


Closed Opened 1 week ago by  **Jannis Kirschner**

## XSS

In the following file:

source-code\online-voting-channel\ov-commons\ov-commons-correctness\src\main\java\com\scytll\products\ov\commons\correctness\builder\attributes\ContestCorrectionBuilder.java

We found this code:

```
public void generateCorrectnessCode(StringBuilder sb) {
    Set<Entry<String, Integer>> entrySet = questions.entrySet();
    for (Entry<String, Integer> entry : entrySet) {
        sb.append("result = function (subSelection, callbackFunction) {");
        sb.append("var partialResult = true;");
        sb.append("var selectionName = ").append(contestId).append(";");
        sb.append("var attributeName = ").append(entry.getKey()).append(";");
        sb.append("var max = ").append(entry.getValue().intValue()).append(";");
        sb.append("var count = 0;");
    }
}
```

This line seems to be vulnerable: `sb.append("var selectionName = ").append(contestId).append(";");`

gets called here:

```
public ContestCorrectionBuilder(String contestId, AttributesCorrectnessBuilder attributesCorrectnessBuilder) {
    this.contestId = contestId;
    this.attributesCorrectnessBuilder = attributesCorrectnessBuilder;
}
```

None of the calls are escaped. Which means that somebody that gets access to a context object can inject dynamic js code and for example redirect to shady sites.

A safe example would be using a js array and converting it to a string or escaping the provided strings.

Best Regards

Jannis Kirschner aka xorkiwi

Anthony Schneider aka muffinx



[Swisspost @swisspost](#) added In Progress topic: general labels 1 week ago



[Swisspost @swisspost](#) · 1 week ago

Maintainer

Hello,

Thank your for your input. We will verify this and get back to you.

Kind regards



[Swisspost @swisspost](#) · 3 days ago

Maintainer

Hello Jannis,

Thank you for your observation. The Javascript vote correctness function is generated prior to the configuration of the election. No user-generated data is passed to this function. While we are thinking to replace the javascript function with something more static, we do not think that the issue qualifies as a vulnerability, since exploitation would require having full access to the administrator portal and manipulating the internal java objects during code execution), but the possibility of exploiting this vulnerability should be considered very low, as in case the direct access to the java objects are available to an attacker, then the attacker have full control over the election configuration, not only the creation of the vote correctness rules.

Does this answer suffice to you?

Kind regards



[Swisspost @swisspost](#) closed [3 days ago](#)



[Swisspost @swisspost](#) removed [In Progress](#) label [1 day ago](#)



[Swisspost @swisspost](#) made the issue visible to everyone [1 day ago](#)