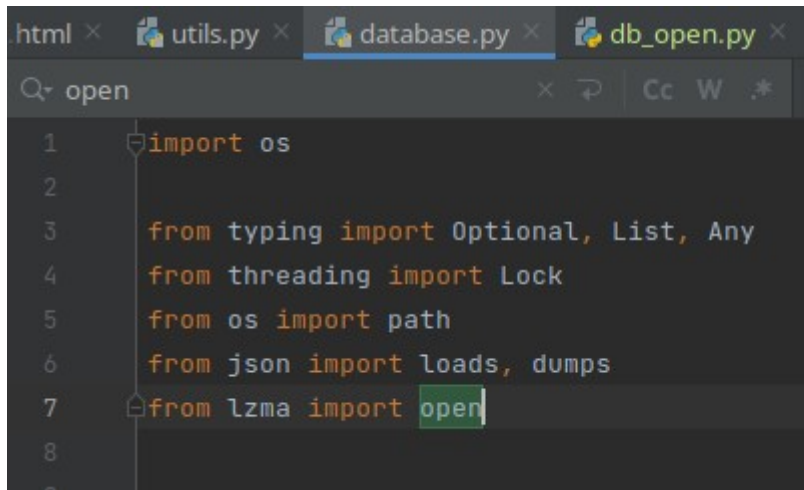
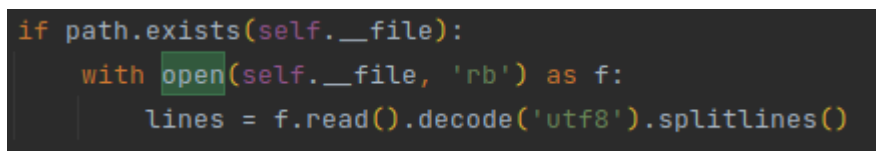


Extrakcia dát z databázy

- Databáza je zašifrovaná vo formáte .xz
- Kód database.py sa snaží zmiast' útočníka tým, že používa knižnicu lzma (<https://docs.python.org/3/library/lzma.html>) so svojou vlastnou funkciou open namiesto zabudovanej funkcie open.



```
html x utils.py x database.py x db_open.py x
Q open
1 import os
2
3 from typing import Optional, List, Any
4 from threading import Lock
5 from os import path
6 from json import loads, dumps
7 from lzma import open
8
```



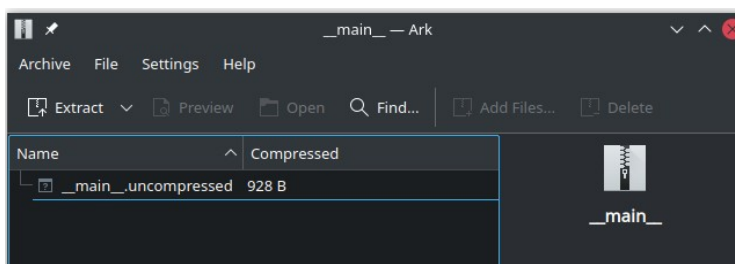
```
if path.exists(self.__file):
    with open(self.__file, 'rb') as f:
        lines = f.read().decode('utf8').splitlines()
```

- Databázu môžeme prečítať tým, že ju otvoríme pomocou knižnice lzma.



```
1 from lzma import open
2
3 filepath = "db/__main__"
4
5 with open(filepath, "rb") as file:
6     print(file.read())
7
```

- Pri kliknutí na databázu môže operačný systém detekovať, že ide o komprimovaný súbor a odkomprimuje ho.



- Môže dôjsť k úniku citlivých údajov užívateľov ako sú heslá ale aj komentáre a hlasovanie.

Nahrание súboru

- Directory traversal
- Vytvorením používateľa s menom ../db/ a následným nahraním súboru sa použije toto meno v názve súboru a pri ukladaní vyjde von z priečinka data do priečinka db.

19:49:34

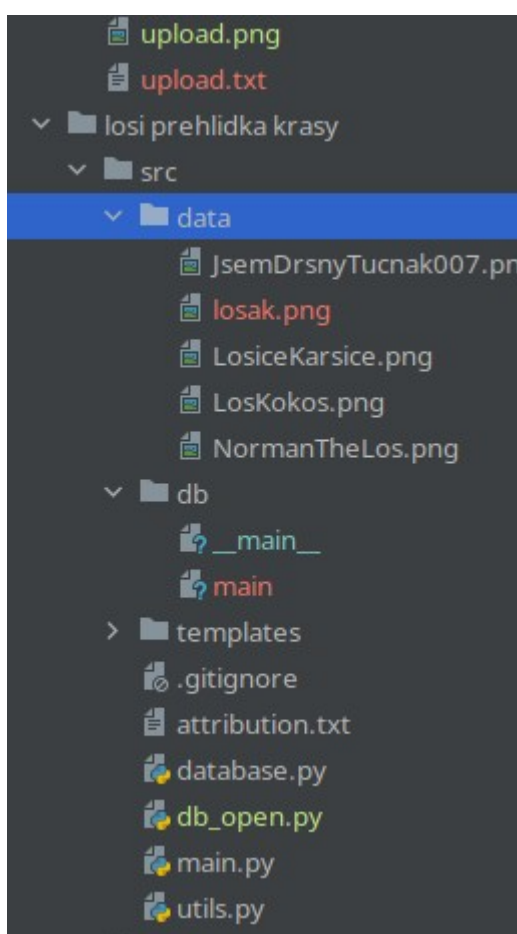
Los Miss

You are logged in as ../db/ [Logout](#)

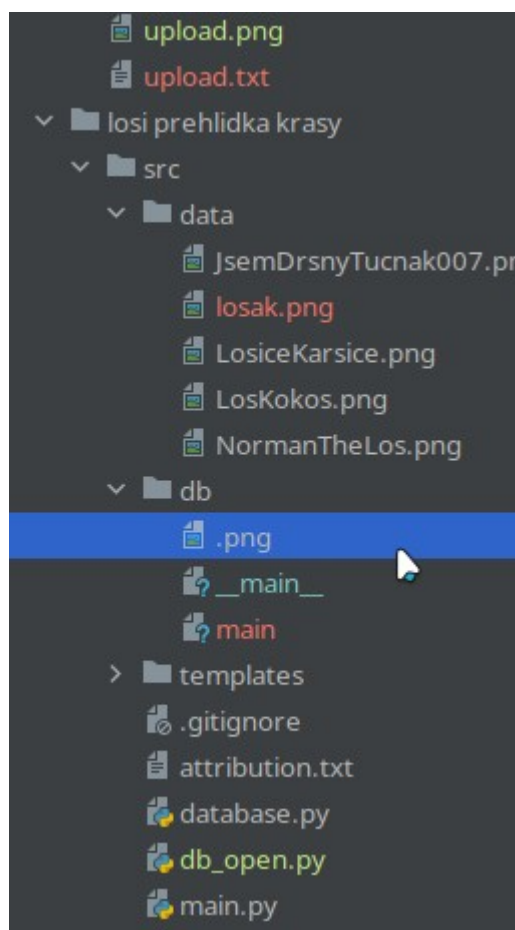
Add your own submission

upload.png

(c) 1921 - 2022 Miss Los: Scholarship pageant - the largest tuition provider for animals



-->



- Zapríčiňuje to 133. riadok v main.py . Pri ukladaní súboru vytvára cestu data/<uzivatel>.png z ktorej môžeme uniknúť zmenením mena užívateľa.

```
132     save_submission(database, get_username(database, request))
133     request.files['file'].save(DATA_DIR + '/' + get_username(database, request) + '.png')
134     return redirect(url_for('homepage'))
```

- Útočníkovi to umožňuje vloženie škodlivého kódu na nechránené miesto na serveri. Tento kód môže byť automaticky spustený a umožní vstup útočníka ku serveru.

Ovplivnenie hlasovania

- Form resubmission vulnerability (<https://stackoverflow.com/questions/3923904/preventing-form-resubmission>)
- Po prihlásení nám stačí poslať POST request na /submission/<meno> s dátami “type“ : “vote“ (ukážka je vo voting_vuln.txt).
- Stránka bráni používateľovi pridávanie hesiel len tým, že po hlasovaní odstráni tlačidlo vote. Http requesty ale stále prijíma.
- Keď po zahlasovaní refreshneme stránku tak prehliadač pošle nový POST request s rovnakým formulárom čo server akceptuje a pridá ďalší hlas.


14:0:8

Submission of LosKokos

Confirm Form Resubmission

The page that you're looking for used information that you entered. Returning to that page might cause any action you took to be repeated. Do you want to continue?

Cancel Continue



1 votes

Enter your comment here...


Send comment

Sympaták :)

14:1:21

Los Miss

Submission of LosKokos



2 votes

Enter your comment here...

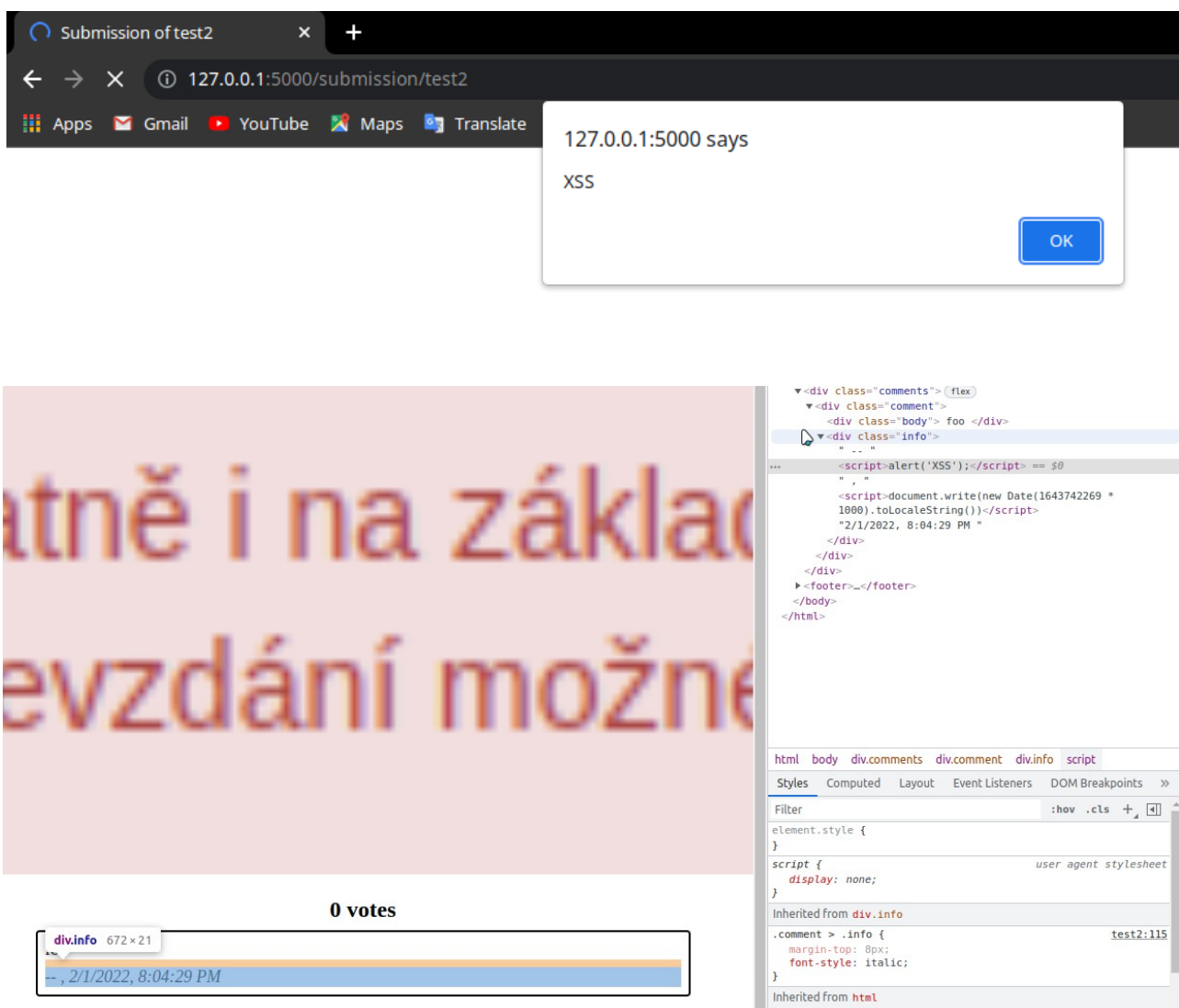
Send comment

Sympaták :)

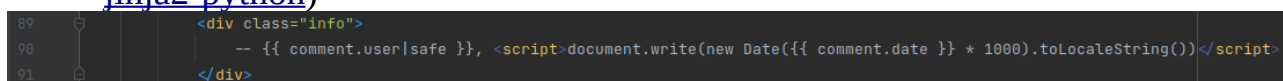
- Zraniteľnosť umožňuje ovplivnenie hlasovania ale aj spam komentárov keďže táto zraniteľnosť pridáva aj komentáre.

Spustenie vlastného JS kódu

- Perzistentné XSS
- Vytvorením používateľa s javascript kódom v mene, napríklad `<script>alert('XSS');</script>` a následné napísanie komentára uloží tento JS kód do databázy a takto aj pre všetkých užívateľov ktorí prídu na stránku.
- Text komentára je chránený proti XSS ale meno, ktoré sa zobrazuje pod komentárom nie je.



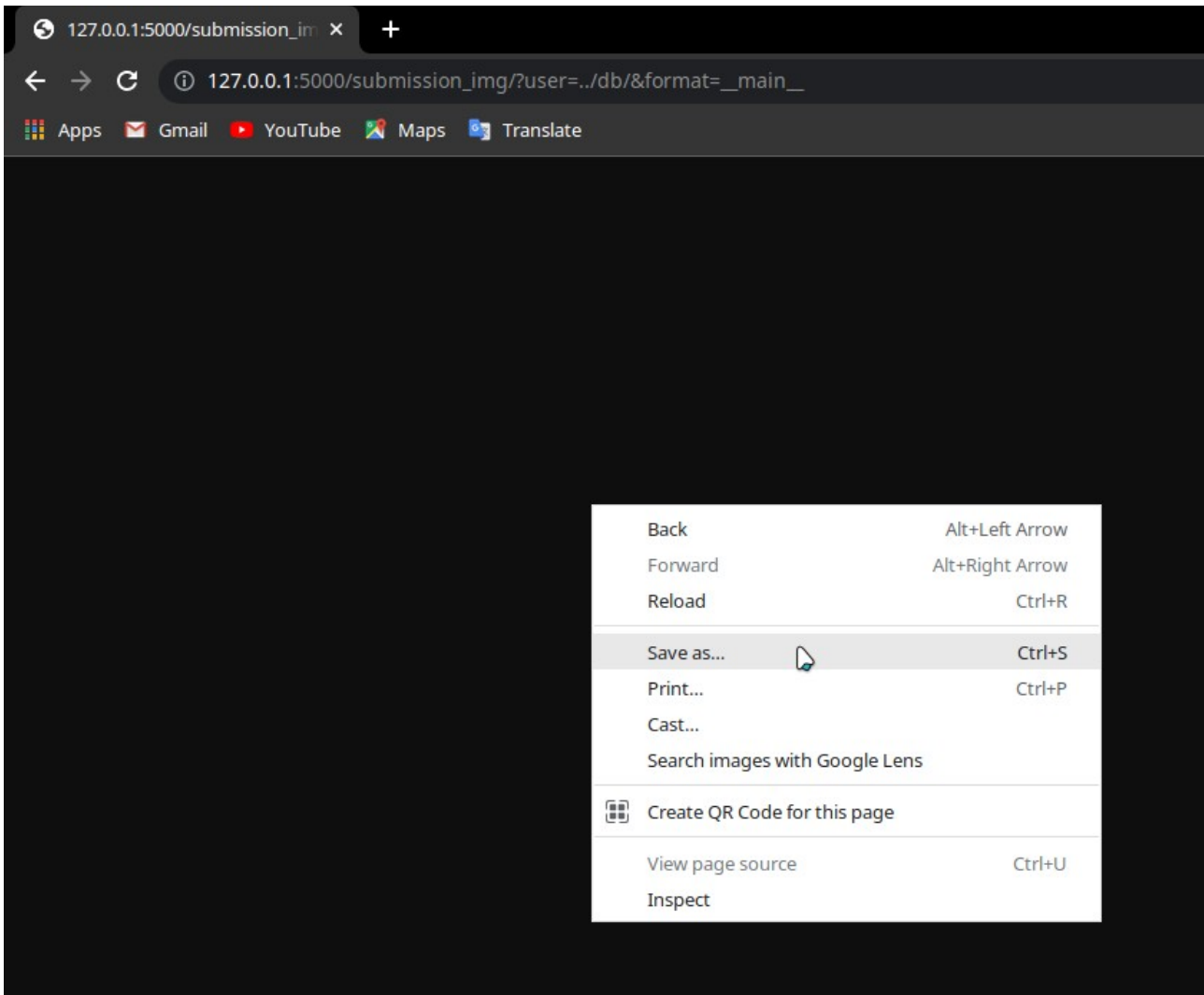
- Chyba je v 90. riadku submission.html kde je použité `{{ comment.user|safe }}`. `|safe` robí to, že jinja2 neodstráni nebezpečné html symboly. V `{{ comment.body }}` to nie je použité a preto je telo komentára ochránené proti XSS. (<https://stackoverflow.com/questions/48975383/why-to-use-safe-in-jinja2-python>)



- Zraniteľnosť umožňuje útočníkovi spustenie JS kódu na počítači obete.

Únik dát databázy

- Directory traversal
- Poslaním GET requestu na túto stránku:
http://127.0.0.1:5000/submission_img/?user=../db/&format=_main_
môžeme stiahnuť databázu zo stránky.
- Prehliadač nestiahne databázu automaticky, pretože očakáva obrázok a nie iný súbor.



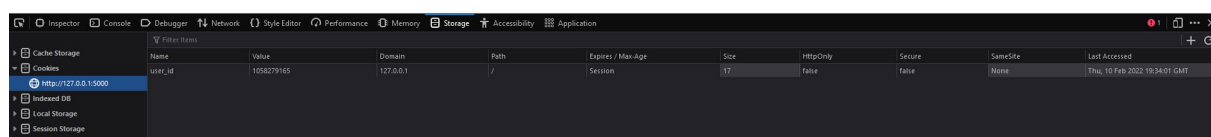
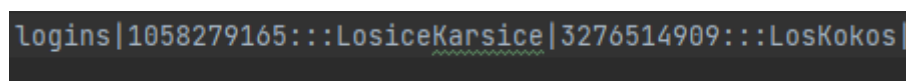
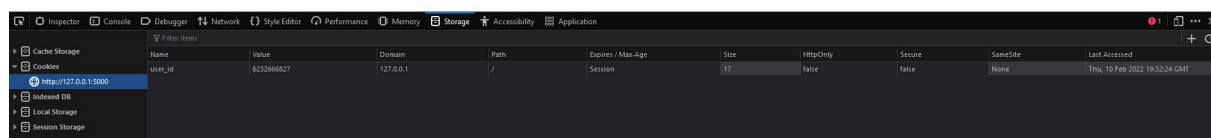
- Je spôsobená funkciou `submission_img()`, špecificky 60. riadok v `main.py`. Táto funkcia slúži na posielanie obrázkov použitých na stránke, `mimetype='image/png'` ale neochráni pred posielaním iných vecí, len pred správnym zobrazením.

```
59     assert img_format is not None
60     return send_file(DATA_DIR + '/' + user + img_format, mimetype='image/png')
61
```

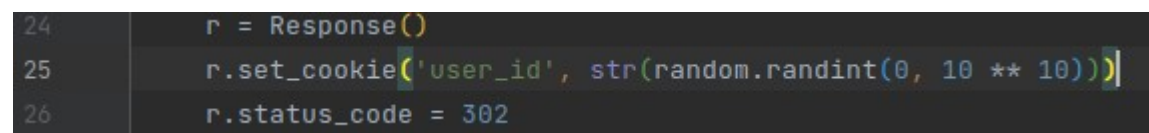
- Zraniteľnosť umožňuje útočníkovi prístup nielen ku databáze ale aj ku ostatným súborom servera na ktorom beží stránka.

Prihlásenie ako iný užívateľ

- Pass the cookie (<https://embracethered.com/blog/posts/passthecookie/>)
- Zadaním ID iného používateľa do cookie user_id sa môžeme prihlásiť bez toho aby sme poznali heslo.



- ID môžeme získať z databáze alebo brute-forcovaním čísla keďže je generované ako náhodné číslo od 0 do 10^{10} (v utils.py, 25. riadok).



- Zraniteľnosť umožňuje útočníkovi ukradnutie identity iného užívateľa.