Remote Code Execution in ParametersParser while using request parameters inside expression language

Critical pamil published GHSA-p4pj-9g59-4ppv on Aug 18, 2020

Package

php sylius/resource-bundle (Composer)

Affected versions	Patched versions
<=1.3.13 || >=1.4.0 <=1.4.6 || >=1.5.0 <=1.5.1 || >=1.6.0 <=1.6.3	1.3.14, 1.4.7, 1.5.2, 1.6.4

Description

Impact

Request parameters injected inside an expression evaluated by `symfony/expression-language` package haven't been sanitized properly. This allows the attacker to access any public service by manipulating that request parameter, allowing for Remote Code Execution.

The vulnerable versions include: `<=1.3.13 || >=1.4.0 <=1.4.6 || >=1.5.0 <=1.5.1 || >=1.6.0 <=1.6.3`.

Example

```
foo:
    path: /foo/{id}
    defaults:
        _sylius:
            repository:
                method: findSome
                arguments:
                    entity: "expr:service('repository').find($id)"
```

In this case, `$id` can be prepared in a way that calls other services.

If you visit `/foo/"~service('doctrine').getManager().getConnection().executeQuery("DELETE * FROM TABLE")~"` , it will result in a following expression `expr:service('repository').find(""~service('doctrine').getManager().getConnection().executeQuery("DELETE * FROM TABLE")~"")` , which will execute a query on the currently connected database.

To find a vulnerability in your application, look for any routing definition that uses request parameters inside expression language.

Patches

This issue has been patched for versions 1.3.14, 1.4.7, 1.5.2 and 1.6.4. Versions prior to 1.3 were not patched.

Workarounds

The fix requires adding `addslashes` in `ParametersParser::parseRequestValueExpression` to sanitize user input before evaluating it using the expression language.

```
- return is_string($variable) ? sprintf('"%s"', $variable) : $variable;
+ return is_string($variable) ? sprintf('"%s"', addslashes($variable)) : $variable;
```

Acknowledgements

This security issue has been reported by Craig Blanchette (**@isometriks**), thanks a lot!

For more information

If you have any questions or comments about this advisory:

- Email us at security@sylius.com

Severity

Critical

CVE ID

CVE-2020-15143

Weaknesses

No CWEs

Credits

isometriks