
ATTACK
DEFENSE
by PentesterAcademy

Name	Eval Function
URL	https://www.attackdefense.com/challengedetails?cid=589
Type	Secure Coding : Python

Important Note: This document illustrates all the important steps required to complete this lab. This is by no means a comprehensive step-by-step solution for this exercise. This is only provided as a reference to various commands needed to complete this exercise and for your further research on this topic.

A vulnerable binary "script" is given in student home directory. The source code file (script.py) of this binary is also given in the same directory.

Objective: Use the "script" binary to read flag file kept in the root directory.

Solution

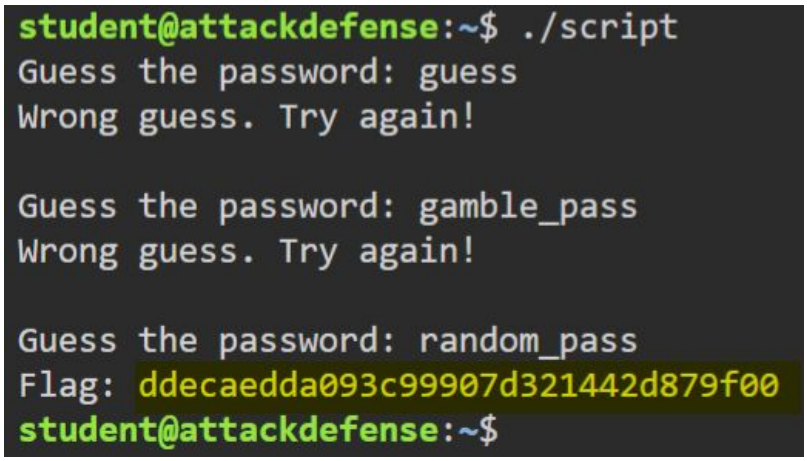
Observe that setuid bit is set for the binary.

```
student@attackdefense:~$ ls -l
total 3832
-rwsr-xr-x 1 root root 3916872 Jan  6 18:47 script
-rw-r--r-- 1 root root     527 Jan  6 18:46 script.py
student@attackdefense:~$
```

From the python code file (script.py), it is clear that eval() function is being applied to the user inputs. Eval function is known to evaluate the passed value rather than just handling it like a string. This property/vulnerability can be exploited here.

```
student@attackdefense:~$ cat script.py
# Game of luck

import random
import subprocess
import os

os.setuid(0)

random_pass=""
for i in range(1, 10):
    random_pass=random_pass+str(chr(96+random.randint(1,26)))

while 1:
    gamble_pass = raw_input("Guess the password: ")
    try:
        gamble_pass = eval(gamble_pass)
        if random_pass == gamble_pass:
            subprocess.call("cat /root/flag", shell=True)
            break
        else:
            print("Wrong guess. Try again! \n")
    except NameError:
        print("Wrong guess. Try again! \n")
student@attackdefense:~$
```

Enter the same variable name as being used by the code during the comparison. This will lead to successful comparison and the flag hidden in the root directory will be revealed.



Flag: ddecaedda093c99907d321442d879f00