saw a video on making a md5 cracker in python so I tried to make my own. The video's however was in 2.7 while I am using version 3. I'm almost done with it but I keep getting this error. " TypeError: Unicode-objects must be encoded before hashing". Any help is appreciated on how to fix this, the code is below. Thanks.

PHP Code

```
<?php
import hashlib
import os

os.system('cls')

md5_hash = input("Md5 Hash: ")
pw_list = input("\nPassword List: ")

try:
    file = open(pw_list, 'r', encoding='utf-8')

except:
    print("\nFile not found")

for password in pw_list:

    md5 = hashlib.md5(password).hexdigest()

    print("Trying password {0}".format(password.strip()))

    if md5_hash == md5:
        print("[*] Key found: {0}".format(password))

    else:
        print("Password not found")
```

SIM-1939868 - May 30, 2017, 2:27 pm

You're looping over pw_list, which is a string, not over the encoded lines in the file.

Change it to 'for password in file'

XxClassifiedxX - May 30, 2017, 2:34 pm

How did i not see that. thank you

Additionally, my if statement is not working for some reason, any ideas why.

hunter24957 - May 30, 2017, 3:28 pm

Needs to be encoded as UTF-8

Code:

PHP Code

<?php

import hashlib

```
if __name__ == '__main__':
    # prompts user for hash
    original_hash = input('Please enter your hash: ')
    # prompts user for password list
    passwords = open(input('Please enter your password list:')).read().splitlines()
    # iterates through password list
    for password in passwords:
        # hashes password then checks it against original hash
        if hashlib.md5(password.encode('utf-8')).hexdigest() == original_hash:
            # notifies user of hash collision
            print('Found password:', password)
```

Output:

Code Please enter your hash: 49f68a5c8493ec2c0bf489821c21fc3b

Please enter your password list: passwords.txt

Found password: hi

XxClassifiedxX - May 30, 2017, 5:38 pm

thank you so much dude!!!

SIM-1939868 - May 31, 2017, 11:17 am

(May 30, 2017 03:28 PM)hunter24957 Wrote: [CITE](#) Needs to be encoded as UTF-8

Code:

PHP Code

<?php

import hashlib

```
if __name__ == '__main__':
    # prompts user for hash
    original_hash = input('Please enter your hash: ')
    # prompts user for password list
    passwords = open(input('Please enter your password list:')).read().splitlines()
    # iterates through password list
    for password in passwords:
        # hashes password then checks it against original hash
        if hashlib.md5(password.encode('utf-8')).hexdigest() == original_hash:
            # notifies user of hash collision
            print('Found password:', password)
```

Output:

Code Please enter your hash: 49f68a5c8493ec2c0bf489821c21fc3b

Please enter your password list: passwords.txt

Found password: hi

He did encode it as utf-8

hunter24957 - May 31, 2017, 2:47 pm

(May 31, 2017 11:17 AM)Thommen Wrote: [\[CITE\]](#) He did encode it as utf-8

No, he simply explicitly told the system the encoding of the file he was reading into memory. As shown below, this prevents most encoding errors when reading a file:

PHP Code

```
<?php
open('passwords.txt', encoding='utf-8')
```

This (below) converts the string into a byte string which can then be hashed, utf-8 is just a standard type of encoding.

PHP Code

```
<?php
.encode('utf-8')
```

This needs to be explicitly done in Python 3.x when hashing a string. I believe, like the socket module, it's done implicitly in Python 2.x.

Works like a charm! ?

P.S. he literally displayed that the error was due to an encoding issue...

(May 30, 2017 01:41 PM)XxClassifiedxX Wrote: [\[CITE\]](#) "TypeError: Unicode-objects must be encoded before hashing"

SIM-1939868 - May 31, 2017, 3:36 pm

(May 31, 2017 02:47 PM)hunter24957 Wrote: [\[CITE\]](#) P.S. he literally displayed that the error was due to an encoding issue...Look at his code.

He opened the file in binary mode, which is fine. He accidentally looped over a string instead of the file itself.

hunter24957 - May 31, 2017, 3:51 pm

(May 31, 2017 03:36 PM)Thommen Wrote: [\[CITE\]](#) Look at his code.

He opened the file in binary mode, which is fine. He accidentally looped over a string instead of the file itself.

Yes, I could see that that was also an issue, hence the reason I rewrote his code to ensure it was working due to a few other simple mistakes he made.

What I'm trying to get at is that the attempt to iterate over a string was not the only issue.

Although I'm glad you pointed it out; as I said before hashlib expects an encoded byte string to hash.

However, he did not read the file in binary mode:

PHP Code

```
<?php
open(pw_list, 'r', encoding='utf-8')
```

The 'r' is just a simple read flag, which is the right way to do it although Python will implicitly do this for you, if he had used 'rb' that's a flag that will tell the system to interpret the file in byte format, which he would not need to do because he simply needs the contents of the file and not the byte code or as you put it, binary format.

As I said before the "encoding='utf-8'" just explicitly defines the encoding in which to read the file.

I'm enjoying this debate but just run my code, it works fine. ?

SIM-1939868 - May 31, 2017, 4:10 pm

(May 31, 2017 03:51 PM)hunter24957 Wrote: [\[CITE\]](#) Yes, I could see that that was also an issue, hence the reason I rewrote his code to ensure it was working due to a few other simple mistakes he made.

What I'm trying to get at is that the attempt to iterate over a string was not the only issue. Although I'm glad you pointed it out; as I said before hashlib expects an encoded byte string to hash.

However, he did not read the file in binary mode:

PHP Code

```
<?php
```

```
open(pw_list, 'r', encoding='utf-8')
```

The 'r' is just a simple read flag, which is the right way to do it although Python will implicitly do this for you, if he had used 'rb' that's a flag that will tell the system to interpret the file in byte format, which he would not need to do because he simply needs the contents of the file and not the byte code or as you put it, binary format.

As I said before the "encoding='utf-8'" just explicitly defines the encoding in which to read the file.

I'm enjoying this debate but just run my code, it works fine. ?Sure he didnt open it in binary mode but opened it with UTF-8 encoding. Its just that his code worked fine as well, only he made a mistake on the iteration.

The error is not about the encoding, its about the mistake that he made in his for loop.

Goodies - May 31, 2017, 4:11 pm

Why on earth would you be using Python for this?

Good luck finding someone who's a bigger Python advocate than I am. I like hammers, too. It's a good tool, but you wouldn't use a hammer to put in a bolt.

Use the right tool for the right job. Python's performance is horrible as a CPU-bound language. Use something lower level.