Group A

Guilherme Pereira

picoCTF 2022 - Forbidden Paths

```
We know that the website files live in /usr/share/nginx/html/ and the flag is at /f
lag.txt but the website is filtering absolute file paths.
Can you get past the filter to read the flag?

1) Using gospider
    (https://github.com/jaeles-project/gospider)

    [0000]  INFO Start crawling: http://saturn.picoctf.net:49700/
    [url] - [code-200] - http://saturn.picoctf.net:49700/
    [href] - http://saturn.picoctf.net:49700/style.css
    [form] - http://saturn.picoctf.net:49700/
    [0000]  INFO Done.


2) So the form is our only point of attack.
   Since the assig description states that `website is filtering absolute file pat
hs`.
   Thus we needed to use the relative address (./)by typing:
   ../../../../flag.txt the filter is bypassed.

F: picoCTF{7h3_p47h_70_5ucc355_6db46514}
```

picoCTF 2022 - Roboto Sans

```
The flag is somewhere on this web application not necessarily on the website. Find
it.

1) gospider -v -s http://saturn.picoctf.net:65352/

    Found robots.txt: http://saturn.picoctf.net:65352/robots.txt

        User-agent *
        Disallow: /cgi-bin/
        Think you have seen your flag or want to keep looking.

        ZmxhZzEudHh0;anMvbXlmaW
        anMvbXlmaWxlLnR4dA==
        svssshjweuiwl;oiho.bsvdaslejg
        Disallow: /wp-admin/

    Text above the second "Disallow" looked like base64.
    So I used cyberchef to decode.

2)
    ZmxhZzEudHh0;anMvbXlmaW
    anMvbXlmaWxlLnR4dA==
    svssshjweuiwl;oiho.bsvdaslejg
    From base64:

    flag1.txtjs/myfif§2ö×.f.ÆRçG.@./²Ë!..®. h...²÷Z²W£


3)
    Using ';' as a delimiter for each string

    ZmxhZzEudHh0 --> flag1.txt

    anMvbXlmaW   --> js/myfi

    anMvbXlmaWxlLnR4dA==                     --> js/myfile.txt
        (same as above but with extra info)

    The rest wasn't decodable.

Since the third substr holds a path, by accessing http://saturn.picoctf.net:65352/
js/myfile.txt
the flag was uncovered.

F: picoCTF{Who_D03sN7_L1k5_90B0T5_718c9043}
```

picoCTF 2022 - Fresh Java

```
1) Using ghidra, mainly the decompile section

    Key can be found in the main function (vertically):

        picoCTF{700l1ng_r3qu1r3d_738cac89}
```

BONUS

```
1) file findMeV4.pdf

    findMeV4.pdf: Zip archive data, at least v2.0 to extract, compression method=de
flate

2) mv findMeV4.pdf findMeV4.zip

3) unzip findMeV4.zip
    Archive:  findMeV4.zip
        inflating: findMeV4.py

4) First run: python3 findMeV4.py. Script expects a password, our goal is to find t
he correct password.

5) cat findMeV4.py
"""
from cryptography.fernet import Fernet
muHa = "bAF0rTG5AY2PMO86AtOzvcqDxibbRyfIAJ38rzvOxZVsSp6!WBi7QvOSHSd3Lz4BtpSePA6jn9K
dcqfrodr4YjP9fdgkGOBjD!N6CLOvEkyvwTx8DAlFSyFHdbOMfeWZpMEyxRbYy1hgat2EOqZLoGgemWhBCy
nRj2tpijuTacdMXZYNctDx8PbSI8FHS5fNdQPVvVWiYnQouUxEWHNYVpNkmyyBUTB4PzvWhLulUVTy4gulG
hzgRLVydgPD3zbu3GlwwWjD2v7itpPBBVz3skHqmmx6MAOskoc7oP90fRq1BW2cA01f1jAST3UQdryVT8x3
iBzYfRToFnpKMRVfmJtQRNlkHv0av1onhjSpLUChop5rwPxXTUWEUHLrrIwIxUO1mAr2TpKAjr1rEycue4f
```

```
1uL7QQ2MPBpzMxFDe6qjU0ytEkWjRFZahlO20CJDsMl4VcVjqo4gyOCaONTQwZuRqClv9yDZQIodTqZj0NZ
XK5fF5tGn8RP$AU45u#TR?as3"
bb = b'EAeeFGJDUZEeaishu4qaxswNUmKcmTDn1HjrKYN8U7Y='
b = b'gAAAAABjDLCIjp7gZPdLEynEO0XaCNusDgM5jjUkKQ7hoXLGIQq4t5UoQw9WmsgAup3XlBkqkwWh
gsQM7qYuj3piy0xpRfQnh4F6Pa6XOkXimrH9gjtQnphSnri_3q8Y6oshPIcz9rbgl1CVjCpEGEi3DPRS5Oe
-CstC_CNQG-pDFmU255PeksQdMWNI2v9e3Ugy22hwB8ndgW6H3jNTZ89ru5DH-MUKLFMwcYEKQtQhC3XlA
wJaOXwU_WZ5X9TMbV1ODBChm-taN8T2V0pU8hx9Yji5brVHfgUfXGcAdGZKGJItMpixYZ0ISvnBvHB04bWN
wlf1eBMmG_3Jpl33SerrCCIe4OfOPpB2ktRbhDceppl6mlnwbfo75ZZAMNbqP3RRK2-eWjFEKY47GTf2pN6
KMKW6V2s3nK9lZpP4FLPVJ16OsuPLd8Q6NrcKwt2o5sYPA54M3fb22-4U6hQhkmVgJ_m24SHRCw=='
exec(Fernet(bb).decrypt(b))
"""

From the documentation (https://cryptography.io/en/latest/fernet/):
    Fernet is used to encrpyt and decrypt messages, where in this case, the key is
bb and b is our "message".
    Since an exec statement is called, the decryption of the variable b must be cod
e (in this case python).

6) reversing findMeV4.py:

"""
f = Fernet(bb)
print(f.decrypt(b))
"""

OUT:
b"\ndyrxjtyjxfgn = input(muHa[501]+muHa[503]+muHa[55]+muHa[57]+muHa[511]+' ')\n\ni
f dyrxjtyjxfgn == muHa[100]+muHa[200]+muHa[300]+muHa[400]+muHa[500]+muHa[250]+muHa[
125]+muHa[75]+muHa[50]+muHa[25]:\n  print(muHa[401]+muHa[403]+muHa[45]+muHa[47])\ne
lse:\n  print(muHa[201]+muHa[203]+muHa[65]+muHa[67]+muHa[97])\n\n"

7) Analyze code from the print above :=
    The equality in the if statement holds the only valid password:
        print(muHa[100]+muHa[200]+muHa[300]+muHa[400]+muHa[500]+muHa[250]+muHa[125
]+muHa[75]+muHa[50]+muHa[25]) --> CYBERPedii
```

In [ ]:

```python
# BONUS
from cryptography.fernet import Fernet

muHa = "bAF0rTG5AY2PMO86AtOzvcqDxibbRyfIAJ38rzvOxZVsSp6!WBi7QvOSHSd3Lz4BtpSePA6jn9Kdcqfro
dr4YjP9fdgkGOBjD!N6CLOvEkyvwTx8DAlFSyFHdbOMfeWZpMEyxRbYy1hgat2EOqZLoGgemWhBCynRj2tpijuTac
dMXZYNctDx8PbSI8FHS5fNdQPVvVWiYnQouUxEWHNYVpNkmyyBUTB4PzvWhLulUVTy4gulGhzgRLVydgPD3zbu3Gl
wwWjD2v7itpPBBVz3skHqmmx6MAOskoc7oP90fRq1BW2cA01f1jAST3UQdryVT8x3iBzYfRToFnpKMRVfmJtQRNlk
Hv0av1onhjSpLUChop5rwPxXTUWEUHLrrIwIxUO1mAr2TpKAjr1rEycue4f1uL7QQ2MPBpzMxFDe6qjU0ytEkWjRF
ZahlO20CJDsMl4VcVjqo4gyOCaONTQwZuRqClv9yDZQIodTqZj0NZXK5fF5tGn8RP$AU45u#TR?as3"
bb = b'EAeeFGJDUZEeaishu4qaxswNUmKcmTDn1HjrKYN8U7Y='
b = b'gAAAAABjDLCIjp7gZPdLEynEO0XaCNusDgM5jjUkKQ7hoXLGIQq4t5UoQw9WmsgAup3XlBkqkwWhgsQM7qY
uj3piy0xpRfQnh4F6Pa6XOkXimrH9gjtQnphSnri_3q8Y6oshPIcz9rbgl1CVjCpEGEi3DPRS5Oe-CstC_CNQG-pD
FmU255PeksQdMWNI2v9e3Ugy22hwB8ndgW6H3jNTZ89ru5DH-MUKLFMwcYEKQtQhC3XlAwJaOXwU_WZ5X9TMbV1OD
BChm-taN8T2V0pU8hx9Yji5brVHfgUfXGcAdGZKGJItMpixYZ0ISvnBvHB04bWNwlf1eBMmG_3Jpl33SerrCCIe4O
fOPpB2ktRbhDceppl6mlnwbfo75ZZAMNbqP3RRK2-eWjFEKY47GTf2pN6KMKW6V2s3nK9lZpP4FLPVJ16OsuPLd8Q
6NrcKwt2o5sYPA54M3fb22-4U6hQhkmVgJ_m24SHRCw=='

f = Fernet(bb)
print(f.decrypt(b))

# dyrxjtyjxfgn = input(muHa[501]+muHa[503]+muHa[55]+muHa[57]+muHa[511]+' ')

print(muHa[100]+muHa[200]+muHa[300]+muHa[400]+muHa[500]+muHa[250]+muHa[125]+muHa[75]+muHa
[50]+muHa[25])
print(muHa[401]+muHa[403]+muHa[45]+muHa[47])
print(muHa[201]+muHa[203]+muHa[65]+muHa[67]+muHa[97])
```

```
# if dyrxjtyjxfgn == muHa[100]+muHa[200]+muHa[300]+muHa[400]+muHa[500]+muHa[250]+muHa[125
]+muHa[75]+muHa[50]+muHa[25]:
#     print(muHa[401]+muHa[403]+muHa[45]+muHa[47])
# else:
#     print(muHa[201]+muHa[203]+muHa[65]+muHa[67]+muHa[97])
```