DVWA

Julian Barfus

4aWI

Table of Contents

[Brute Force 4](#_Toc194052745)

[1.1 Definition: 4](#_Toc194052746)

[1.2 Beliebte Brute Force Tools: 4](#_Toc194052747)

[1.3 Code 4](#_Toc194052748)

[Unsicherer Code 4](#_Toc194052749)

[Sicherer Code 5](#_Toc194052750)

[1.4 Grafik 6](#_Toc194052751)

[Brute Force Tools 7](#_Toc194052752)

[Aircrack-ng 7](#_Toc194052753)

[John the ripper 8](#_Toc194052754)

[Hydra 9](#_Toc194052755)

[Burpsuite 10](#_Toc194052756)

[Ncrack 11](#_Toc194052757)

[Command Injection 12](#_Toc194052758)

[2.1 Definition: 12](#_Toc194052759)

[2.2 Code 12](#_Toc194052760)

[Unsicherer Code 12](#_Toc194052761)

[Sicherer Code 13](#_Toc194052762)

[2.3 Grafik 14](#_Toc194052763)

[2.4 Commands 14](#_Toc194052764)

[CSRF 16](#_Toc194052765)

[3.1 Definition: 16](#_Toc194052766)

[3.2 Code 16](#_Toc194052767)

[Unsicherer Code 16](#_Toc194052768)

[Sicherer Code 17](#_Toc194052769)

[3.3 Grafik 18](#_Toc194052770)

[File Inclusion 19](#_Toc194052771)

[4.1 Definition: 19](#_Toc194052772)

[4.2 Code 19](#_Toc194052773)

[Unsicherer Code 19](#_Toc194052774)

[Sicherer Code 19](#_Toc194052775)

[4.3 Grafik 20](#_Toc194052776)

[File Upload 21](#_Toc194052777)

[5.1 Definition: 21](#_Toc194052778)

[5.2 Code 21](#_Toc194052779)

[Unsicherer Code 21](#_Toc194052780)

[Sicherer Code 22](#_Toc194052781)

[5.3 Grafik 23](#_Toc194052782)

[Insecure CAPTCHA 24](#_Toc194052783)

[6.1 Definition: 24](#_Toc194052784)

[6.2 Code 24](#_Toc194052785)

[Unsicherer Code 24](#_Toc194052786)

[Sicherer Code 25](#_Toc194052787)

[6.3 Grafik 26](#_Toc194052788)

[SQL-Injektion 27](#_Toc194052789)

[7.1 Definition: 27](#_Toc194052790)

[7.2 Code 27](#_Toc194052791)

[Unsicherer Code 27](#_Toc194052792)

[Sicherer Code 28](#_Toc194052793)

[7.3 Grafik 29](#_Toc194052794)

# Brute Force

## 1.1 Definition:

Ein Brute Force Angriff (Brute Force Attack auf Englisch) ist eine Methode, die Cyberkriminelle anwenden, um Passwörter und andere Zugangsdaten zu knacken. Bei einem Brute Force Angriff greift ein Angreifer auf eine Liste an häufigen Wörtern zurück und probiert sie der Reihe nach durch, bis eines funktioniert. Wenn diese Liste zu keinem Ergebnis führt, versucht er es mit verschiedenen Buchstabenkombinationen. Es braucht manchmal tausende von Versuchen, bis ein Passwort geknackt ist. Deshalb kommen für Brute Force, Automatisierungstools zum Einsatz, die es einem Angreifer ermöglichen, sehr viele Versuche möglichst schnell durchzuführen.

[Literaturhinweis](https://www.proofpoint.com/de/threat-reference/brute-force-attack#:~:text=Ein%20Brute%20Force%20Angriff%20(Brute,nach%20durch%2C%20bis%20eines%20funktioniert.) 13.01.2025

## 1.2 Beliebte Brute Force Tools:

* Aircrack-ng
* John the Ripper
* Klassische Programierung
* Hydra
* Burpsuite
* Ncrack

[Literaturhinweis](https://www.proofpoint.com/de/threat-reference/brute-force-attack#:~:text=Ein%20Brute%20Force%20Angriff%20(Brute,nach%20durch%2C%20bis%20eines%20funktioniert.) 13.01.2025

## 1.3 Code

### Unsicherer Code

<?php

if( isset( $\_GET[ 'Login' ] ) ) {

// Get username

$user = $\_GET[ 'username' ];

// Get password

$pass = $\_GET[ 'password' ];

$pass = md5( $pass );

// Check the database

$query = "SELECT \* FROM `users` WHERE user = '$user' AND password = '$pass';";

$result = mysqli\_query($GLOBALS["\_\_\_mysqli\_ston"], $query ) or die( '<pre>' . ((is\_object($GLOBALS["\_\_\_mysqli\_ston"])) ? mysqli\_error($GLOBALS["\_\_\_mysqli\_ston"]) : (($\_\_\_mysqli\_res = mysqli\_connect\_error()) ? $\_\_\_mysqli\_res : false)) . '</pre>' );

if( $result && mysqli\_num\_rows( $result ) == 1 ) {

// Get users details

$row = mysqli\_fetch\_assoc( $result );

$avatar = $row["avatar"];

// Login successful

echo "<p>Welcome to the password protected area {$user}</p>";

echo "<img src=\"{$avatar}\" />";

}

else {

// Login failed

echo "<pre><br />Username and/or password incorrect.</pre>";

}

((is\_null($\_\_\_mysqli\_res = mysqli\_close($GLOBALS["\_\_\_mysqli\_ston"]))) ? false : $\_\_\_mysqli\_res);

}

?>

#### Warum ist dieser Code unsicher

Mann kann so viele passwort versuche machen wie man möchte

### Sicherer Code

<?php

if( isset( $\_POST[ 'Login' ] ) && isset ($\_POST['username']) && isset ($\_POST['password']) ) {

// Check Anti-CSRF token

checkToken( $\_REQUEST[ 'user\_token' ], $\_SESSION[ 'session\_token' ], 'index.php' );

// Sanitise username input

$user = $\_POST[ 'username' ];

$user = stripslashes( $user );

$user = ((isset($GLOBALS["\_\_\_mysqli\_ston"]) && is\_object($GLOBALS["\_\_\_mysqli\_ston"])) ? mysqli\_real\_escape\_string($GLOBALS["\_\_\_mysqli\_ston"], $user ) : ((trigger\_error("[MySQLConverterToo] Fix the mysql\_escape\_string() call! This code does not work.", E\_USER\_ERROR)) ? "" : ""));

// Sanitise password input

$pass = $\_POST[ 'password' ];

$pass = stripslashes( $pass );

$pass = ((isset($GLOBALS["\_\_\_mysqli\_ston"]) && is\_object($GLOBALS["\_\_\_mysqli\_ston"])) ? mysqli\_real\_escape\_string($GLOBALS["\_\_\_mysqli\_ston"], $pass ) : ((trigger\_error("[MySQLConverterToo] Fix the mysql\_escape\_string() call! This code does not work.", E\_USER\_ERROR)) ? "" : ""));

$pass = md5( $pass );

// Default values

$total\_failed\_login = 3;

$lockout\_time = 15;

$account\_locked = false;

// Check the database (Check user information)

$data = $db->prepare( 'SELECT failed\_login, last\_login FROM users WHERE user = (:user) LIMIT 1;' );

$data->bindParam( ':user', $user, PDO::PARAM\_STR );

$data->execute();

$row = $data->fetch();

// Check to see if the user has been locked out.

if( ( $data->rowCount() == 1 ) && ( $row[ 'failed\_login' ] >= $total\_failed\_login ) ) {

// User locked out. Note, using this method would allow for user enumeration!

//echo "<pre><br />This account has been locked due to too many incorrect logins.</pre>";

// Calculate when the user would be allowed to login again

$last\_login = strtotime( $row[ 'last\_login' ] );

$timeout = $last\_login + ($lockout\_time \* 60);

$timenow = time();

/\*

print "The last login was: " . date ("h:i:s", $last\_login) . "<br />";

print "The timenow is: " . date ("h:i:s", $timenow) . "<br />";

print "The timeout is: " . date ("h:i:s", $timeout) . "<br />";

\*/

// Check to see if enough time has passed, if it hasn't locked the account

if( $timenow < $timeout ) {

$account\_locked = true;

// print "The account is locked<br />";

}

}

// Check the database (if username matches the password)

$data = $db->prepare( 'SELECT \* FROM users WHERE user = (:user) AND password = (:password) LIMIT 1;' );

$data->bindParam( ':user', $user, PDO::PARAM\_STR);

$data->bindParam( ':password', $pass, PDO::PARAM\_STR );

$data->execute();

$row = $data->fetch();

// If its a valid login...

if( ( $data->rowCount() == 1 ) && ( $account\_locked == false ) ) {

// Get users details

$avatar = $row[ 'avatar' ];

$failed\_login = $row[ 'failed\_login' ];

$last\_login = $row[ 'last\_login' ];

// Login successful

echo "<p>Welcome to the password protected area <em>{$user}</em></p>";

echo "<img src=\"{$avatar}\" />";

// Had the account been locked out since last login?

if( $failed\_login >= $total\_failed\_login ) {

echo "<p><em>Warning</em>: Someone might of been brute forcing your account.</p>";

echo "<p>Number of login attempts: <em>{$failed\_login}</em>.<br />Last login attempt was at: <em>{$last\_login}</em>.</p>";

}

// Reset bad login count

$data = $db->prepare( 'UPDATE users SET failed\_login = "0" WHERE user = (:user) LIMIT 1;' );

$data->bindParam( ':user', $user, PDO::PARAM\_STR );

$data->execute();

} else {

// Login failed

sleep( rand( 2, 4 ) );

// Give the user some feedback

echo "<pre><br />Username and/or password incorrect.<br /><br/>Alternative, the account has been locked because of too many failed logins.<br />If this is the case, <em>please try again in {$lockout\_time} minutes</em>.</pre>";

// Update bad login count

$data = $db->prepare( 'UPDATE users SET failed\_login = (failed\_login + 1) WHERE user = (:user) LIMIT 1;' );

$data->bindParam( ':user', $user, PDO::PARAM\_STR );

$data->execute();

}

// Set the last login time

$data = $db->prepare( 'UPDATE users SET last\_login = now() WHERE user = (:user) LIMIT 1;' );

$data->bindParam( ':user', $user, PDO::PARAM\_STR );

$data->execute();

}

// Generate Anti-CSRF token

generateSessionToken();

?>

#### Entscheidende Punkte für sicheren Code sind

* Check Loginversuche!!
* CSRF-Token
* Verhinderung über Zeiteinstieg!!
* 2FA
* Check auf mensch als Bediner(Captures!!!)

## 1.4 Grafik



# Brute Force Tools

## Aircrack-ng

**Beschreibung:**

* Ein Paket von Tools zum Testen der Sicherheit von WLAN-Netzwerken.
* Hauptsächlich verwendet für:
  + Überwachung von Netzwerken.
  + Angriff auf WLAN-Verschlüsselungen wie WEP und WPA/WPA2-PSK.
  + Capturing und Cracking von Handshakes.

**Hauptfunktionen:**

* **Airodump-ng**: Netzwerk-Sniffing und Paketerfassung.
* **Aircrack-ng**: Passwort-Cracking basierend auf den erfassten Daten.
* **Aireplay-ng**: Packet-Injection für Replay-Angriffe.
* **Airdecap-ng**: Entschlüsselung von erfassten WLAN-Paketen.

**Unterstützte Betriebssysteme:**

* **Linux**: Native Unterstützung, z. B. in Kali Linux.
* **Windows**: Unterstützt, aber eingeschränkte Funktionalität.
* **macOS**: Unterstützt über Homebrew.
* **BSD-Systeme**: Teilweise Unterstützung.

## John the ripper



**Beschreibung:**

* Ein Passwort-Cracking-Tool zur Überprüfung der Passwortsicherheit.
* Kann verschiedene Hash-Typen (z. B. MD5, SHA-1, bcrypt) knacken.
* Unterstützt Wörterbuch-Angriffe, Brute-Force und hybride Angriffe.

**Hauptfunktionen:**

* Flexible Konfiguration für verschiedene Hash-Typen.
* Unterstützung von Multi-Core-Prozessoren und GPU-basiertem Cracking.
* Erweiterbar durch Plugins und benutzerdefinierte Regeln.

**Unterstützte Betriebssysteme:**

* **Linux**: Native Unterstützung.
* **Windows**: Unterstützt (über Cygwin oder vorkompilierte Versionen).
* **macOS**: Unterstützt.
* **UNIX und BSD**: Unterstützt.

## Hydra



**Beschreibung:**

* Ein Passwort-Cracking-Tool für Online-Dienste und Protokolle.
* Unterstützt zahlreiche Protokolle wie HTTP, FTP, SSH, RDP, MySQL, etc.

**Hauptfunktionen:**

* Sehr schnell und parallelisiert.
* Unterstützt Brute-Force- und Wörterbuch-Angriffe.
* Flexibel und erweiterbar durch benutzerdefinierte Module.

**Unterstützte Betriebssysteme:**

* **Linux**: Native Unterstützung (z. B. in Kali Linux).
* **Windows**: Unterstützt über Cygwin.
* **macOS**: Unterstützt über Homebrew.
* **UNIX-Systeme**: Teilweise unterstützt.

## Burpsuite



**Beschreibung:**

* Ein umfassendes Tool für Penetrationstests von Webanwendungen.
* Ermöglicht das Identifizieren von Sicherheitslücken wie XSS, SQL-Injection, CSRF und mehr.

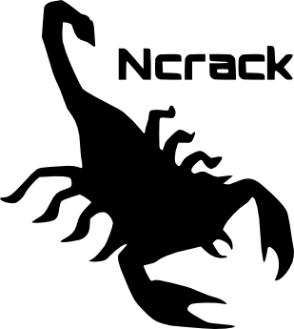
**Hauptfunktionen:**

* **Proxy**: Interceptieren und Modifizieren von HTTP/S-Anfragen.
* **Scanner**: Automatisches Scannen nach Schwachstellen.
* **Intruder**: Automatisierte Angriffe auf Parameter und Formulare.
* **Repeater**: Manuelles Testen von HTTP/S-Anfragen.

**Unterstützte Betriebssysteme:**

* **Linux**: Unterstützt.
* **Windows**: Unterstützt.
* **macOS**: Unterstützt.
* **Java-basiert**: Funktioniert auf jedem Betriebssystem mit einer Java Runtime Environment (JRE).

## Ncrack



**Beschreibung:**

* Ein Tool zum Cracking von Netzwerk-Authentifizierung.
* Speziell entwickelt, um hochsichere Passwörter in Protokollen wie SSH, RDP, FTP, Telnet, und HTTP zu knacken.

**Hauptfunktionen:**

* Schnelle und parallele Authentifizierungsprüfungen.
* Unterstützung für passwortgeschützte Dienste (z. B. VNC, POP3).
* Erweiterbar durch benutzerdefinierte Module.

**Unterstützte Betriebssysteme:**

* **Linux**: Native Unterstützung.
* **Windows**: Unterstützt.
* **macOS**: Unterstützt.
* **BSD-Systeme**: Unterstützt.

# Command Injection

## 2.1 Definition:

OS Command Injection ist eine Schwachstelle bei Webanwendungen, die es Angreifern ermöglicht, beliebige Befehle auf dem zugrunde liegenden Betriebssystem auszuführen. Solche Schwachstellen treten auf, wenn Webanwendungen Betriebssystembefehle mit vom Nutzer bereitgestellten Eingaben als Argumente aufrufen.

## 2.2 Code

### Unsicherer Code

<?php

if( isset( $\_POST[ 'Submit' ] ) ) {

// Get input

$target = $\_REQUEST[ 'ip' ];

// Determine OS and execute the ping command.

if( stristr( php\_uname( 's' ), 'Windows NT' ) ) {

// Windows

$cmd = shell\_exec( 'ping ' . $target );

}

else {

// \*nix

$cmd = shell\_exec( 'ping -c 4 ' . $target );

}

// Feedback for the end user

echo "<pre>{$cmd}</pre>";

}

?>

### Sicherer Code

<?php

if( isset( $\_POST[ 'Submit' ] ) ) {

// Check Anti-CSRF token

checkToken( $\_REQUEST[ 'user\_token' ], $\_SESSION[ 'session\_token' ], 'index.php' );

// Get input

$target = $\_REQUEST[ 'ip' ];

$target = stripslashes( $target );

// Split the IP into 4 octects

$octet = explode( ".", $target );

// Check IF each octet is an integer

if( ( is\_numeric( $octet[0] ) ) && ( is\_numeric( $octet[1] ) ) && ( is\_numeric( $octet[2] ) ) && ( is\_numeric( $octet[3] ) ) && ( sizeof( $octet ) == 4 ) ) {

// If all 4 octets are int's put the IP back together.

$target = $octet[0] . '.' . $octet[1] . '.' . $octet[2] . '.' . $octet[3];

// Determine OS and execute the ping command.

if( stristr( php\_uname( 's' ), 'Windows NT' ) ) {

// Windows

$cmd = shell\_exec( 'ping ' . $target );

}

else {

// \*nix

$cmd = shell\_exec( 'ping -c 4 ' . $target );

}

// Feedback for the end user

echo "<pre>{$cmd}</pre>";

}

else {

// Ops. Let the user name theres a mistake

echo '<pre>ERROR: You have entered an invalid IP.</pre>';

}

}

// Generate Anti-CSRF token

generateSessionToken();

?>

## 2.3 Grafik

A diagram of a computer program

AI-generated content may be incorrect.

## 2.4 Commands

Man kann mehrere Commands in ein feld hinein schreiben.

In der Kommandozeile kannst du verschiedene Trenner verwenden, um Befehle auszuführen oder zu verketten. Hier sind die wichtigsten:

**1. Semikolon (;)**

Ermöglicht das Ausführen mehrerer Befehle nacheinander, unabhängig vom Erfolg des vorherigen Befehls.

echo "Hallo"; echo "Welt"

**2. Und (&&)**

Führt den zweiten Befehl nur aus, wenn der erste erfolgreich war (Exit-Code 0).

mkdir test && cd test

**3. Oder (||)**

Führt den zweiten Befehl nur aus, wenn der erste fehlschlägt (Exit-Code ≠ 0).

mkdir test || echo "Ordner konnte nicht erstellt werden"

**4. Pipe (|)**

Leitet die Ausgabe des ersten Befehls als Eingabe an den zweiten weiter.

ls -l | grep ".txt"

5. Hintergrund-Ausführung (&)

Führt den Befehl im Hintergrund aus, sodass du die Konsole weiter nutzen kannst.

sleep 10 &

**6. Klammern (()) und {}**

() führt Befehle in einer Subshell aus.

(cd /tmp && ls)

{} führt mehrere Befehle in der aktuellen Shell aus (Leerzeichen vor } nötig).

{ echo "Erster"; echo "Zweiter"; }

**7. Doppelte Pipe (|&)**

Leitet sowohl stdout als auch stderr in den nächsten Befehl weiter.

ls /nicht\_existent |& grep "No such file"

A screenshot of a computer

AI-generated content may be incorrect.

# CSRF

## 3.1 Definition:

Eine Cross-Site-Request-Forgery (meist CSRF oder XSRF abgekürzt, deutsch etwa „Webseitenübergreifende Anfragenfälschung“, auch Session-Riding genannt) ist ein Angriff auf ein Computersystem, bei dem der Angreifer eine Transaktion in einer Webanwendung durchführt. Dies geschieht nicht direkt, sondern der Angreifer bedient sich dazu eines Opfers, das bei einer Webanwendung bereits angemeldet sein muss. Dem Webbrowser des Opfers wird ohne dessen Wissen eine HTTP-Anfrage untergeschoben. Der Angreifer wählt die Anfrage so, dass bei deren Aufruf die Webanwendung die vom Angreifer gewünschte Aktion ausführt.

## 3.2 Code

### Unsicherer Code

<?php

if( isset( $\_GET[ 'Change' ] ) ) {

// Get input

$pass\_new = $\_GET[ 'password\_new' ];

$pass\_conf = $\_GET[ 'password\_conf' ];

// Do the passwords match?

if( $pass\_new == $pass\_conf ) {

// They do!

$pass\_new = ((isset($GLOBALS["\_\_\_mysqli\_ston"]) && is\_object($GLOBALS["\_\_\_mysqli\_ston"])) ? mysqli\_real\_escape\_string($GLOBALS["\_\_\_mysqli\_ston"], $pass\_new ) : ((trigger\_error("[MySQLConverterToo] Fix the mysql\_escape\_string() call! This code does not work.", E\_USER\_ERROR)) ? "" : ""));

$pass\_new = md5( $pass\_new );

// Update the database

$current\_user = dvwaCurrentUser();

$insert = "UPDATE `users` SET password = '$pass\_new' WHERE user = '" . $current\_user . "';";

$result = mysqli\_query($GLOBALS["\_\_\_mysqli\_ston"], $insert ) or die( '<pre>' . ((is\_object($GLOBALS["\_\_\_mysqli\_ston"])) ? mysqli\_error($GLOBALS["\_\_\_mysqli\_ston"]) : (($\_\_\_mysqli\_res = mysqli\_connect\_error()) ? $\_\_\_mysqli\_res : false)) . '</pre>' );

// Feedback for the user

echo "<pre>Password Changed.</pre>";

}

else {

// Issue with passwords matching

echo "<pre>Passwords did not match.</pre>";

}

((is\_null($\_\_\_mysqli\_res = mysqli\_close($GLOBALS["\_\_\_mysqli\_ston"]))) ? false : $\_\_\_mysqli\_res);

}

?>

### Sicherer Code

<?php

if( isset( $\_GET[ 'Change' ] ) ) {

// Check Anti-CSRF token

checkToken( $\_REQUEST[ 'user\_token' ], $\_SESSION[ 'session\_token' ], 'index.php' );

// Get input

$pass\_curr = $\_GET[ 'password\_current' ];

$pass\_new = $\_GET[ 'password\_new' ];

$pass\_conf = $\_GET[ 'password\_conf' ];

// Sanitise current password input

$pass\_curr = stripslashes( $pass\_curr );

$pass\_curr = ((isset($GLOBALS["\_\_\_mysqli\_ston"]) && is\_object($GLOBALS["\_\_\_mysqli\_ston"])) ? mysqli\_real\_escape\_string($GLOBALS["\_\_\_mysqli\_ston"], $pass\_curr ) : ((trigger\_error("[MySQLConverterToo] Fix the mysql\_escape\_string() call! This code does not work.", E\_USER\_ERROR)) ? "" : ""));

$pass\_curr = md5( $pass\_curr );

// Check that the current password is correct

$data = $db->prepare( 'SELECT password FROM users WHERE user = (:user) AND password = (:password) LIMIT 1;' );

$current\_user = dvwaCurrentUser();

$data->bindParam( ':user', $current\_user, PDO::PARAM\_STR );

$data->bindParam( ':password', $pass\_curr, PDO::PARAM\_STR );

$data->execute();

// Do both new passwords match and does the current password match the user?

if( ( $pass\_new == $pass\_conf ) && ( $data->rowCount() == 1 ) ) {

// It does!

$pass\_new = stripslashes( $pass\_new );

$pass\_new = ((isset($GLOBALS["\_\_\_mysqli\_ston"]) && is\_object($GLOBALS["\_\_\_mysqli\_ston"])) ? mysqli\_real\_escape\_string($GLOBALS["\_\_\_mysqli\_ston"], $pass\_new ) : ((trigger\_error("[MySQLConverterToo] Fix the mysql\_escape\_string() call! This code does not work.", E\_USER\_ERROR)) ? "" : ""));

$pass\_new = md5( $pass\_new );

// Update database with new password

$data = $db->prepare( 'UPDATE users SET password = (:password) WHERE user = (:user);' );

$data->bindParam( ':password', $pass\_new, PDO::PARAM\_STR );

$current\_user = dvwaCurrentUser();

$data->bindParam( ':user', $current\_user, PDO::PARAM\_STR );

$data->execute();

// Feedback for the user

echo "<pre>Password Changed.</pre>";

}

else {

// Issue with passwords matching

echo "<pre>Passwords did not match or current password incorrect.</pre>";

}

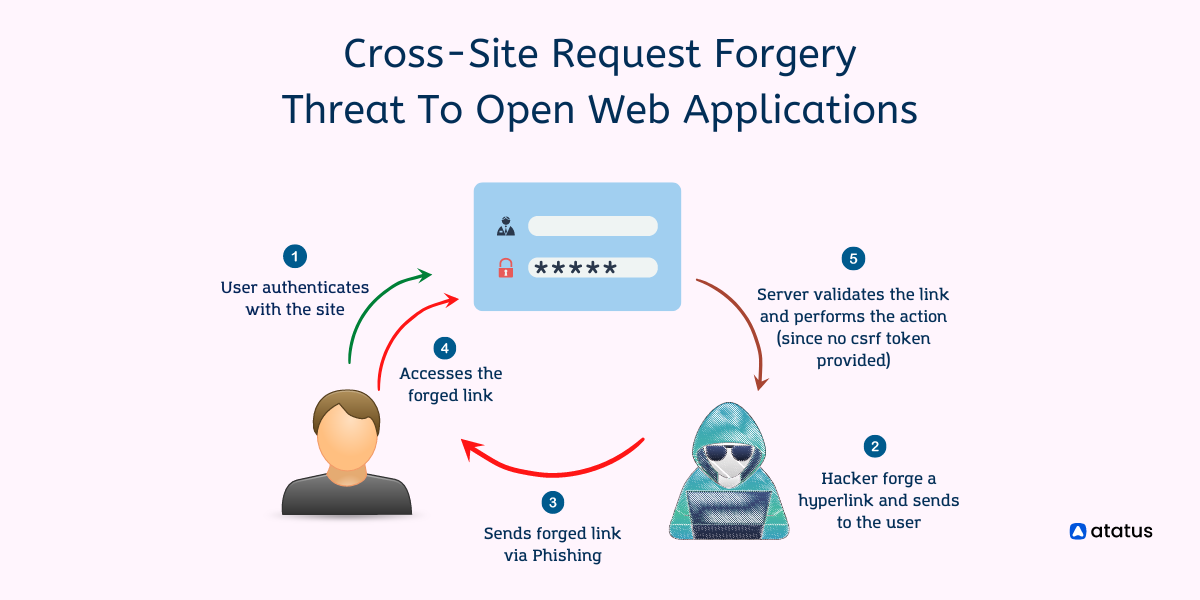
}

// Generate Anti-CSRF token

generateSessionToken();

?>

## 3.3 Grafik



# File Inclusion

## 4.1 Definition:

LFIs sind Web-Schwachstellen, die durch Fehler der Programmierer ermöglicht werden. Durch die Einführung einer Sicherheitslücke in Webanwendungen ermöglichen unvorsichtige Programmierer unbefugten Benutzern den Zugriff auf Dateien, die Nutzung von Download-Funktionen, das Durchsuchen der verfügbaren Informationen und vieles mehr.  
Code

[Literaturhinweis](https://qawerk.de/blog/was-ist-die-local-file-inclusion-lfi/) 07.03.2025

## 4.2 Code

### Unsicherer Code

<?php

// The page we wish to display

$file = $\_GET[ 'page' ];

?>

### Sicherer Code

<?php

// The page we wish to display

$file = $\_GET[ 'page' ];

// Only allow include.php or file{1..3}.php

$configFileNames = [

'include.php',

'file1.php',

'file2.php',

'file3.php',

];

if( !in\_array($file, $configFileNames) ) {

// This isn't the page we want!

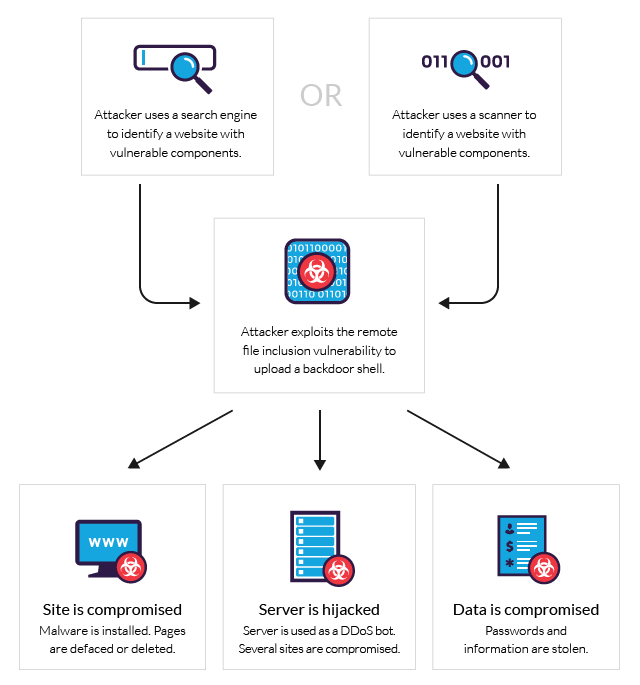
echo "ERROR: File not found!";

exit;

}

?>

## 4.3 Grafik



# File Upload

## 5.1 Definition:

Datei-Uploads sind für die Benutzerproduktivität und viele Unternehmensdienste und -anwendungen unerlässlich. Beispielsweise sind Datei-Uploads eine wichtige Funktion für Content-Management-Systeme, Gesundheitsportale, Versicherungsseiten und Messaging-Anwendungen. In dem Maße, in dem Unternehmen zu dezentralen und entfernten Arbeitsumgebungen übergehen, wird es immer wichtiger, Maßnahmen zur Gewährleistung der Sicherheit von Datei-Uploads zu implementieren, denn wenn Datei-Uploads nicht eingeschränkt werden, entsteht ein Angriffsvektor für böswillige Akteure.

[Literaturhinweis](https://german.opswat.com/blog/file-upload-protection-best-practices) 07.03.2025

## 5.2 Code

### Unsicherer Code

<?php

if( isset( $\_POST[ 'Upload' ] ) ) {

// Where are we going to be writing to?

$target\_path = DVWA\_WEB\_PAGE\_TO\_ROOT . "hackable/uploads/";

$target\_path .= basename( $\_FILES[ 'uploaded' ][ 'name' ] );

// Can we move the file to the upload folder?

if( !move\_uploaded\_file( $\_FILES[ 'uploaded' ][ 'tmp\_name' ], $target\_path ) ) {

// No

echo '<pre>Your image was not uploaded.</pre>';

}

else {

// Yes!

echo "<pre>{$target\_path} succesfully uploaded!</pre>";

}

}

?>

### Sicherer Code

<?php

if( isset( $\_POST[ 'Upload' ] ) ) {

// Check Anti-CSRF token

checkToken( $\_REQUEST[ 'user\_token' ], $\_SESSION[ 'session\_token' ], 'index.php' );

// File information

$uploaded\_name = $\_FILES[ 'uploaded' ][ 'name' ];

$uploaded\_ext = substr( $uploaded\_name, strrpos( $uploaded\_name, '.' ) + 1);

$uploaded\_size = $\_FILES[ 'uploaded' ][ 'size' ];

$uploaded\_type = $\_FILES[ 'uploaded' ][ 'type' ];

$uploaded\_tmp = $\_FILES[ 'uploaded' ][ 'tmp\_name' ];

// Where are we going to be writing to?

$target\_path = DVWA\_WEB\_PAGE\_TO\_ROOT . 'hackable/uploads/';

//$target\_file = basename( $uploaded\_name, '.' . $uploaded\_ext ) . '-';

$target\_file = md5( uniqid() . $uploaded\_name ) . '.' . $uploaded\_ext;

$temp\_file = ( ( ini\_get( 'upload\_tmp\_dir' ) == '' ) ? ( sys\_get\_temp\_dir() ) : ( ini\_get( 'upload\_tmp\_dir' ) ) );

$temp\_file .= DIRECTORY\_SEPARATOR . md5( uniqid() . $uploaded\_name ) . '.' . $uploaded\_ext;

// Is it an image?

if( ( strtolower( $uploaded\_ext ) == 'jpg' || strtolower( $uploaded\_ext ) == 'jpeg' || strtolower( $uploaded\_ext ) == 'png' ) &&

( $uploaded\_size < 100000 ) &&

( $uploaded\_type == 'image/jpeg' || $uploaded\_type == 'image/png' ) &&

getimagesize( $uploaded\_tmp ) ) {

// Strip any metadata, by re-encoding image (Note, using php-Imagick is recommended over php-GD)

if( $uploaded\_type == 'image/jpeg' ) {

$img = imagecreatefromjpeg( $uploaded\_tmp );

imagejpeg( $img, $temp\_file, 100);

}

else {

$img = imagecreatefrompng( $uploaded\_tmp );

imagepng( $img, $temp\_file, 9);

}

imagedestroy( $img );

// Can we move the file to the web root from the temp folder?

if( rename( $temp\_file, ( getcwd() . DIRECTORY\_SEPARATOR . $target\_path . $target\_file ) ) ) {

// Yes!

echo "<pre><a href='{$target\_path}{$target\_file}'>{$target\_file}</a> succesfully uploaded!</pre>";

}

else {

// No

echo '<pre>Your image was not uploaded.</pre>';

}

// Delete any temp files

if( file\_exists( $temp\_file ) )

unlink( $temp\_file );

}

else {

// Invalid file

echo '<pre>Your image was not uploaded. We can only accept JPEG or PNG images.</pre>';

}

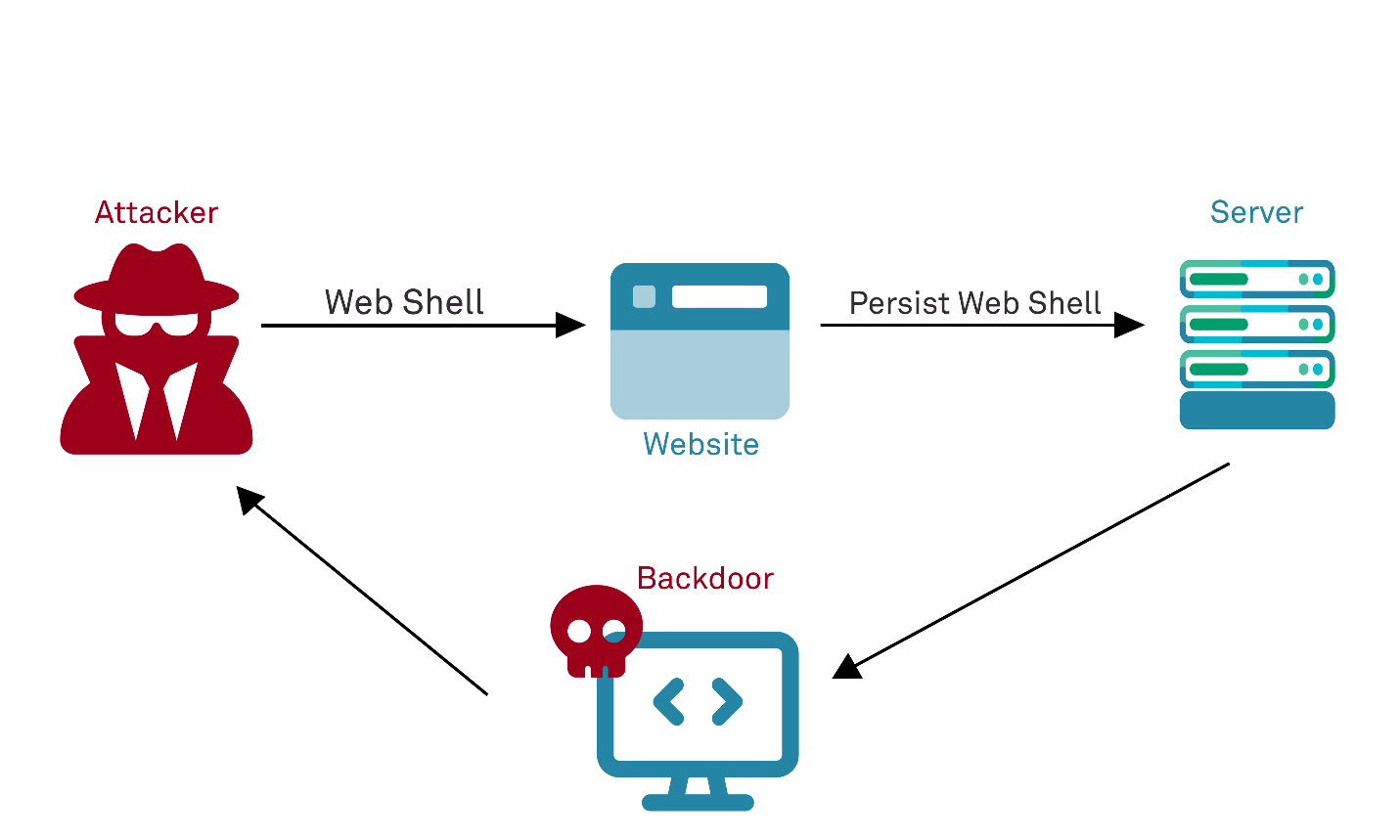
}

// Generate Anti-CSRF token

generateSessionToken();

?>

## 5.3 Grafik



# Insecure CAPTCHA

## 6.1 Definition:

Ein CAPTCHA-Test soll feststellen, ob ein Online-Nutzer wirklich ein Mensch und kein Bot ist. CAPTCHA ist eine Abkürzung für „Completely Automated Public Turing test to tell Computers and Humans Apart“ (etwa: Vollständig automatisierter öffentlicher Turing-Test, um zu erkennen, ob Computer oder Mensch). Nutzer stoßen im Internet häufig auf CAPTCHA- und reCAPTCHA-Tests. Solche Tests sind eine Möglichkeit, Bot-Aktivitäten zu steuern, obwohl der Ansatz auch Nachteile hat.

Auch wenn CAPTCHAs dafür ausgelegt sind, automatisierte Bots zu blockieren, sind CAPTCHAs selbst automatisiert. Sie sind so programmiert, dass sie an bestimmten Stellen auf einer Website erscheinen, und sie genehmigen automatisch Benutzer oder lehnen diese ab.

[Literaturhinweis](https://www.cloudflare.com/de-de/learning/bots/how-captchas-work/)

## 6.2 Code

### Unsicherer Code

<?php

if( isset( $\_POST[ 'Change' ] ) && ( $\_POST[ 'step' ] == '1' ) ) {

// Hide the CAPTCHA form

$hide\_form = true;

// Get input

$pass\_new = $\_POST[ 'password\_new' ];

$pass\_conf = $\_POST[ 'password\_conf' ];

// Check CAPTCHA from 3rd party

$resp = recaptcha\_check\_answer(

$\_DVWA[ 'recaptcha\_private\_key'],

$\_POST['g-recaptcha-response']

);

// Did the CAPTCHA fail?

if( !$resp ) {

// What happens when the CAPTCHA was entered incorrectly

$html .= "<pre><br />The CAPTCHA was incorrect. Please try again.</pre>";

$hide\_form = false;

return;

}

else {

// CAPTCHA was correct. Do both new passwords match?

if( $pass\_new == $pass\_conf ) {

// Show next stage for the user

echo "

<pre><br />You passed the CAPTCHA! Click the button to confirm your changes.<br /></pre>

<form action=\"#\" method=\"POST\">

<input type=\"hidden\" name=\"step\" value=\"2\" />

<input type=\"hidden\" name=\"password\_new\" value=\"{$pass\_new}\" />

<input type=\"hidden\" name=\"password\_conf\" value=\"{$pass\_conf}\" />

<input type=\"submit\" name=\"Change\" value=\"Change\" />

</form>";

}

else {

// Both new passwords do not match.

$html .= "<pre>Both passwords must match.</pre>";

$hide\_form = false;

}

}

}

if( isset( $\_POST[ 'Change' ] ) && ( $\_POST[ 'step' ] == '2' ) ) {

// Hide the CAPTCHA form

$hide\_form = true;

// Get input

$pass\_new = $\_POST[ 'password\_new' ];

$pass\_conf = $\_POST[ 'password\_conf' ];

// Check to see if both password match

if( $pass\_new == $pass\_conf ) {

// They do!

$pass\_new = ((isset($GLOBALS["\_\_\_mysqli\_ston"]) && is\_object($GLOBALS["\_\_\_mysqli\_ston"])) ? mysqli\_real\_escape\_string($GLOBALS["\_\_\_mysqli\_ston"], $pass\_new ) : ((trigger\_error("[MySQLConverterToo] Fix the mysql\_escape\_string() call! This code does not work.", E\_USER\_ERROR)) ? "" : ""));

$pass\_new = md5( $pass\_new );

// Update database

$insert = "UPDATE `users` SET password = '$pass\_new' WHERE user = '" . dvwaCurrentUser() . "';";

$result = mysqli\_query($GLOBALS["\_\_\_mysqli\_ston"], $insert ) or die( '<pre>' . ((is\_object($GLOBALS["\_\_\_mysqli\_ston"])) ? mysqli\_error($GLOBALS["\_\_\_mysqli\_ston"]) : (($\_\_\_mysqli\_res = mysqli\_connect\_error()) ? $\_\_\_mysqli\_res : false)) . '</pre>' );

// Feedback for the end user

echo "<pre>Password Changed.</pre>";

}

else {

// Issue with the passwords matching

echo "<pre>Passwords did not match.</pre>";

$hide\_form = false;

}

((is\_null($\_\_\_mysqli\_res = mysqli\_close($GLOBALS["\_\_\_mysqli\_ston"]))) ? false : $\_\_\_mysqli\_res);

}

?>

### Sicherer Code

<?php

if( isset( $\_POST[ 'Change' ] ) ) {

// Check Anti-CSRF token

checkToken( $\_REQUEST[ 'user\_token' ], $\_SESSION[ 'session\_token' ], 'index.php' );

// Hide the CAPTCHA form

$hide\_form = true;

// Get input

$pass\_new = $\_POST[ 'password\_new' ];

$pass\_new = stripslashes( $pass\_new );

$pass\_new = ((isset($GLOBALS["\_\_\_mysqli\_ston"]) && is\_object($GLOBALS["\_\_\_mysqli\_ston"])) ? mysqli\_real\_escape\_string($GLOBALS["\_\_\_mysqli\_ston"], $pass\_new ) : ((trigger\_error("[MySQLConverterToo] Fix the mysql\_escape\_string() call! This code does not work.", E\_USER\_ERROR)) ? "" : ""));

$pass\_new = md5( $pass\_new );

$pass\_conf = $\_POST[ 'password\_conf' ];

$pass\_conf = stripslashes( $pass\_conf );

$pass\_conf = ((isset($GLOBALS["\_\_\_mysqli\_ston"]) && is\_object($GLOBALS["\_\_\_mysqli\_ston"])) ? mysqli\_real\_escape\_string($GLOBALS["\_\_\_mysqli\_ston"], $pass\_conf ) : ((trigger\_error("[MySQLConverterToo] Fix the mysql\_escape\_string() call! This code does not work.", E\_USER\_ERROR)) ? "" : ""));

$pass\_conf = md5( $pass\_conf );

$pass\_curr = $\_POST[ 'password\_current' ];

$pass\_curr = stripslashes( $pass\_curr );

$pass\_curr = ((isset($GLOBALS["\_\_\_mysqli\_ston"]) && is\_object($GLOBALS["\_\_\_mysqli\_ston"])) ? mysqli\_real\_escape\_string($GLOBALS["\_\_\_mysqli\_ston"], $pass\_curr ) : ((trigger\_error("[MySQLConverterToo] Fix the mysql\_escape\_string() call! This code does not work.", E\_USER\_ERROR)) ? "" : ""));

$pass\_curr = md5( $pass\_curr );

// Check CAPTCHA from 3rd party

$resp = recaptcha\_check\_answer(

$\_DVWA[ 'recaptcha\_private\_key' ],

$\_POST['g-recaptcha-response']

);

// Did the CAPTCHA fail?

if( !$resp ) {

// What happens when the CAPTCHA was entered incorrectly

echo "<pre><br />The CAPTCHA was incorrect. Please try again.</pre>";

$hide\_form = false;

}

else {

// Check that the current password is correct

$data = $db->prepare( 'SELECT password FROM users WHERE user = (:user) AND password = (:password) LIMIT 1;' );

$data->bindParam( ':user', dvwaCurrentUser(), PDO::PARAM\_STR );

$data->bindParam( ':password', $pass\_curr, PDO::PARAM\_STR );

$data->execute();

// Do both new password match and was the current password correct?

if( ( $pass\_new == $pass\_conf) && ( $data->rowCount() == 1 ) ) {

// Update the database

$data = $db->prepare( 'UPDATE users SET password = (:password) WHERE user = (:user);' );

$data->bindParam( ':password', $pass\_new, PDO::PARAM\_STR );

$data->bindParam( ':user', dvwaCurrentUser(), PDO::PARAM\_STR );

$data->execute();

// Feedback for the end user - success!

echo "<pre>Password Changed.</pre>";

}

else {

// Feedback for the end user - failed!

echo "<pre>Either your current password is incorrect or the new passwords did not match.<br />Please try again.</pre>";

$hide\_form = false;

}

}

}

// Generate Anti-CSRF token

generateSessionToken();

?>

## 6.3 Grafik

A screenshot of a computer

AI-generated content may be incorrect.

# SQL-Injektion

## 7.1 Definition:

Eine SQL-Injection, manchmal abgekürzt als SQLi, ist eine Art von Sicherheitslücke, bei der ein Angreifer einen Teil des SQL-Codes (Structured Query Language) verwendet, um eine Datenbank zu manipulieren und Zugriff auf potenziell wertvolle Informationen zu erhalten. Dies ist eine der häufigsten und bedrohlichsten Angriffsarten, da sie potenziell gegen jede Webanwendung oder Webseite eingesetzt werden kann, die eine SQL-basierte Datenbank verwendet (was bei den meisten der Fall ist).

[Literaturhinweis](https://www.kaspersky.de/resource-center/definitions/sql-injection)

## 7.2 Code

### Unsicherer Code

<?php

if( isset( $\_REQUEST[ 'Submit' ] ) ) {

// Get input

$id = $\_REQUEST[ 'id' ];

switch ($\_DVWA['SQLI\_DB']) {

case MYSQL:

// Check database

$query = "SELECT first\_name, last\_name FROM users WHERE user\_id = '$id';";

$result = mysqli\_query($GLOBALS["\_\_\_mysqli\_ston"], $query ) or die( '<pre>' . ((is\_object($GLOBALS["\_\_\_mysqli\_ston"])) ? mysqli\_error($GLOBALS["\_\_\_mysqli\_ston"]) : (($\_\_\_mysqli\_res = mysqli\_connect\_error()) ? $\_\_\_mysqli\_res : false)) . '</pre>' );

// Get results

while( $row = mysqli\_fetch\_assoc( $result ) ) {

// Get values

$first = $row["first\_name"];

$last = $row["last\_name"];

// Feedback for end user

echo "<pre>ID: {$id}<br />First name: {$first}<br />Surname: {$last}</pre>";

}

mysqli\_close($GLOBALS["\_\_\_mysqli\_ston"]);

break;

case SQLITE:

global $sqlite\_db\_connection;

#$sqlite\_db\_connection = new SQLite3($\_DVWA['SQLITE\_DB']);

#$sqlite\_db\_connection->enableExceptions(true);

$query = "SELECT first\_name, last\_name FROM users WHERE user\_id = '$id';";

#print $query;

try {

$results = $sqlite\_db\_connection->query($query);

} catch (Exception $e) {

echo 'Caught exception: ' . $e->getMessage();

exit();

}

if ($results) {

while ($row = $results->fetchArray()) {

// Get values

$first = $row["first\_name"];

$last = $row["last\_name"];

// Feedback for end user

echo "<pre>ID: {$id}<br />First name: {$first}<br />Surname: {$last}</pre>";

}

} else {

echo "Error in fetch ".$sqlite\_db->lastErrorMsg();

}

break;

}

}

?>

### Sicherer Code

<?php

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

// Check Anti-CSRF token

checkToken( $\_REQUEST[ 'user\_token' ], $\_SESSION[ 'session\_token' ], 'index.php' );

// Get input

$id = $\_GET[ 'id' ];

// Was a number entered?

if(is\_numeric( $id )) {

$id = intval ($id);

switch ($\_DVWA['SQLI\_DB']) {

case MYSQL:

// Check the database

$data = $db->prepare( 'SELECT first\_name, last\_name FROM users WHERE user\_id = (:id) LIMIT 1;' );

$data->bindParam( ':id', $id, PDO::PARAM\_INT );

$data->execute();

$row = $data->fetch();

// Make sure only 1 result is returned

if( $data->rowCount() == 1 ) {

// Get values

$first = $row[ 'first\_name' ];

$last = $row[ 'last\_name' ];

// Feedback for end user

echo "<pre>ID: {$id}<br />First name: {$first}<br />Surname: {$last}</pre>";

}

break;

case SQLITE:

global $sqlite\_db\_connection;

$stmt = $sqlite\_db\_connection->prepare('SELECT first\_name, last\_name FROM users WHERE user\_id = :id LIMIT 1;' );

$stmt->bindValue(':id',$id,SQLITE3\_INTEGER);

$result = $stmt->execute();

$result->finalize();

if ($result !== false) {

// There is no way to get the number of rows returned

// This checks the number of columns (not rows) just

// as a precaution, but it won't stop someone dumping

// multiple rows and viewing them one at a time.

$num\_columns = $result->numColumns();

if ($num\_columns == 2) {

$row = $result->fetchArray();

// Get values

$first = $row[ 'first\_name' ];

$last = $row[ 'last\_name' ];

// Feedback for end user

echo "<pre>ID: {$id}<br />First name: {$first}<br />Surname: {$last}</pre>";

}

}

break;

}

}

}

// Generate Anti-CSRF token

generateSessionToken();

?>

## 7.3 Grafik

A computer screen shot of a computer

AI-generated content may be incorrect.