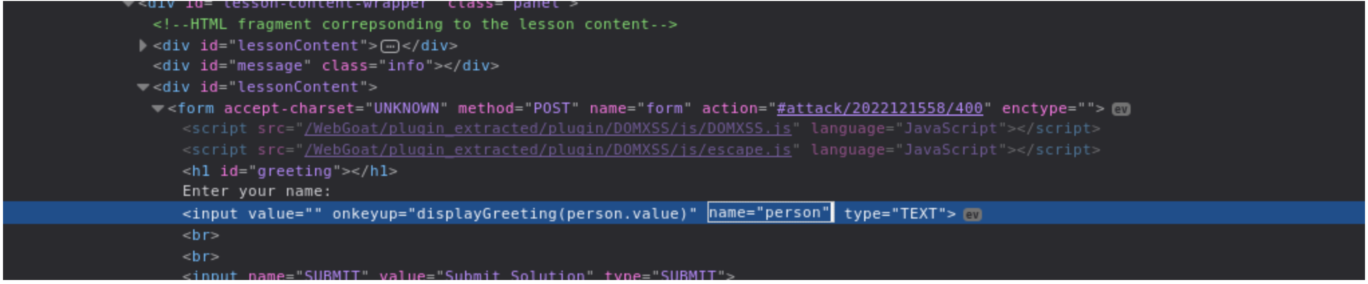
DOM-Based XSS

Thursday, October 19, 2017 7:41 AM

The Document Object Model (DOM) presents an interesting problem from a security standpoint.
It allows the content of a web page to be dynamically modified, but that can abused by attackers during a malicious code injection.

- XSS is a type of malicious code injection that can occur when unvalidated user input is used directly to modify the content of a page on the client side

<u>Solution</u>:
Inspect element of the text field input



```
▼<div id="lesson-content-wrapper" class="panel">
   <!--HTML fragment correpsonding to the lesson content-->
  ▶<div id="lessonContent">⋯</div>
   <div id="message" class="info"></div>
  ▼<div id="lessonContent">
    ▼<form accept-charset="UNKNOWN" method="POST" name="form" action="#attack/2022121558/400" enctype=""> ev
       <script src="/WebGoat/plugin_extracted/plugin/DOMXSS/js/DOMXSS.js" language="JavaScript"></script>
       <script src="/WebGoat/plugin_extracted/plugin/DOMXSS/js/escape.js" language="JavaScript"></script>
       <h1 id="greeting"></h1>
       Enter your name:
       <input value="" onkeyup="displayGreeting(person.value)" name="person" type="TEXT"> ev
       <br>
       <br>
       <input name="SUBMIT" value="Submit Solution" type="SUBMIT">
```

We can see that input field is using a JavaScript function named displayGreeting, which is listed in the script tag above.
Navigate to that link to see the JS source code.
Here we see…

```
function displayGreeting(name) {
        if (name != ''){
                document.getElementById("greeting").innerHTML="Hello, " + name+ "!";
        }
}
```

The user is inputing directly into the JS code.
Because of this, the user can simply type " in the input field to close the string and then insert malicious JavaScript code directly onto the server.
In this case, we are inserting a .jpg file onto the webpage.
e.g. we will enter..
"<IMG src=images/logos/owasp.jpg>
Into the form field