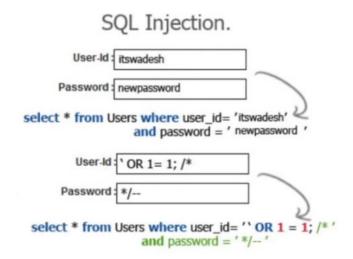
## Relatório de vulnerabilidade web

## Escolha do laboratório:

- O laboratório que eu escolhi foi "Lab: SQL injection vulnerability in WHERE clause allowing retrieval of hidden data". A minha primeira estrategia usada foi me basear nos comandos aprendidos na get, segue o exemplo abaixo:
- Durante a aula, foi dito que existem diversas formas de fazer comentários em Mysql.

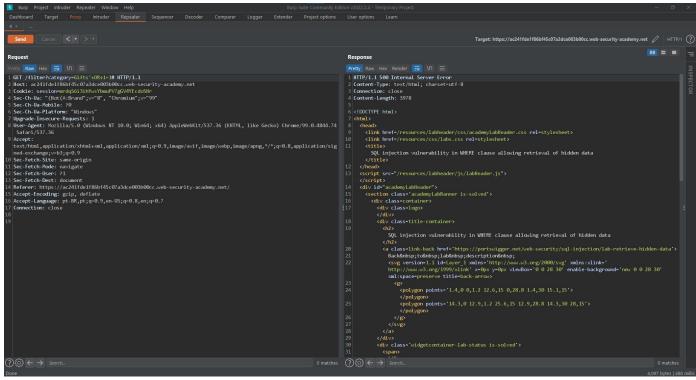


## Formas de fazer comentários em Mysql:

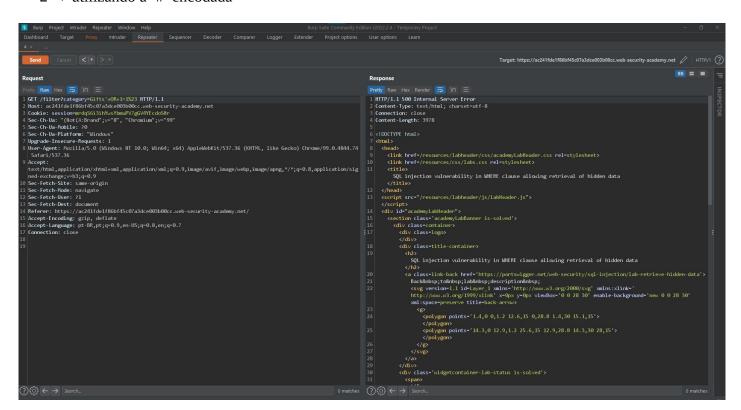
## **Tentativas frustradas:**

- o servidor esta retornando uma resposta de erro "500"

#### 1 → utilizando #



# 2 → utilizando a '#' encodada



## **Comando correto:**

- a forma correta de fazer o comentário era utilizando de traço duplo
- o servidor esta retornando a resposta de exito "200"
- o resultado foi maior, cortei a imagem apenas para caber no arquivo de relatório, ao todo foram 233 linhas de html.

#### Conclusão:

content-lenght da resposta correta = 12874 content-lenght das respostas erradas = 3978

Se compararmos a resposta correta com as respostas erradas, notamos que o tamanho do conteúdo retornado foi muito maior na resposta correta, 12874 > 3978, ou seja, obtivemos o retorno de mais dados do que a pagina "front" permite ver.

Essa vulnerabilidade permitiria ao invasor injetar comandos Sql, para apagar dados importantes por meio de comandos como "DROP TABLE table\_name;" , isso se o banco de dados permitir o uso de stacked queries.