Fix(Patch):

1,

Using escape, will not be interpreted and executed by the browser. Filter special characters in user input to prevent XSS： <script> alert("") </script> attacks



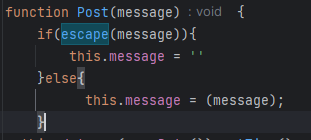
Test by:

<script> alert("") </script>

2,

Same as question 1, Filter special characters in user input to prevent XSS：<img src='' onerror="alert('')" /> attacks



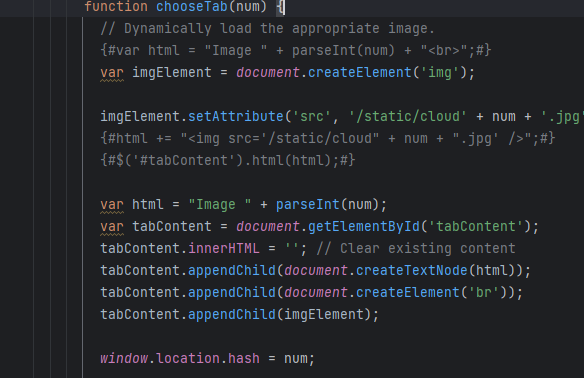
in the backend to determine if there’s js

Test by:

<img src='' onerror="alert('')" />

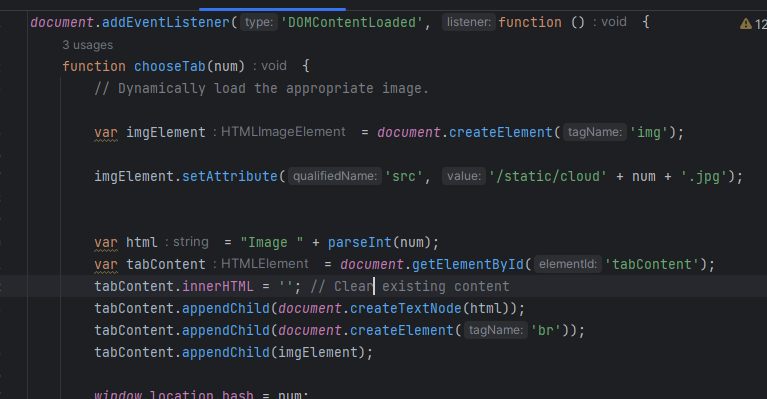
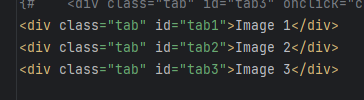
3，

avoid directly injecting untrusted data into the HTML

By use Dom to create and append with properly setting attributes, to avoid the injection of malicious: 'onerror="alert()"/> after number. And the console will get img report 404

Also further load js out of the html,





Using event listener to avoid the xss attack(onclick)

Test by:

[127.0.0.1:5000/level3/frame#1](https://xss-game.appspot.com/level3/frame#1)'onerror="alert()"/>

4,



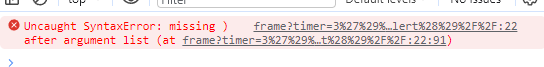
Convert the timer to an integer on the back end and check. If an exception occurs when converting to an integer, set timer to 3. And escape for further filtering.

If use |safe after timer instead the previous change:

A filter in Jinja2 that outputs the original value of a variable directly, avoiding the inclusion of special characters that affect script execution.

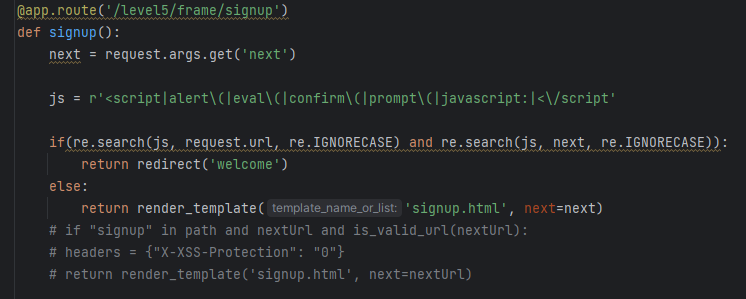
The console will report an error, and it can also be used together with the previous code





Test by enter 3'); alert('1

5,



The next string and url are filtered, if JS-related scripts are included, it will return to the initial page

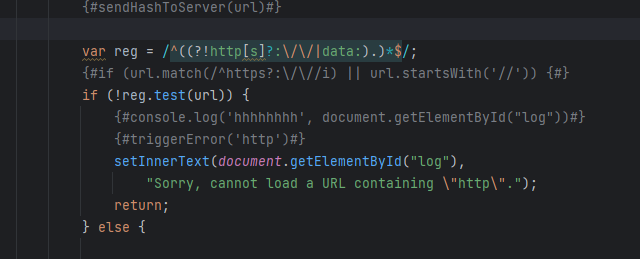
Also,change <a>tag to button tag, and in js, get the element by button id, to check if {{next}} param has malicious code. If has, block.

Test by: change from  
<http://127.0.0.1:5000/level5/frame/signup?next=confirm>

To  
[http://127.0.0.1:5000/level5/frame/signup?next=javascript:alert(1)](https://xss-game.appspot.com/level5/frame/signup?next=javascript:alert(1))

By enter, if it includes js, will return to welcome page

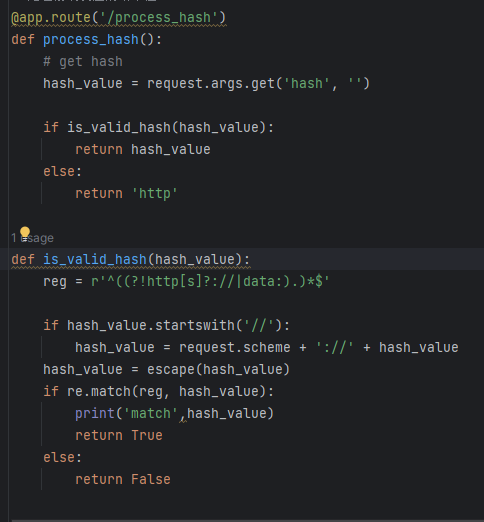
6,



Filter in the js, for any letter case http, https, data and //



Also send the url back to the backend to determine



Test by: [127.0.0.1:5000/last/frame#HTTPS://www.google.com/jsapi?callback=alert](https://xss-game.appspot.com/level6/frame#HTTPS:/www.google.com/jsapi?callback=alert)

[127.0.0.1:5000/last/frame#//www.google.com/jsapi?callback=alert](https://xss-game.appspot.com/level6/frame#HTTPS:/www.google.com/jsapi?callback=alert)

127.0.0.1:5000/last/frame#data:text,alert()