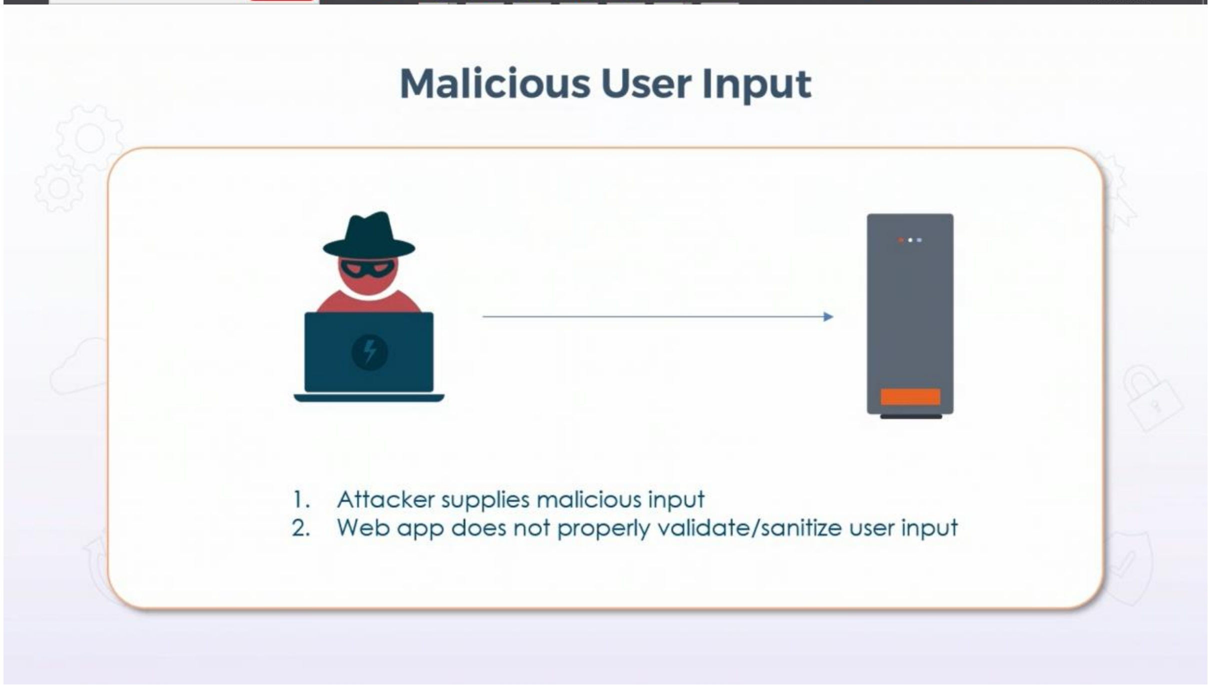
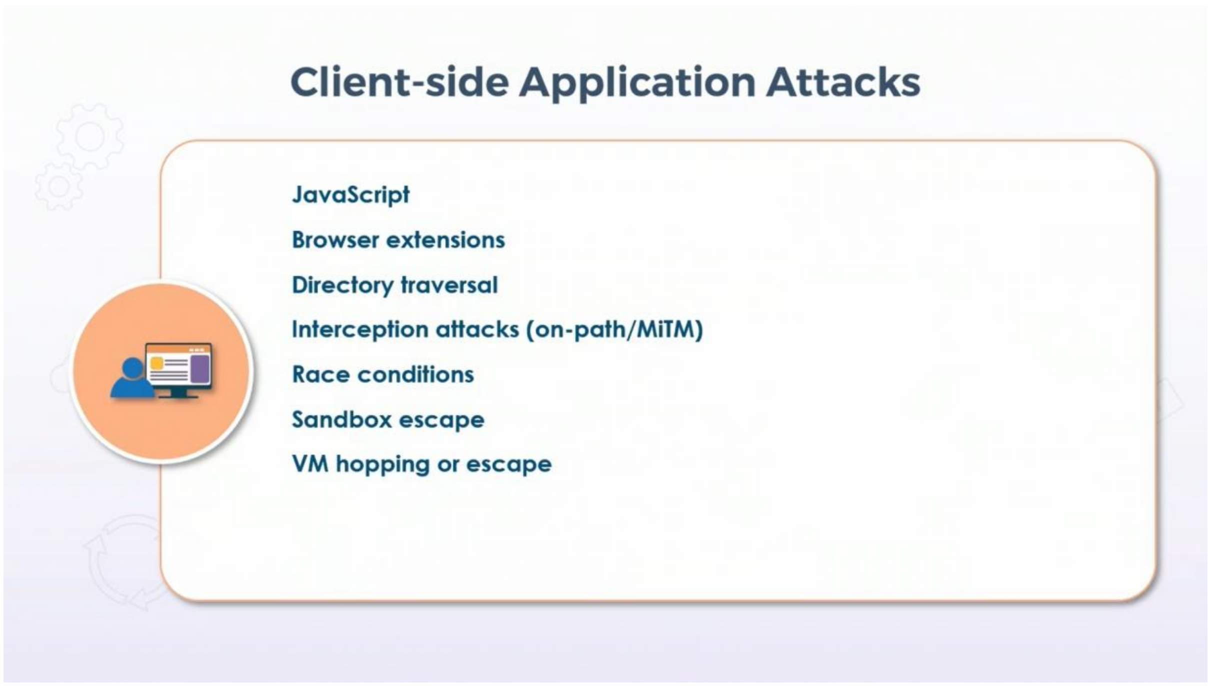
## Injection Attacks

Malicious input is accepted by the web application



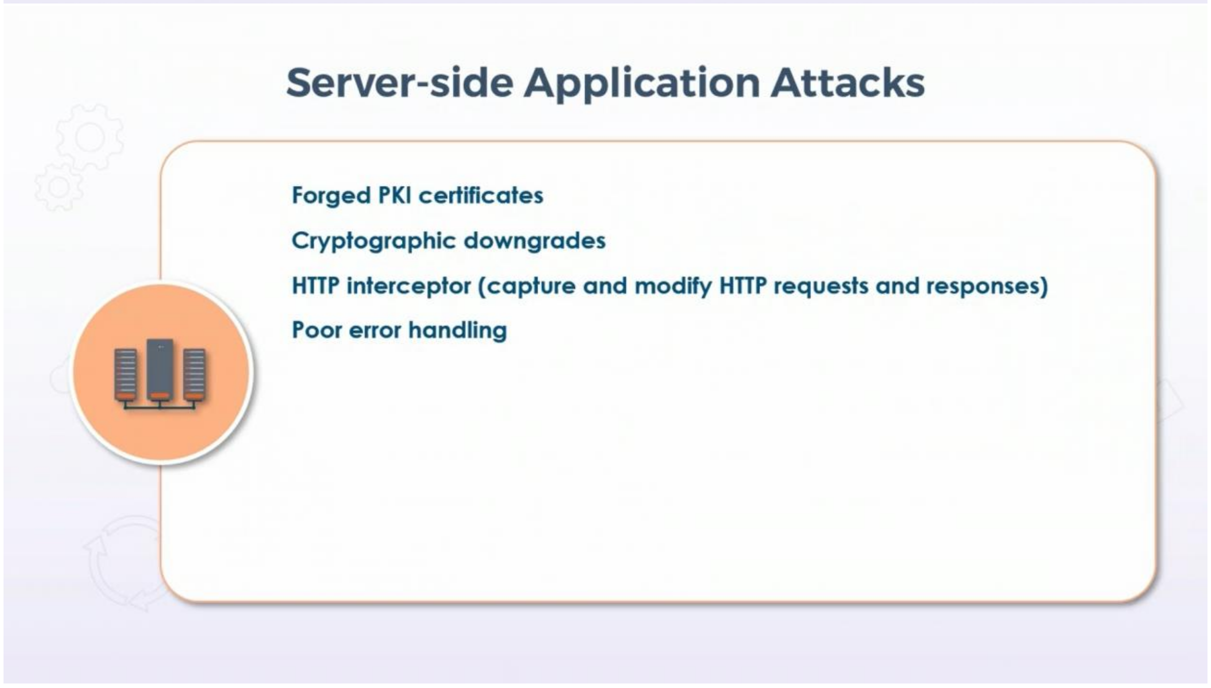
## Malicious User Input

1. Attacker supplies malicious input
2. Web app does not properly validate/sanitize user input



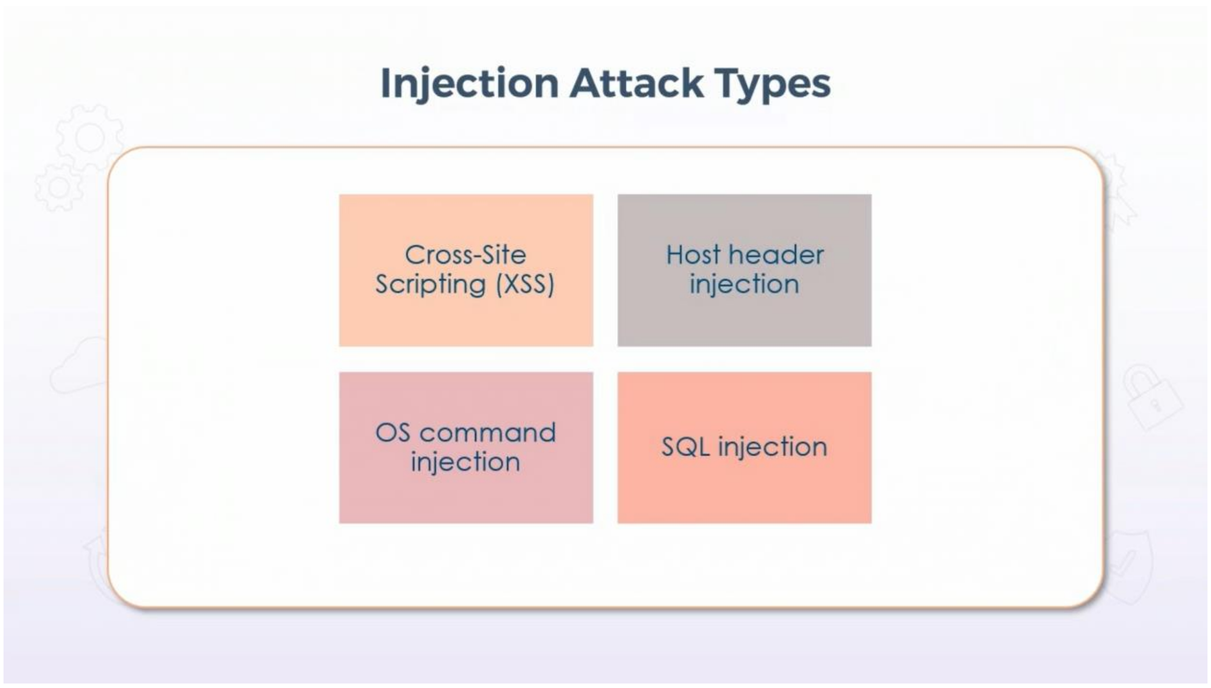
## Client-side Application Attacks

- JavaScript
- Browser extensions
- Directory traversal
- Interception attacks (on-path/MiTM)
- Race conditions
- Sandbox escape
- VM hopping or escape



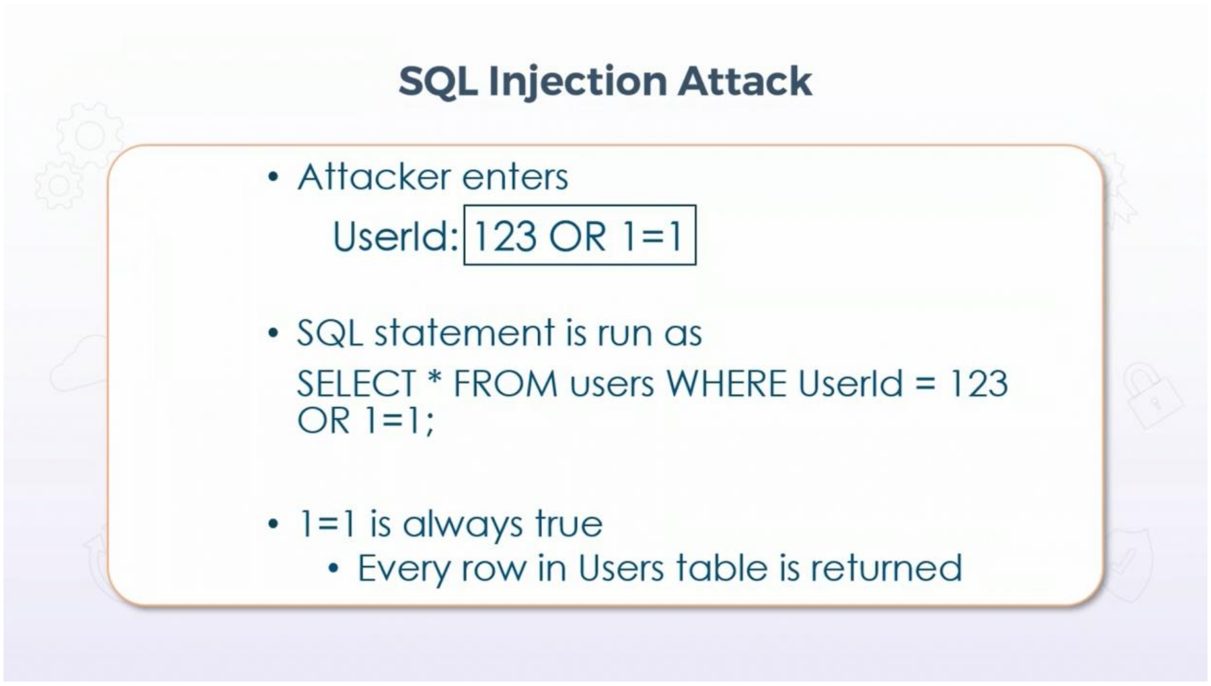
## Server-side Application Attacks

- Forged PKI certificates
- Cryptographic downgrades
- HTTP interceptor (capture and modify HTTP requests and responses)
- Poor error handling



## Injection Attack Types

- Cross-Site Scripting (XSS)
- Host header injection
- OS command injection
- SQL injection



## SQL Injection Attack

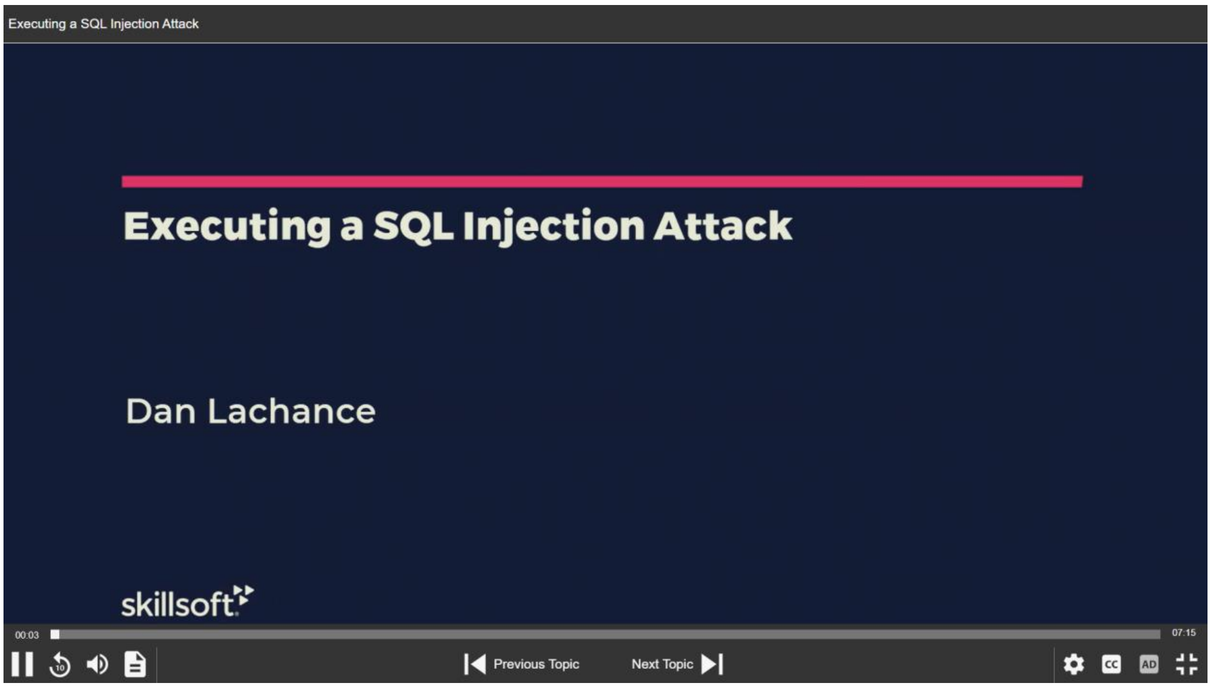
- Attacker enters

  UserId: 123 OR 1=1

- SQL statement is run as

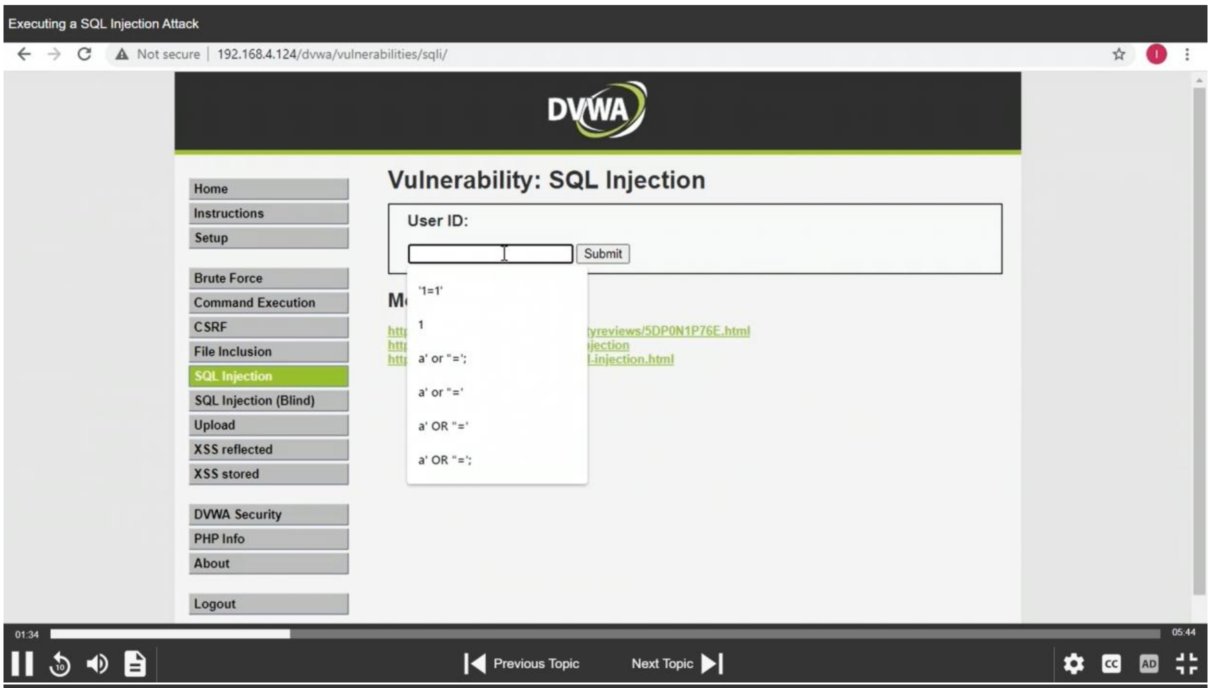
  SELECT * FROM users WHERE UserId = 123 OR 1=1;

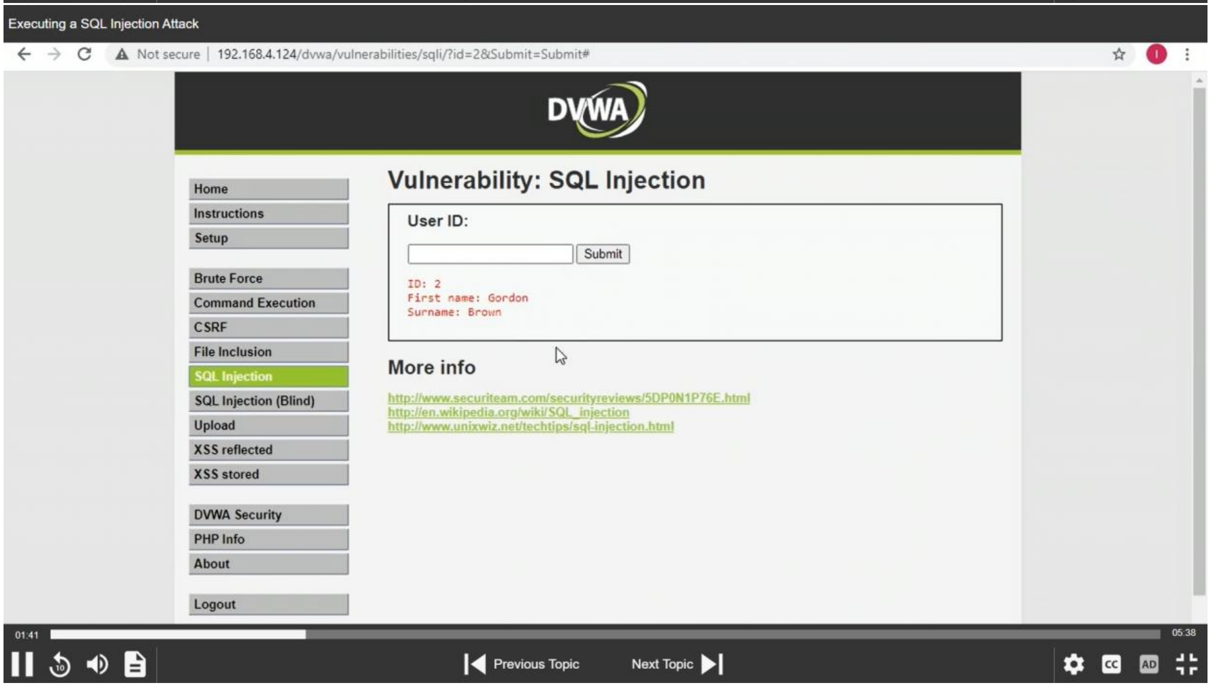
- 1=1 is always true
  - Every row in Users table is returned


Executing a SQL Injection Attack

# Executing a SQL Injection Attack

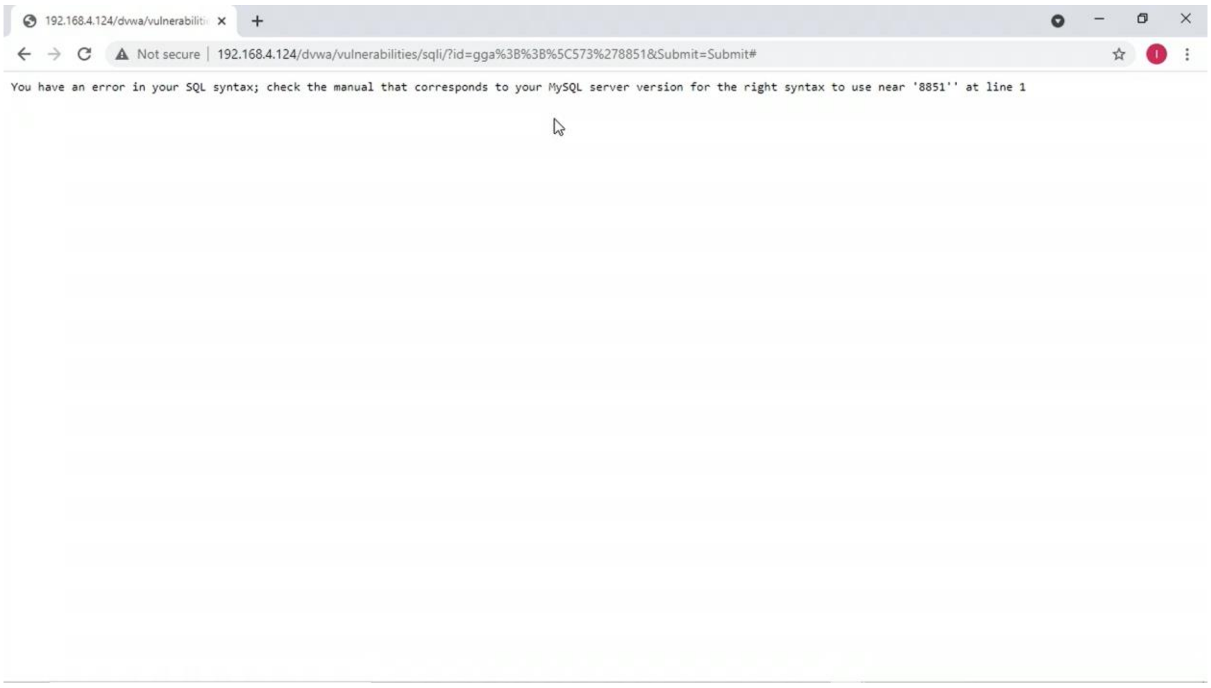
Dan Lachance



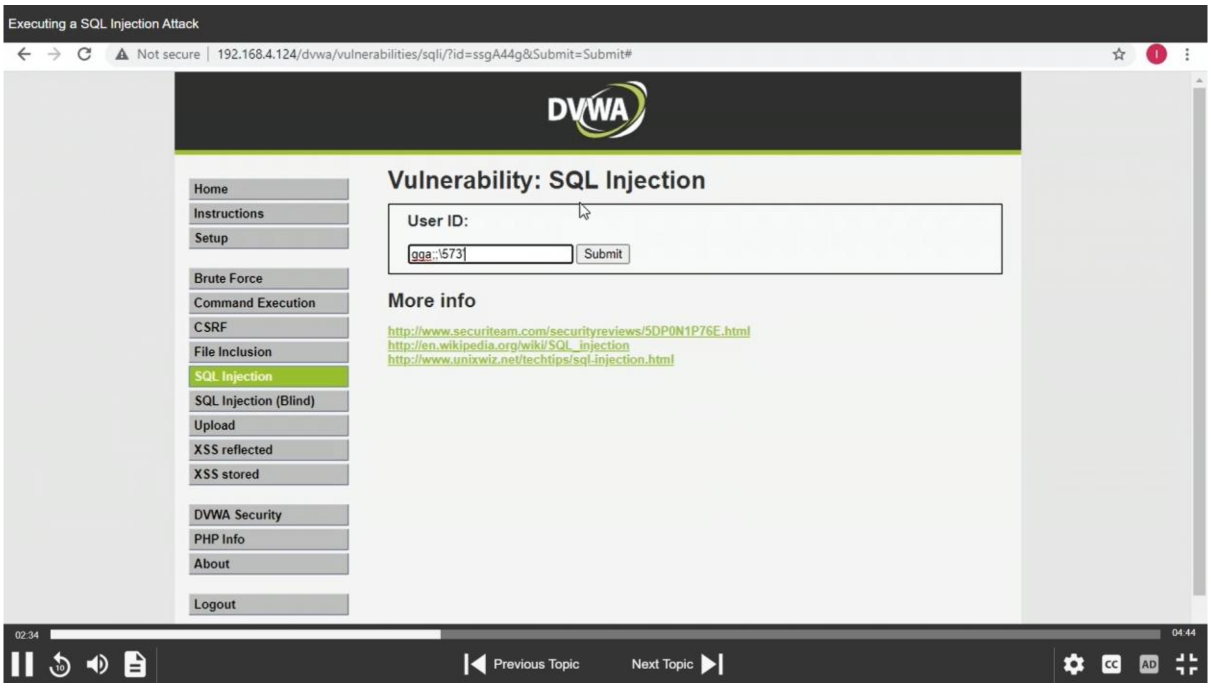

Executing a SQL Injection Attack
Not secure | 192.168.4.124/dvwa/vulnerabilities/sqli/?id=2&Submit=Submit#
DVWA
Home
Instructions
Setup
Brute Force
Command Execution
CSRF
File Inclusion
SQL Injection
SQL Injection (Blind)
Upload
XSS reflected
XSS stored
DVWA Security
PHP Info
About
Logout
Vulnerability: SQL Injection
User ID:
Submit

```
ID: 2
First name: Gordon
Surname: Brown
```

More info
http://www.securiteam.com/securityreviews/5DP0N1P76E.html
http://en.wikipedia.org/wiki/SQL_injection
http://www.unixwiz.net/techtips/sql-injection.html
01:41
05:38
Previous Topic
Next Topic


192.168.4.124/dvwa/vulnerabiliti
Not secure
192.168.4.124/dvwa/vulnerabilities/sqli/?id=gga%3B%3B%5C573%278851&Submit=Submit#

```
You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near '8851'' at line 1
```

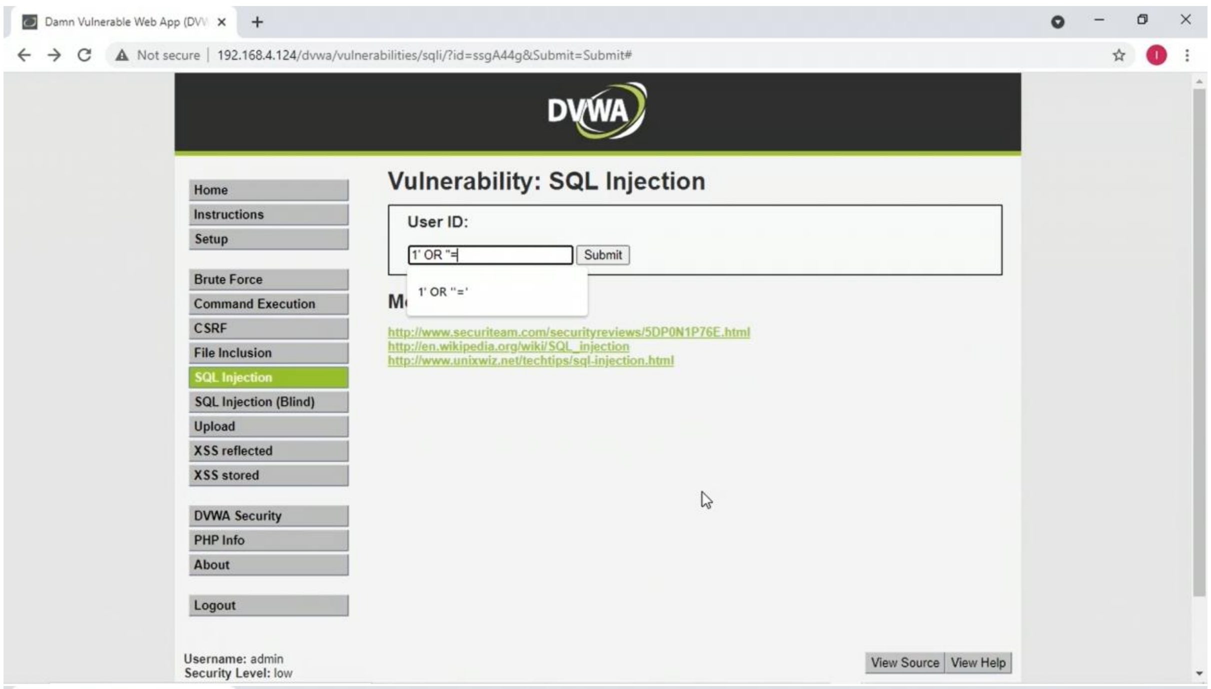


Vulnerability: SQL Injection

User ID: gga;;\573

More info

http://www.securiteam.com/securityreviews/5DP0N1P76E.html
http://en.wikipedia.org/wiki/SQL_injection
http://www.unixwiz.net/techtips/sql-injection.html



Vulnerability: SQL Injection

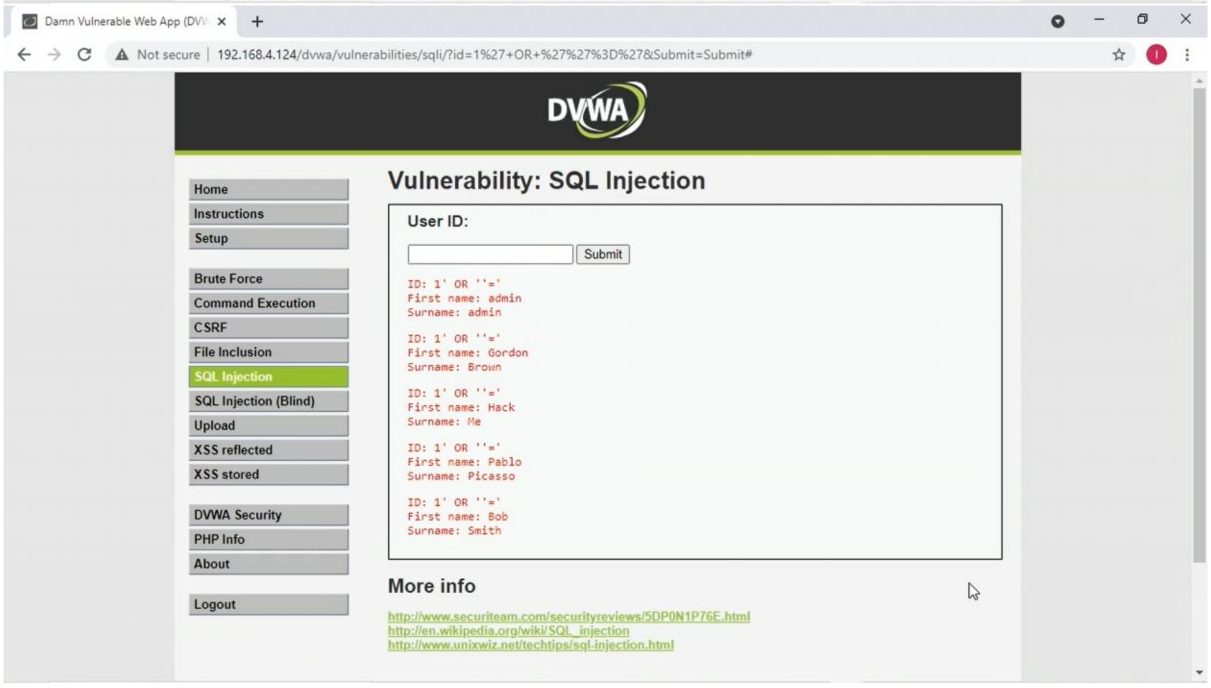
User ID:

1' OR ''=

1' OR ''='

http://www.securiteam.com/securityreviews/5DP0N1P76E.html
http://en.wikipedia.org/wiki/SQL_injection
http://www.unixwiz.net/techtips/sql-injection.html

Username: admin
Security Level: low



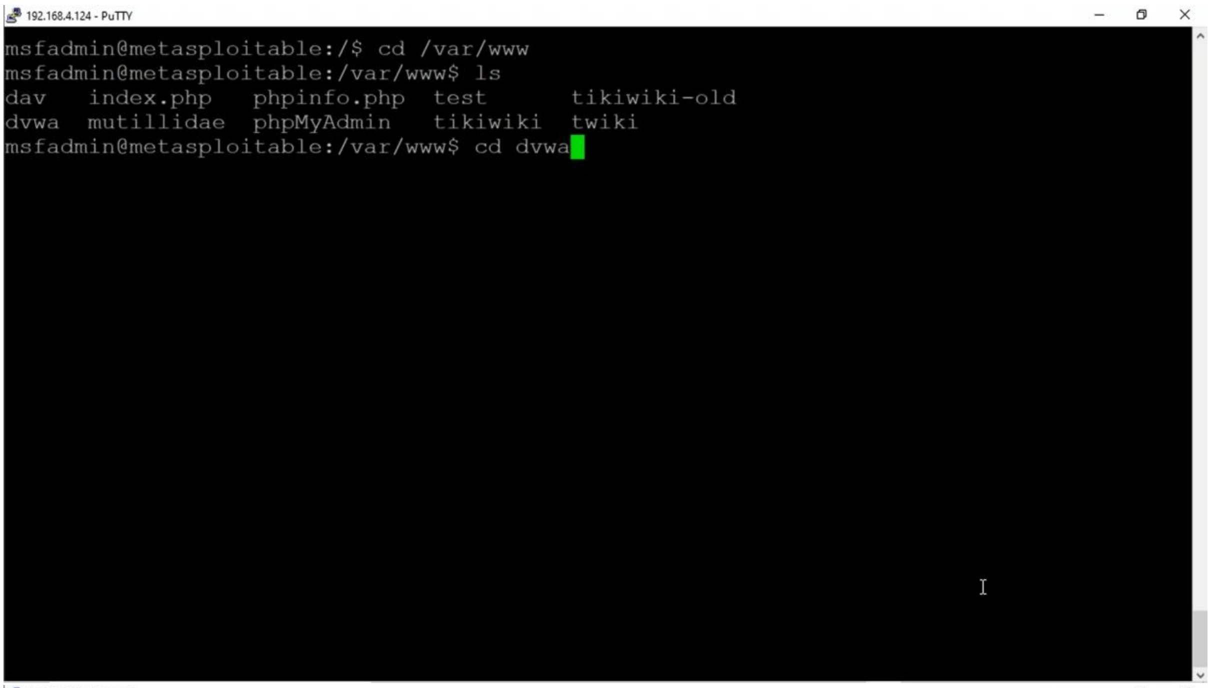
Vulnerability: SQL Injection

User ID:

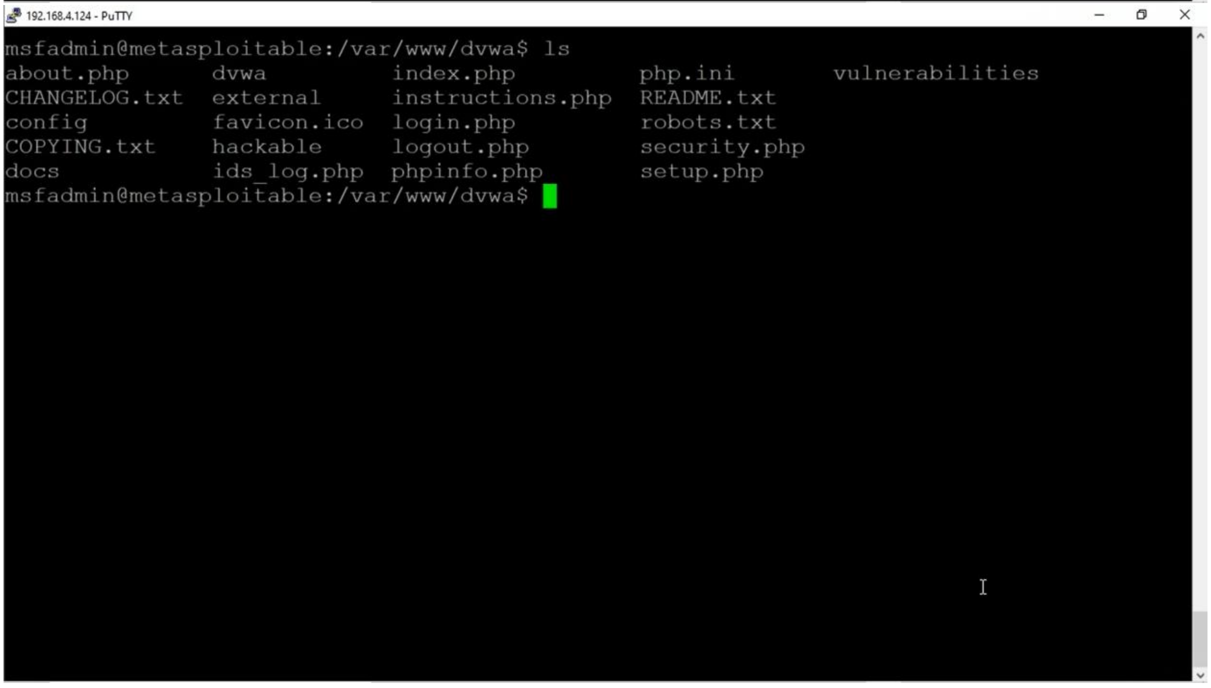
```
ID: 1' OR ''='
First name: admin
Surname: admin

ID: 1' OR ''='
First name: Gordon
Surname: Brown

ID: 1' OR ''='
First name: Hack
Surname: Me

ID: 1' OR ''='
First name: Pablo
Surname: Picasso

ID: 1' OR ''='
First name: Bob
Surname: Smith
```

More info

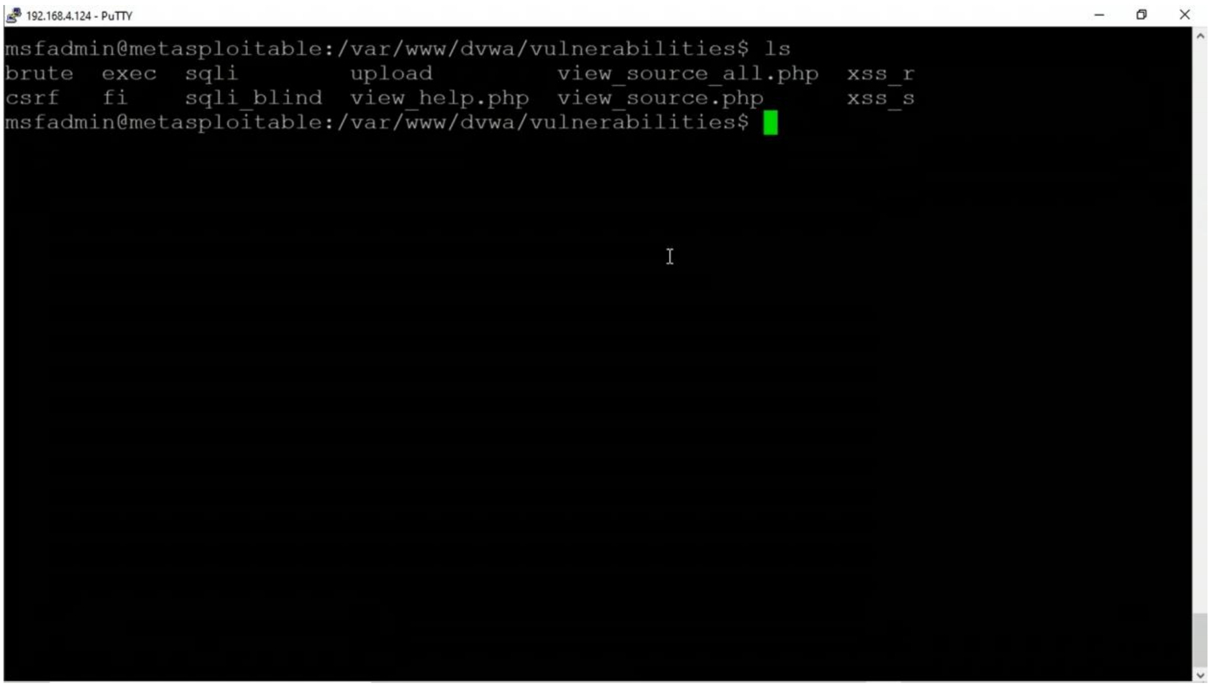
http://www.securiteam.com/securityreviews/5DP0N1P76E.html
http://en.wikipedia.org/wiki/SQL_injection
http://www.unixwiz.net/techtips/sql-injection.html


192.168.4.124 - PuTTY

```
msfadmin@metasploitable:/$ cd /var/www
msfadmin@metasploitable:/var/www$ ls
dav   index.php   phpinfo.php  test      tikiwiki-old
dvwa  mutillidae  phpMyAdmin   tikiwiki  twiki
msfadmin@metasploitable:/var/www$ cd dvwa
```

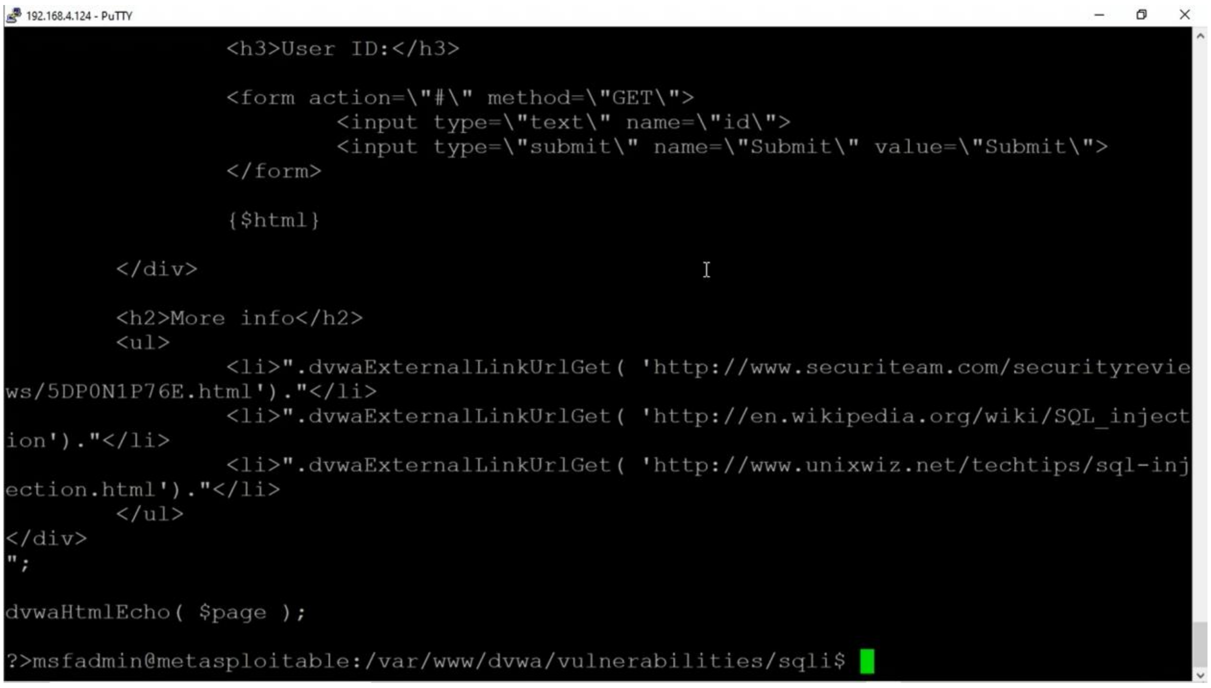

192.168.4.124 - PuTTY

```
msfadmin@metasploitable:/var/www/dvwa$ ls
about.php      dvwa         index.php         php.ini       vulnerabilities
CHANGELOG.txt  external     instructions.php  README.txt
config         favicon.ico  login.php         robots.txt
COPYING.txt    hackable     logout.php        security.php
docs           ids_log.php  phpinfo.php       setup.php
msfadmin@metasploitable:/var/www/dvwa$
```

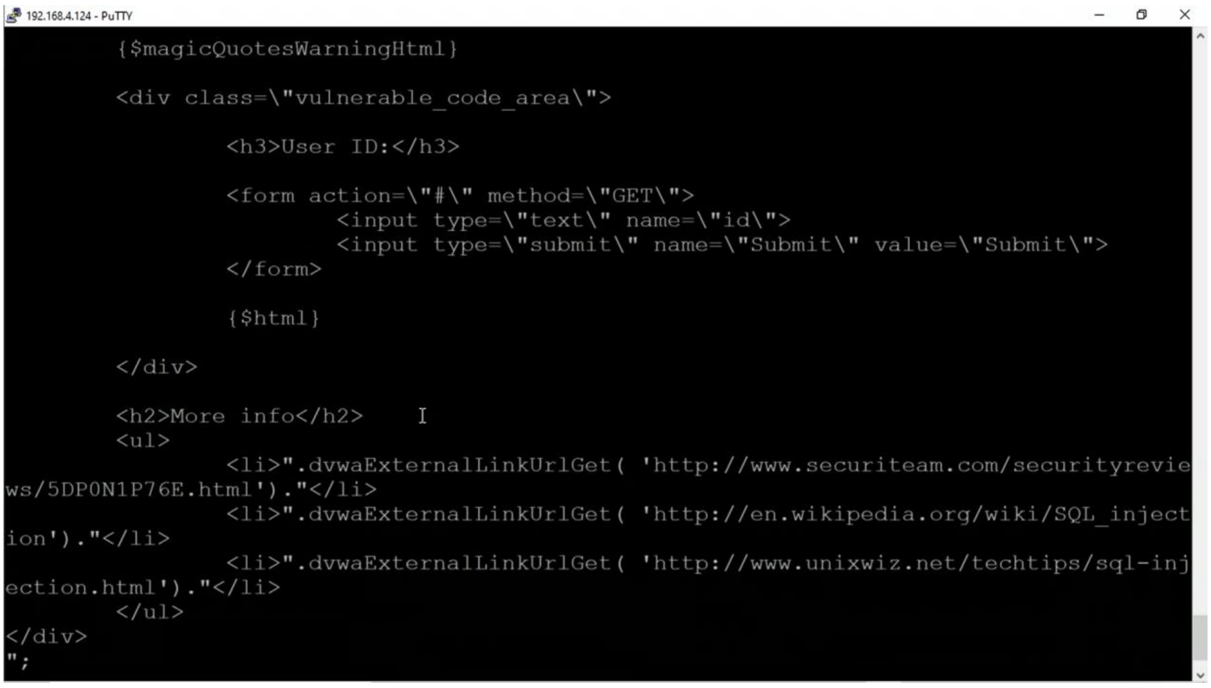


```
msfadmin@metasploitable:/var/www/dvwa/vulnerabilities$ ls
brute  exec  sqli        upload          view_source_all.php  xss_r
csrf   fi    sqli_blind  view_help.php   view_source.php      xss_s
msfadmin@metasploitable:/var/www/dvwa/vulnerabilities$ 
```

Cat index.php

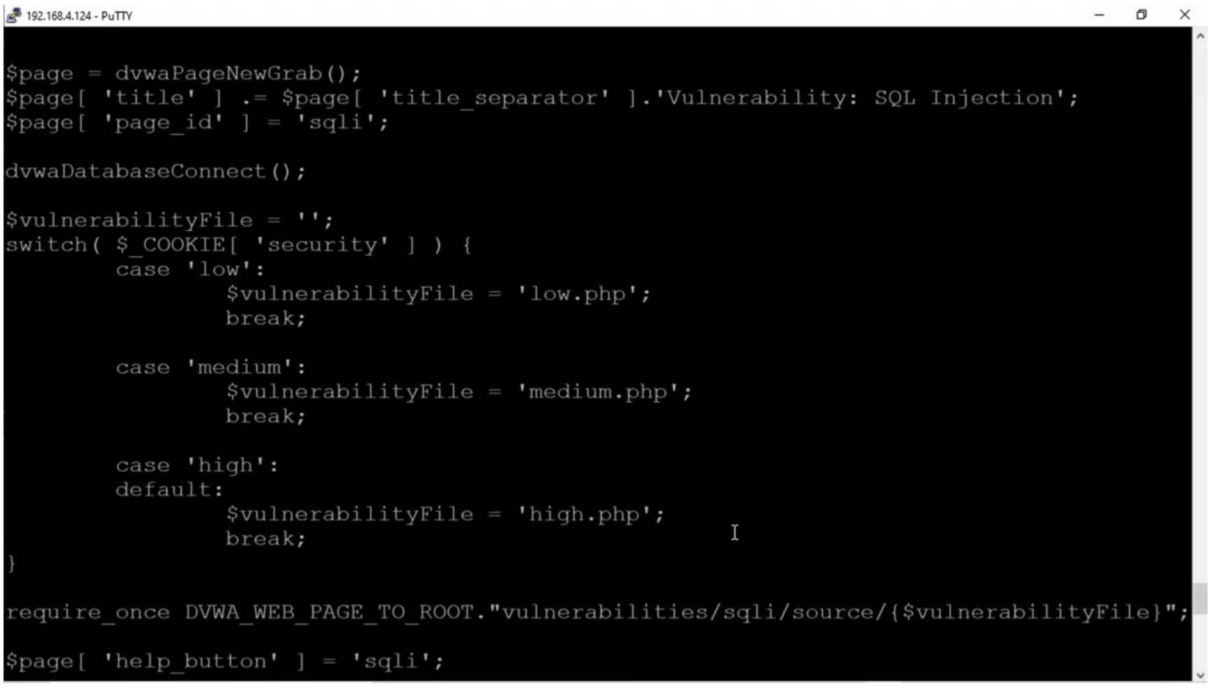


```
                <h3>User ID:</h3>

                <form action=\"#\" method=\"GET\">
                        <input type=\"text\" name=\"id\">
                        <input type=\"submit\" name=\"Submit\" value=\"Submit\">
                </form>

                {$html}

        </div>

        <h2>More info</h2>
        <ul>
                <li>".dvwaExternalLinkUrlGet( 'http://www.securiteam.com/securityrevie
ws/5DP0N1P76E.html')."</li>
                <li>".dvwaExternalLinkUrlGet( 'http://en.wikipedia.org/wiki/SQL_inject
ion')."</li>
                <li>".dvwaExternalLinkUrlGet( 'http://www.unixwiz.net/techtips/sql-inj
ection.html')."</li>
        </ul>
</div>
";

dvwaHtmlEcho( $page );

?>msfadmin@metasploitable:/var/www/dvwa/vulnerabilities/sqli$ 
```



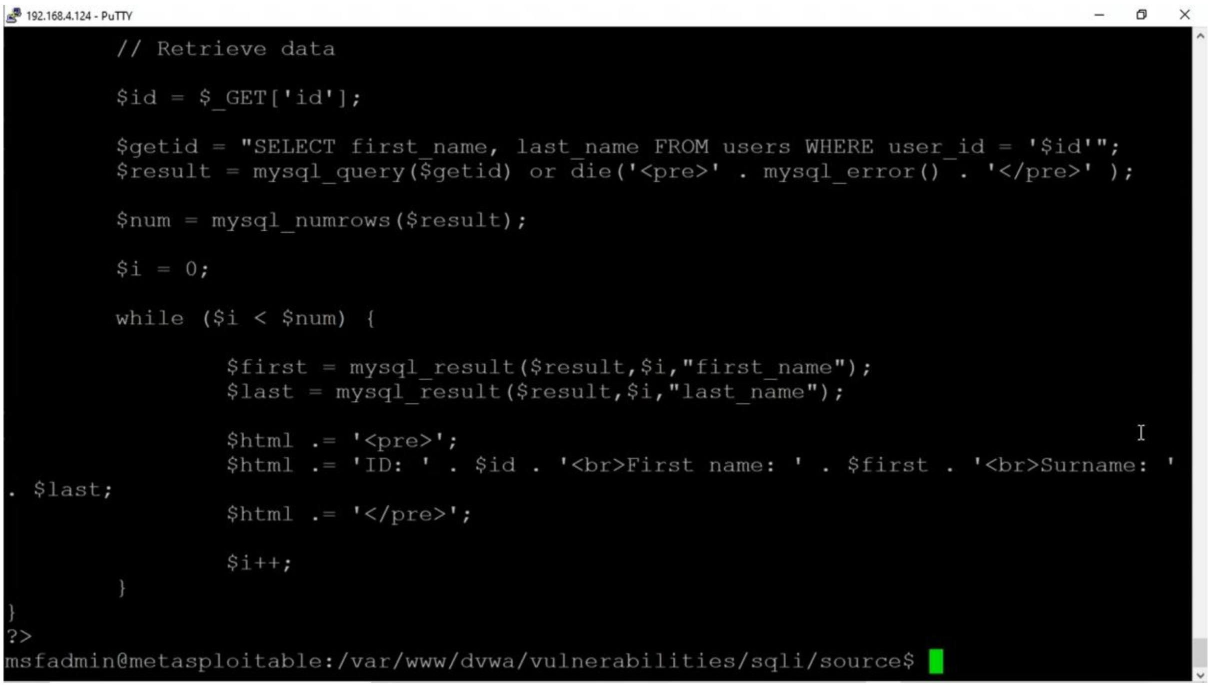
192.168.4.124 - PuTTY

```
        {$magicQuotesWarningHtml}

        <div class=\"vulnerable_code_area\">

                <h3>User ID:</h3>

                <form action=\"#\" method=\"GET\">
                        <input type=\"text\" name=\"id\">
                        <input type=\"submit\" name=\"Submit\" value=\"Submit\">
                </form>

                {$html}

        </div>

        <h2>More info</h2>
        <ul>
                <li>".dvwaExternalLinkUrlGet( 'http://www.securiteam.com/securityrevie
ws/5DP0N1P76E.html')."</li>
                <li>".dvwaExternalLinkUrlGet( 'http://en.wikipedia.org/wiki/SQL_inject
ion')."</li>
                <li>".dvwaExternalLinkUrlGet( 'http://www.unixwiz.net/techtips/sql-inj
ection.html')."</li>
        </ul>
</div>
";
```



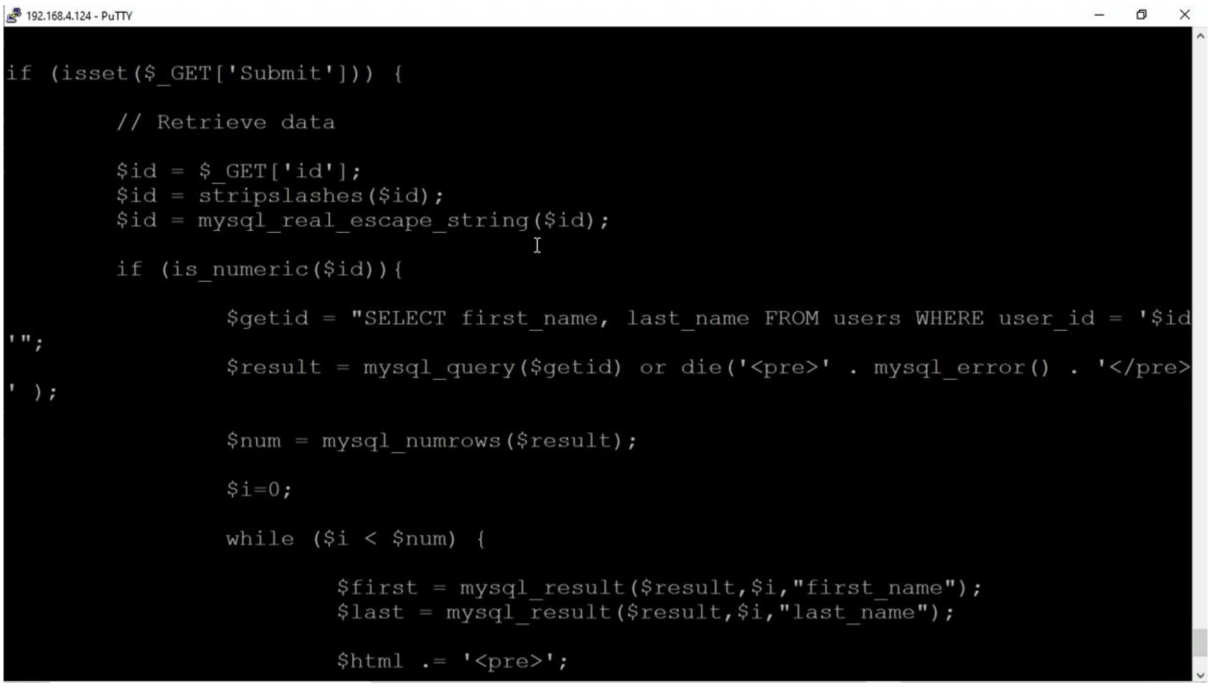
192.168.4.124 - PuTTY

```

$page = dvwaPageNewGrab();
$page[ 'title' ] .= $page[ 'title_separator' ].'Vulnerability: SQL Injection';
$page[ 'page_id' ] = 'sqli';

dvwaDatabaseConnect();

$vulnerabilityFile = '';
switch( $_COOKIE[ 'security' ] ) {
        case 'low':
                $vulnerabilityFile = 'low.php';
                break;

        case 'medium':
                $vulnerabilityFile = 'medium.php';
                break;

        case 'high':
        default:
                $vulnerabilityFile = 'high.php';
                break;
}

require_once DVWA_WEB_PAGE_TO_ROOT."vulnerabilities/sqli/source/{$vulnerabilityFile}";

$page[ 'help_button' ] = 'sqli';
```



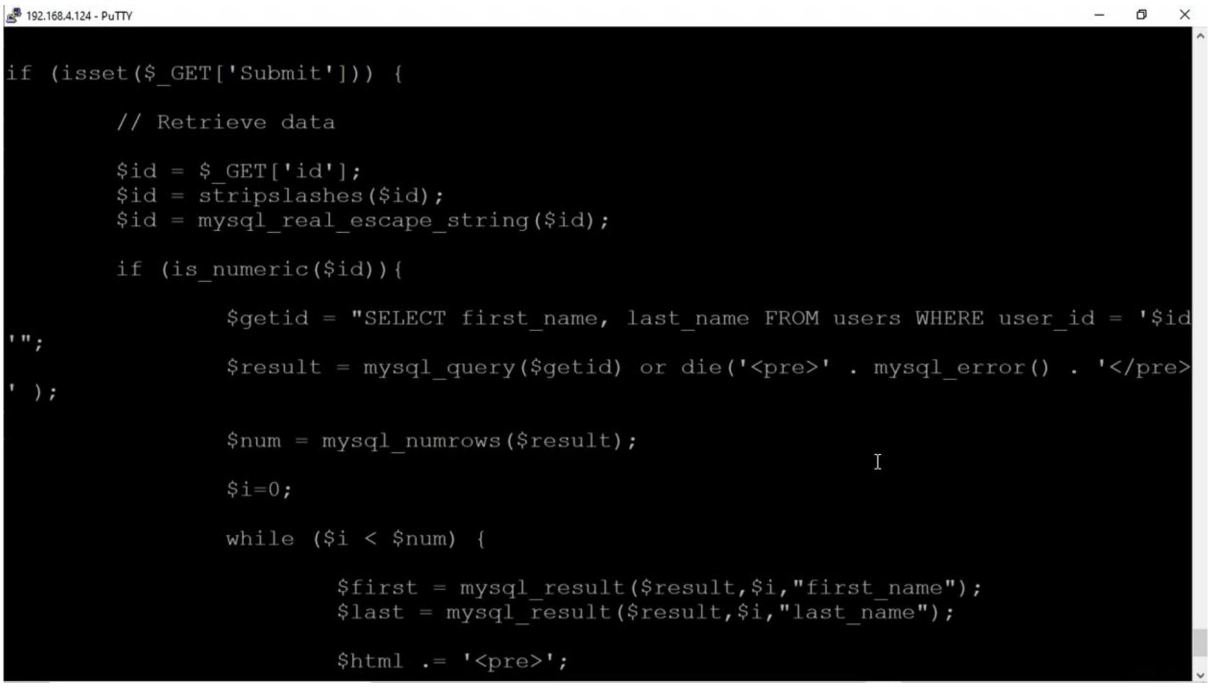
192.168.4.124 - PuTTY

```
        // Retrieve data

        $id = $_GET['id'];

        $getid = "SELECT first_name, last_name FROM users WHERE user_id = '$id'";
        $result = mysql_query($getid) or die('<pre>' . mysql_error() . '</pre>' );

        $num = mysql_numrows($result);

        $i = 0;

        while ($i < $num) {

                $first = mysql_result($result,$i,"first_name");
                $last = mysql_result($result,$i,"last_name");

                $html .= '<pre>';
                $html .= 'ID: ' . $id . '<br>First name: ' . $first . '<br>Surname: '
. $last;
                $html .= '</pre>';

                $i++;
        }
}
?>
msfadmin@metasploitable:/var/www/dvwa/vulnerabilities/sqli/source$
```



192.168.4.124 - PuTTY

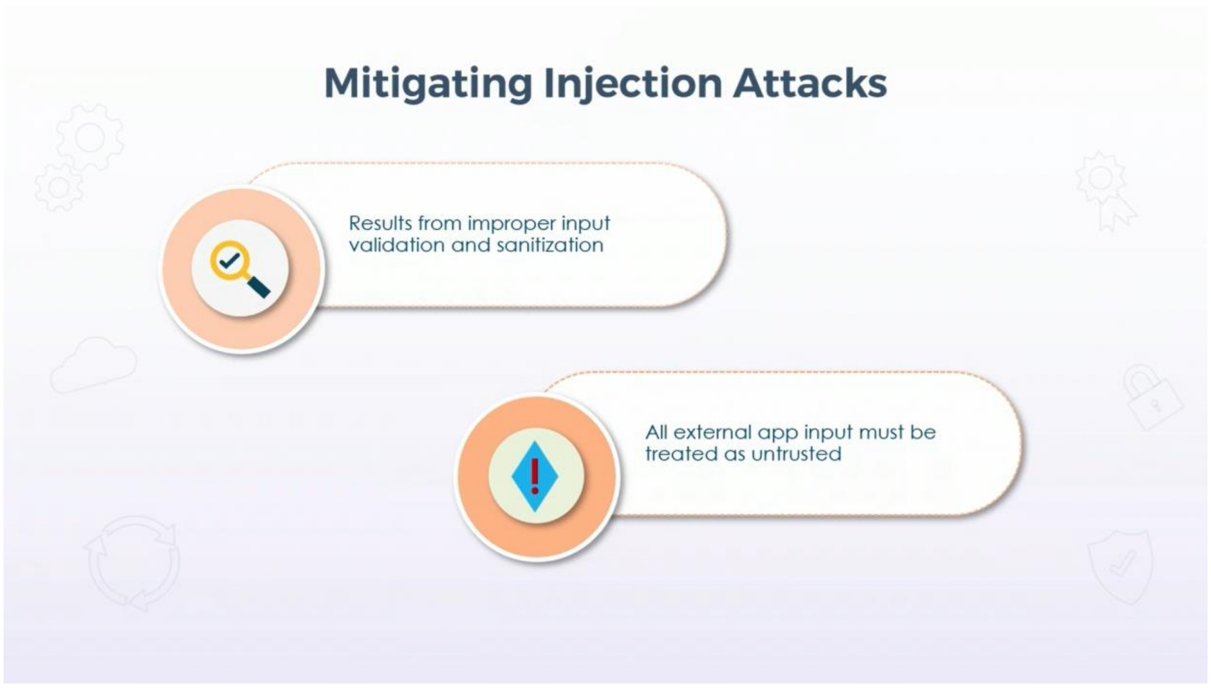
```
if (isset($_GET['Submit'])) {

        // Retrieve data

        $id = $_GET['id'];
        $id = stripslashes($id);
        $id = mysql_real_escape_string($id);

        if (is_numeric($id)){

                $getid = "SELECT first_name, last_name FROM users WHERE user_id = '$id
'";
                $result = mysql_query($getid) or die('<pre>' . mysql_error() . '</pre>
' );

                $num = mysql_numrows($result);

                $i=0;

                while ($i < $num) {

                        $first = mysql_result($result,$i,"first_name");
                        $last = mysql_result($result,$i,"last_name");

                        $html .= '<pre>';
```



```
if (isset($_GET['Submit'])) {

        // Retrieve data

        $id = $_GET['id'];
        $id = stripslashes($id);
        $id = mysql_real_escape_string($id);

        if (is_numeric($id)){

                $getid = "SELECT first_name, last_name FROM users WHERE user_id = '$id
'";
                $result = mysql_query($getid) or die('<pre>' . mysql_error() . '</pre>
' );

                $num = mysql_numrows($result);

                $i=0;

                while ($i < $num) {

                        $first = mysql_result($result,$i,"first_name");
                        $last = mysql_result($result,$i,"last_name");

                        $html .= '<pre>';
```

## Mitigating Injection Attacks

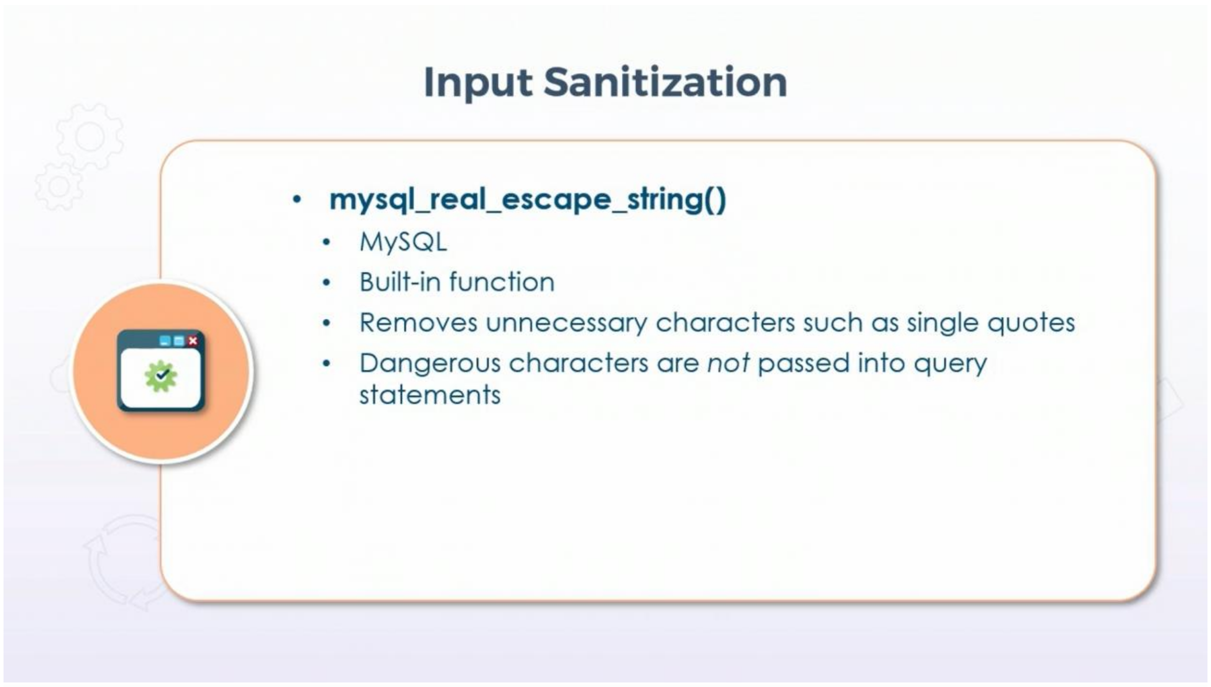
Trust no one!
(Zero Trust)



## Mitigating Injection Attacks

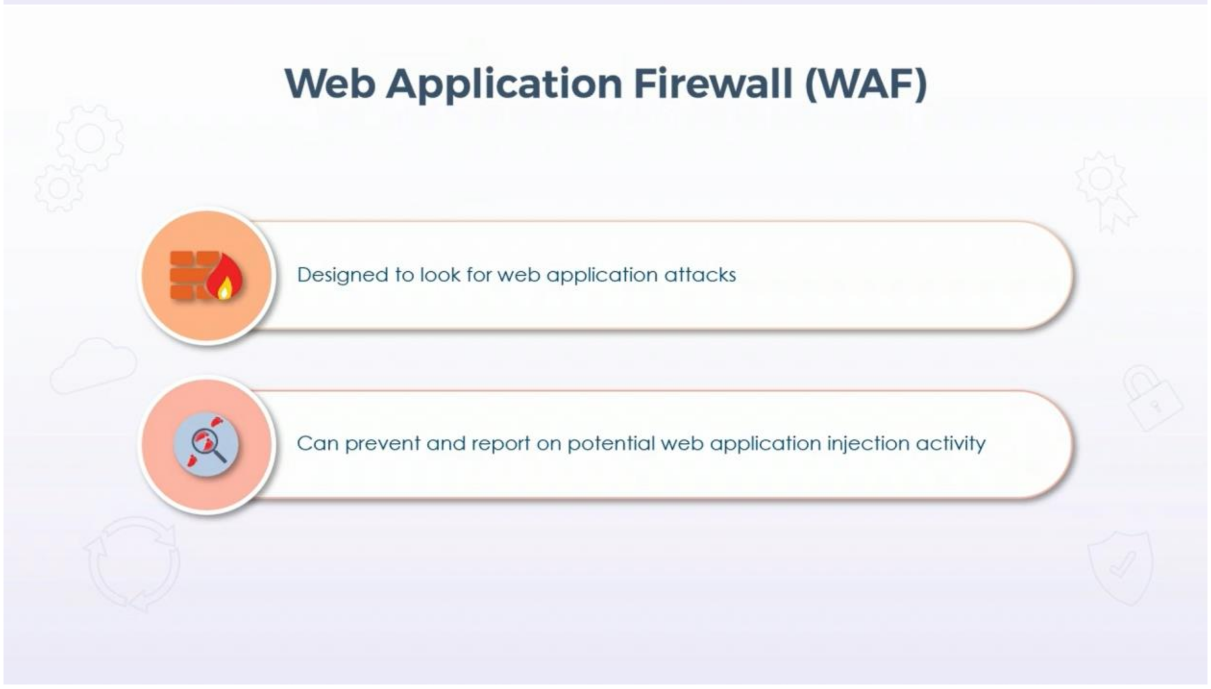
Results from improper input validation and sanitization

All external app input must be treated as untrusted



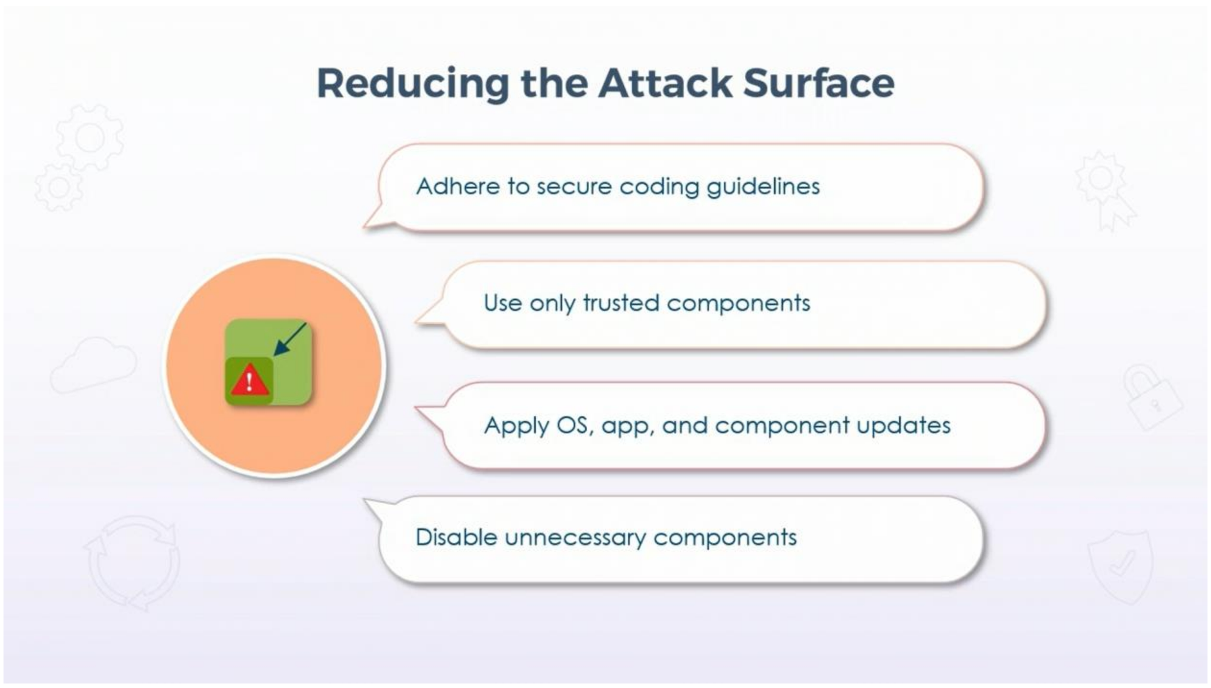
## Input Sanitization

- **mysql_real_escape_string()**
  - MySQL
  - Built-in function
  - Removes unnecessary characters such as single quotes
  - Dangerous characters are *not* passed into query statements



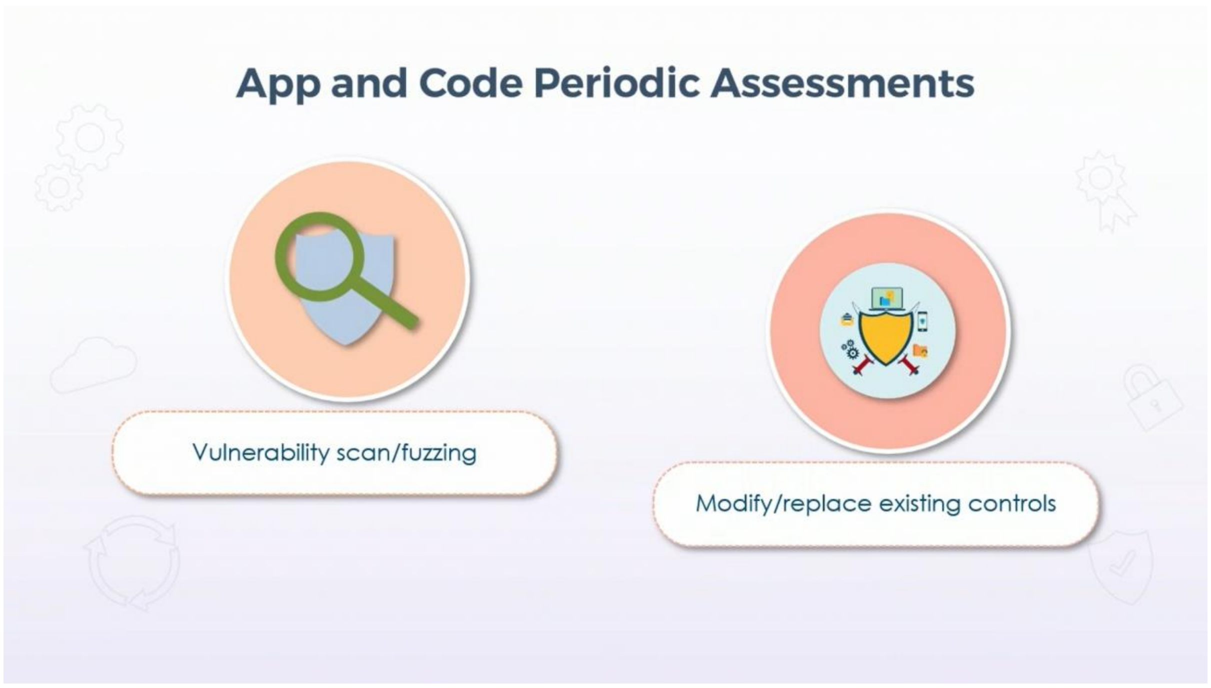
## Web Application Firewall (WAF)

- Designed to look for web application attacks
- Can prevent and report on potential web application injection activity



## Reducing the Attack Surface

- Adhere to secure coding guidelines
- Use only trusted components
- Apply OS, app, and component updates
- Disable unnecessary components



## App and Code Periodic Assessments

- Vulnerability scan/fuzzing
- Modify/replace existing controls