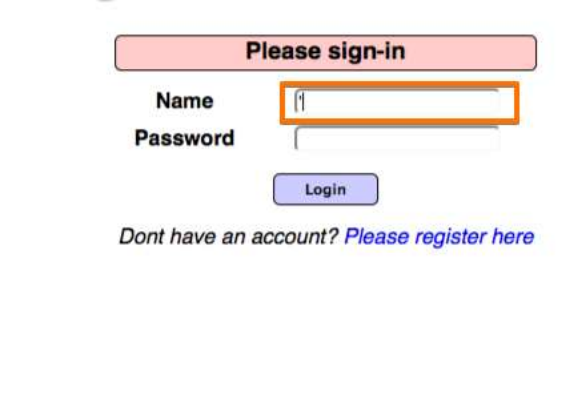
Using SQL Injection to Bypass Authentication

In this example we will demonstrate a technique to bypass the authentication of a vulnerable login page using SQL injection.

This tutorial uses an exercise from the "Mutillidae" training tool taken from OWASP's Broken Web Application Project. Find out how to download, install a this project.

To check for potential SQL injection vulnerabilities we have entered a single quote in to the "Name" field and submitted the request using the "Login" button.

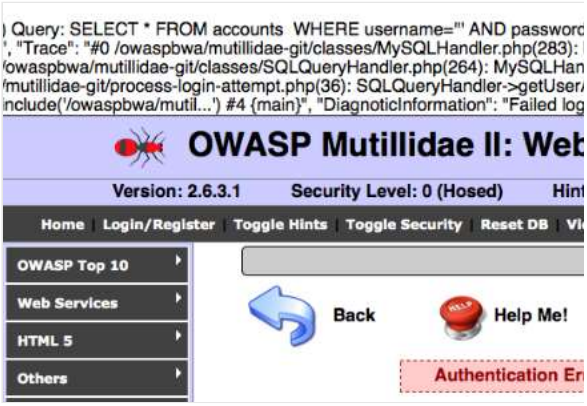


The application provides us with an SQL error message.

The error message includes the SQL query used by the login function.

We can use this information to construct an injection attack to bypass authentication.

The first account in a database is often an administrative user, we can exploit this behavior to log in as the first user in the database.



Enter some appropriate syntax to modify the SQL query into the "Name" input.

In this example we used ' or 1=1 -- .

This causes the application to perform the query:

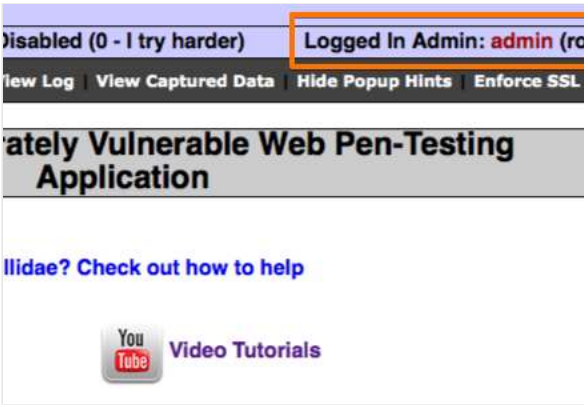
```
SELECT * FROM users WHERE username = '' OR 1=1-- ' AND
password = 'foo'
```

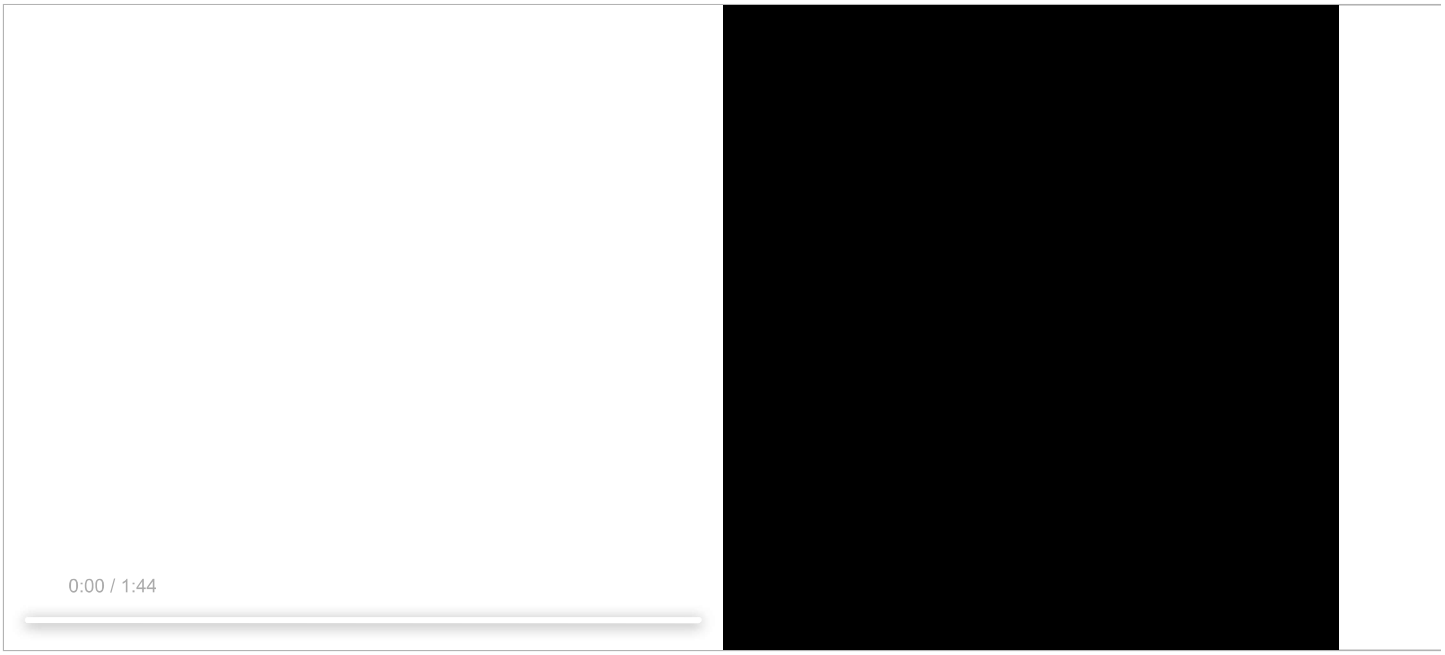
Because the comment sequence (--) causes the remainder of the query to be ignored, this is equivalent to:

```
SELECT * FROM users WHERE username = ' ' OR 1=1
```

In this example the SQL injection attack has resulted in a bypass of the login, and we are now authenticated as "admin".

You can learn more about this type of detection in our article; Using Burp to Detect Blind SQL Injection Bugs.





Related articles:

Using Burp to Test For SQL Injection Flaws