**25Q4 Alumni-dev Portal SQL injection Test**

### Introduction:

The applications that are owned by Zalaris are actively tested for vulnerability using manual Penetration test

# 

Table of Contents

[Introduction: 2](#_Toc20000)

[2025 Q4 Alumni Portal SQL Injection Test 4](#_Toc28547)

### 2025 Q4 Alumni Portal SQL Injection Test

**Description**

Manual Test is done against alumni portal for SQL injection

**Steps:**

The URLs parameters and endpoints found are given below

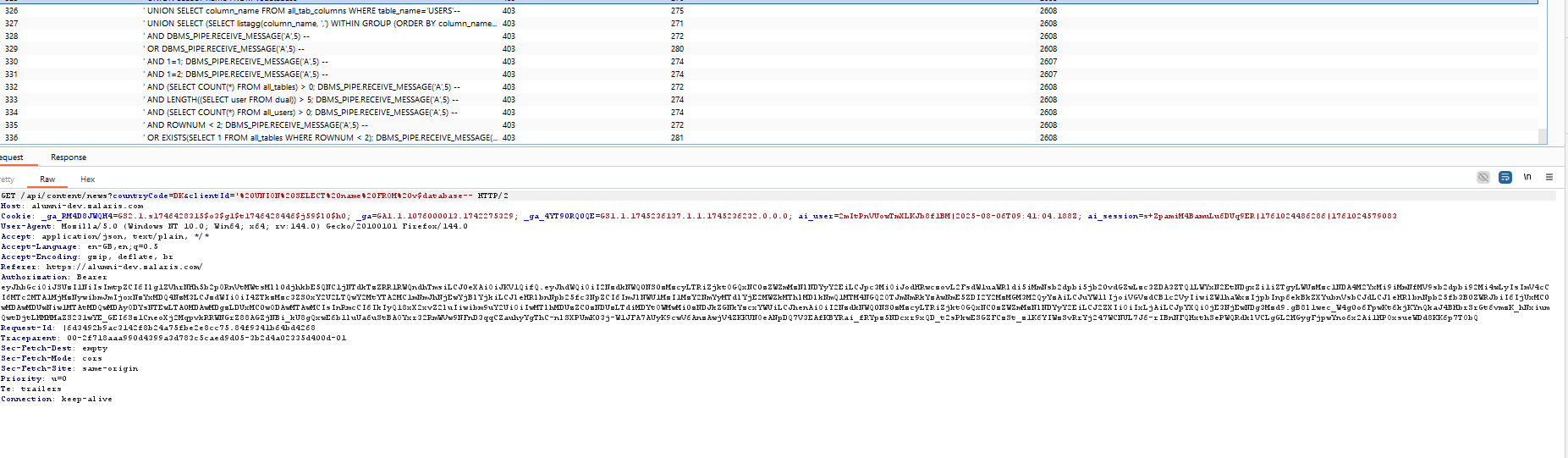
|  |  |  |
| --- | --- | --- |
| URL ID | URL | Comments |
| 1 | <https://alumni-dev.zalaris.com/api/content/news> | no endpoint or injection point |
| 2 | <https://alumni-dev.zalaris.com/api/content/news?countryCode=DK&clientId=510> | -countryCode  -clientID |
| 3 | <https://alumni-dev.zalaris.com/api/employees/510-00000506/personal-data> | no endpoint or injection point |
| 4 | <https://alumni-dev.zalaris.com/api/employees/510-08001000/documents> | no endpoint or injection point |
| 5 | <https://alumni-dev.zalaris.com/api/employees/510-04000083/payroll-results> | no endpoint or injection point |
| 6 | <https://alumni-dev.zalaris.com/api/content/news?countryCode=NO&clientId=510> | no endpoint or injection point |
| 7 | <https://alumni-dev.zalaris.com/api/content/67e10698e962d87541197d7a> | no endpoint or injection point |

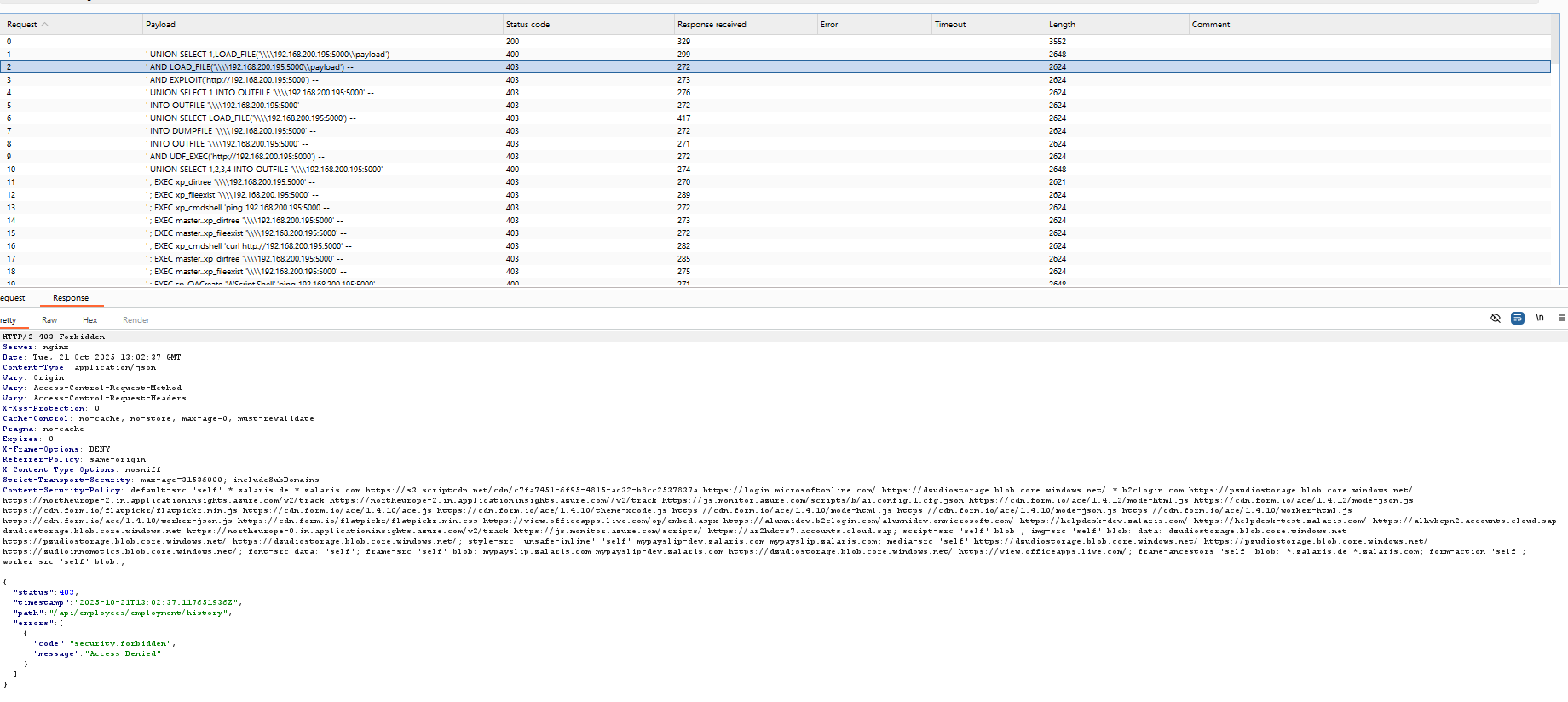
Attack Result

|  |  |  |  |
| --- | --- | --- | --- |
| **Payload Type** | **URL Id** | **Parameter** | **Vulnerable / Not** |
| Error Based SQLi Payload | 2 | CountryCode | Not Vunerable |
| Union Based SQLi Payload | 2 | Not Vunerable |
| Boolean Based Blind SQLi Payload | 2 | Not Vunerable |
| Time Based SQLi Payload | 2 | Not Vunerable |
| Out Of Band SQLi Payload | 2 | Not Vunerable |
| Stored Procedure SQLi Payload | 2 | Not Vunerable |
| Second Order SQLi Payloads | 2 | Not Vunerable |
| Stacked Queries SQLi Payloads | 2 | Not Vunerable |
| WAF Bypass Techniques | 2 | Not Vunerable |
| DNS exfiltration Technique | 2 | Not Vunerable |
| Hybrid SQLi Technique | 2 | Not Vunerable |
| Comment Based SQLi Technique | 2 | Not Vunerable |

|  |  |  |  |
| --- | --- | --- | --- |
| **Payload Type** | **URL Id** | **Parameter** | **Vulnerable / Not** |
| Error Based SQLi Payload | 2 | clientID | Not Vunerable |
| Union Based SQLi Payload | 2 | Not Vunerable |
| Boolean Based Blind SQLi Payload | 2 | Not Vunerable |
| Time Based SQLi Payload | 2 | Not Vunerable |
| Out Of Band SQLi Payload | 2 | Not Vunerable |
| Stored Procedure SQLi Payload | 2 | Not Vunerable |
| Second Order SQLi Payloads | 2 | Not Vunerable |
| Stacked Queries SQLi Payloads | 2 | Not Vunerable |
| WAF Bypass Techniques | 2 | Not Vunerable |
| DNS exfiltration Technique | 2 | Not Vunerable |
| Hybrid SQLi Technique | 2 | Not Vunerable |
| Comment Based SQLi Technique | 2 | Not Vunerable |

Supporting screenshots for the injection done are attached below



****

**WORKLOG**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| When | What | Who | Inference | Action plan |
| 21-10-2025 | Manually tested Alumni-dev.zalaris.com application for SQL injection | Mohammed Sharoz | The application is not vulnerable to SQL Injection | No Action Needed |