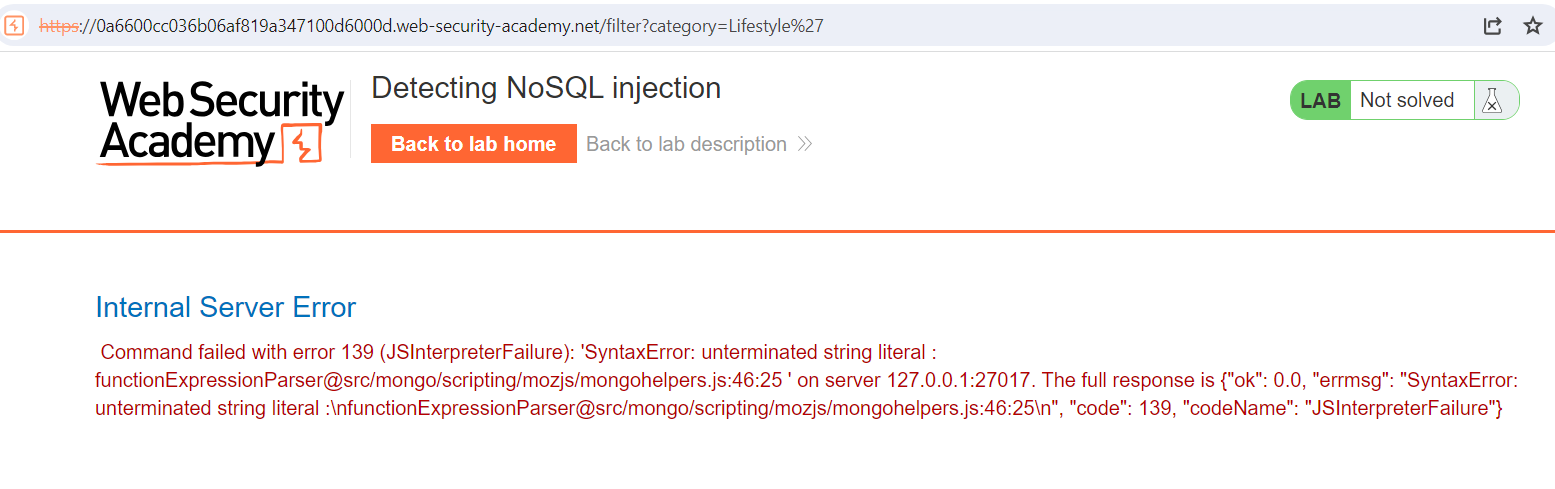
**NoSQL injection**

**LAB 103** [**Detecting NoSQL injection**](https://portswigger.net/web-security/nosql-injection/lab-nosql-injection-detection)

Dealing with websites catalogue, I tested it on SQL injections, by adding single quote mark ‘ in the end of category parameter inside URL and faced such error:



Examining the error message, I can see that the service uses MongoDB that does not use SQL queries, so further it will be tested on NoSQL injections.

As next thing, I tried to inject a valid JS payload into URL: ?category=Lifestyle’+’, fully encoding it into URL:

Изображение выглядит как текст, снимок экрана, Веб-сайт, веб-страница

Автоматически созданное описание

Now, I do not see any error.

The next thing worth testing is to check if it is possible to operate with Boolean expressions. I prepared a simple payload for this, that will trigger a false condition:

?category=Lifestyle‘ && 0 && ‘x

The result:



It does not show any products in any of categories (because it’s false condition). And for true condition, we should change to: Lifestyle’ && 1 && ‘x

Изображение выглядит как текст, снимок экрана, Веб-сайт, веб-страница

Автоматически созданное описание

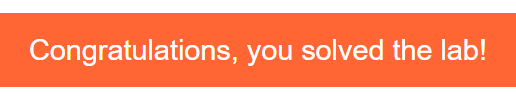
Now, the products of ‘Lifestyle’ category are available.

Constructing a payload that will always trigger a true condition will reveal unreleased products:

Изображение выглядит как текст, снимок экрана, Реклама в Интернете, веб-страница

Автоматически созданное описание

Lab’s done!



**LAB 103** [**Exploiting NoSQL operator injection to bypass authentication**](https://portswigger.net/web-security/nosql-injection/lab-nosql-injection-bypass-authentication)

Valid credentials: wiener:peter

Goal: log in as administrator

Browsing the website, I tested the functionality for logging in and tried my valid credentials, Here are the contents of POST /login:

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Since it is known, that MongoDB is used, I prepared a NoSQL injection using operators $ne (not equal) and tried to bypass the authentication functionality, The following payload will grab all existing fields for username and password:

Изображение выглядит как текст, программное обеспечение, веб-страница, Шрифт

Автоматически созданное описание  
As expected, I cannot log in as multiple users, so I changed the username parameter to {“$in”: [“admin”, “administrator”]}. However, the result ended up with error “Invalid username or password”, so I had to change the payload to

{“$regex” : “admin.\*”}, to find any username, containing admin in the beginning:

Изображение выглядит как текст, снимок экрана, программное обеспечение, число

Автоматически созданное описание

It resulted to a successful log in as admin92rcrg0l

Изображение выглядит как текст, снимок экрана, Шрифт, веб-страница

Автоматически созданное описание

**LAB 104** [**Exploiting NoSQL injection to extract data**](https://portswigger.net/web-security/nosql-injection/lab-nosql-injection-extract-data)

Valid credentials: wiener:peter

Goal: log in as administrator

The user lookup functionality for this lab is powered by a MongoDB NoSQL database. It is vulnerable to NoSQL injection.

Изображение выглядит как текст, снимок экрана, Шрифт, линия

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, Шрифт, линия

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, Шрифт, линия

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, Шрифт, линия

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, Шрифт, линия

Автоматически созданное описание