Secure Systems Engineering #1 - Add a content-security-policy

Æde Symen Hoekstra, Luigi Coniglio (Group 24)

June 22, 2018

1 Problem

The problem in the application is that it does not sanitise any input the user enters. For example, the user can enter: `<img src="https://i.imgur.com/vMOJDOk.jpg"/>`, which will render an image in the results page, see the example below. The browser will then load this image and display it. An attacker can misuse this by sending a URL to a results page which will load malicious content from an external website. For example: `http://localhost/?s=%3Cimg+src%3D%22https%3A%2F%2Fi.imgur.com%2FvMOJDOk.jpg%22%2F%3E`. This can be used to do for example cross site scripting attacks.

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>US Presidents</title>
</head>

<body>
<h1>US president search</h1>
<form action="/" method="GET">
<p>Type a part of the name here: <input type="text" name="s"/></p>
</form>

<h1>Searching for <img src="https://i.imgur.com/vMOJDOk.jpg"/></h1>
<ul>

</ul>

</body>

</html>
```

2 Solution

To prevent loading content from other websites, a content security policy can be added to the page. This can be done in two ways, the first option is to include this policy in the http headers, the second option is to add this option as meta tag in the HTML head section.
The first option is generally the better one because this is often handled in different parts of the application, and the second option can sometimes be bypassed by using an attack which makes the meta tag in the head into a comment. We therefore opted to implement to first option:

```
1 r = make_response(render_template("index.html", r=r, results=results))
2 r.headers.set('Content-Security-Policy', "default-src 'self'")
3 return r
```

We chose the strongest policy which only allows content to be loaded from the same domain. In this case it is fitting since the page only needs to load content from the same domain. In other cases some domain may need to be white-listed in the policy.