

**Finding Name:** Unencrypted Communications.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Team** | **Role** | **Project** | **Quality Assurance** | **Is this a re-tested Finding?** |
| Donghwan Kim | PT | Junior Team Member | Ontrack |  |  |
|  |  |  |  |  |  |

|  |
| --- |
| **Was this Finding Successful?** |
| Yes |

**Finding Description**

I found a vulnerability in SQL Injection, one of the very easy security vulnerabilities. SQL Injection is a technique that maliciously manipulates SQL queries used in web applications to hack databases.  
 A vulnerability I found during SQL Injection is how to run a random text program into your account to steal your account by substituting username and password.

**Risk Rating**  
Impact: Minor   
Likelihood: Significant

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Impact values** | | | | |
| **Very Minor** | **Minor** | **Significant** | **Major** | **Severe** |
| Risk that holds little to no impact. Will not cause damage and regular activity can continue. | Risk that holds minor form of impact, but not significant enough to be of threat. Can cause some damage but not enough to impede regular activity. | Risk that holds enough impact to be somewhat of a threat. Will cause damage that can impede regular activity but will be able to run normally. | Risk that holds major impact to be of threat. Will cause damage that will impede regular activity and will not be able to run normally. | Risk that holds severe impact and is a threat. Will cause critical damage that can cease activity to be run. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Likelihood** | | | | |
| **Rare** | **Unlikely** | **Moderate** | **High** | **Certain** |
| Event may occur and/or if it did, it happens in specific circumstances. | Event could occur occasionally and/or could happen (at some point) | Event may occur and/or happens. | Event occurs at times and/or probably happens a lot. | Event is occurring now and/or happens frequently. |

**Business Impact**

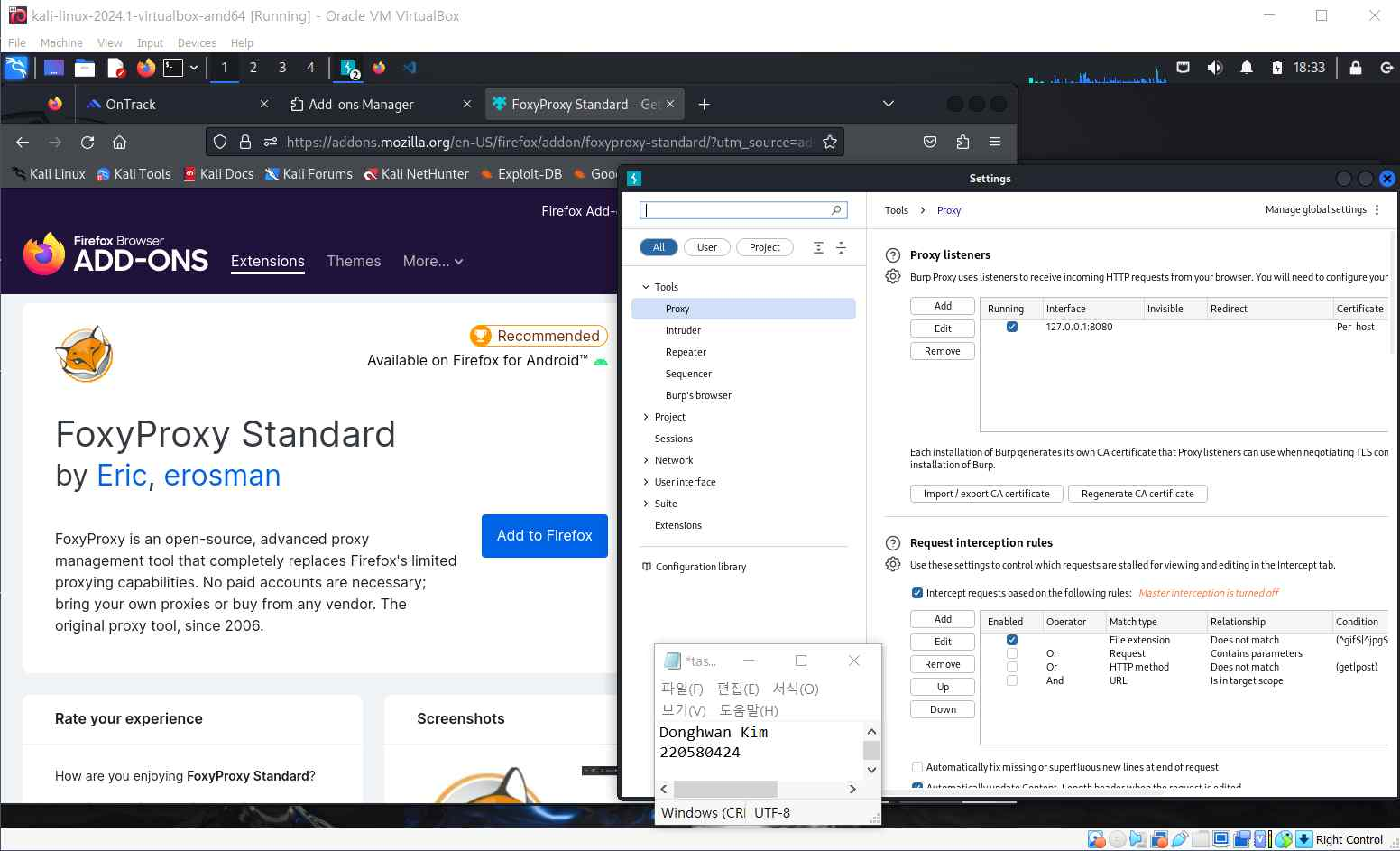
Explain how the finding will impact the company and the application as a whole

**Affected Assets**

Identify assets that are affected by this vulnerability.

**Evidence**

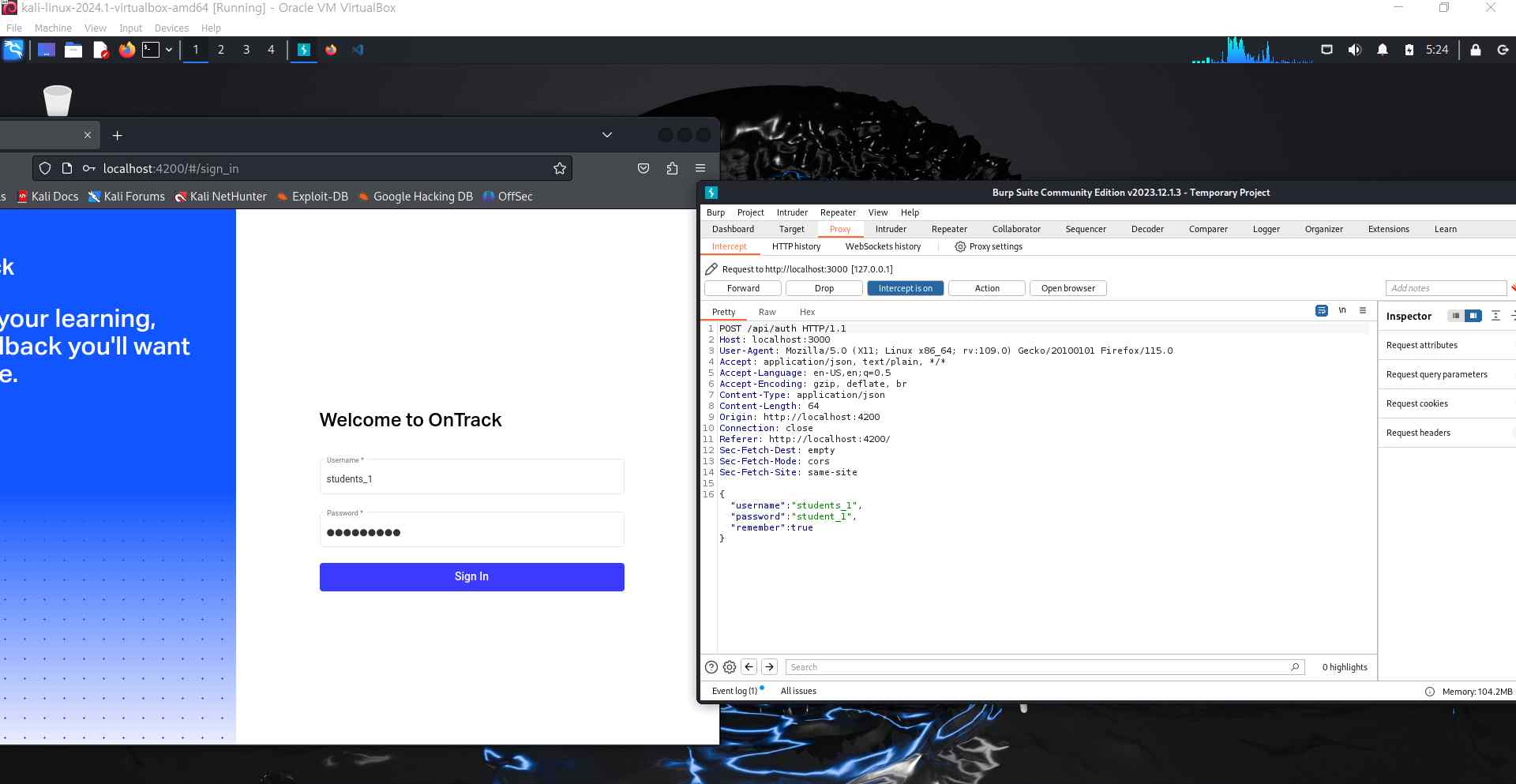
Provide a step by step guide on how to reproduce the vulnerability with screenshots

**Step 1. **

**I added foxyproxy on firefox and matched the host name and port number of Buff**

**Suite to prepare to use Buff Suite.**

**Step 2.**

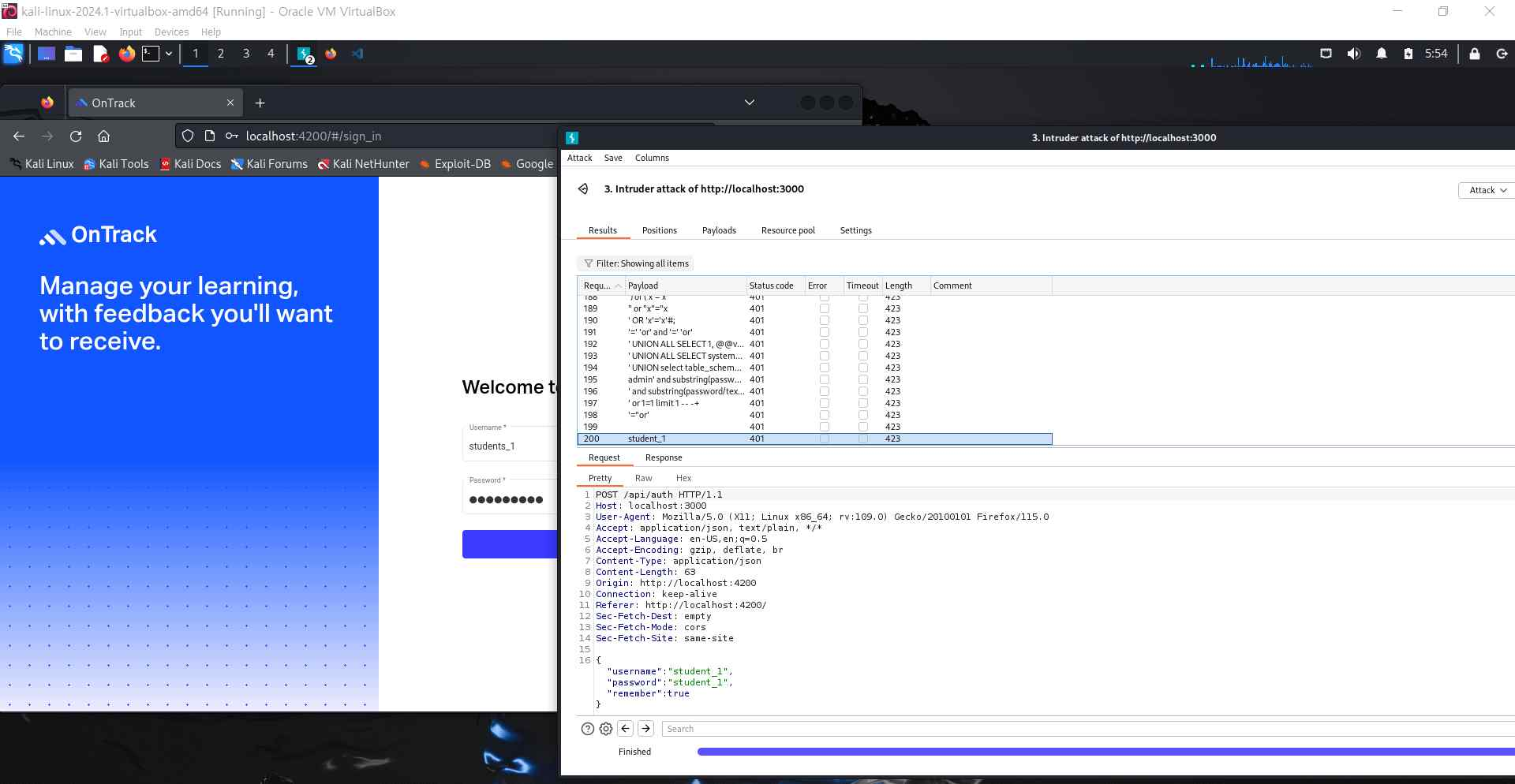
****

**I turned on the intercept of the proxy in Buff Suite.**

**This is a concept that allows you to intercept the proxy of localhost:4200 in the**

**middle to know the operation status within Burp Suite.**

**Step 3.**

****

**I ran a SQL Injection attack and was able to find the same ID as the current one.**

**That is, it was found that localhost:4200 has a vulnerability in SQL**

**Etc..**

**Remediation Advice**

In order to resolve this vulnerability in SQL injection, it seems necessary to use a method of verifying and rescuing the values entered.  
 Escaping is the conversion of special characters that can be used in SQL queries to prevent malicious code from being executed.

**References**

<https://portswigger.net/burp/documentation/desktop/testing-workflow/input-validation/sql-injection/testing>

Testing for SQL injection vulnerabilities with Burp Suite

(March, 2024)

**Testing for SQL injection vulnerabilities with Burp Suite**

**Contact Details**

Donghwan Kim (220580424)

kimdongh@deakin.edu.au

**Pentest Leader Feedback.**

The lead will provide feedback to enact on.