

DVWA-Test system

Daniel Karacic 4aWI



17. Januar 2025

Inhalt

[1 Bruteforce 2](#_Toc192236082)

[1.1 Information 2](#_Toc192236083)

[1.2 Grafische Darstellung 2](#_Toc192236084)

[1.3 Tools 2](#_Toc192236085)

[1.3.1 Gobuster 2](#_Toc192236086)

[1.3.2 BruteX 2](#_Toc192236087)

[1.3.3 Dirsearch 3](#_Toc192236088)

[1.4 Code 3](#_Toc192236089)

[2. Command Injection 4](#_Toc192236090)

[2.1 Information 4](#_Toc192236091)

[2.2 Grafische Darstellung 4](#_Toc192236092)

[2.3 Tools 4](#_Toc192236093)

[2.4 Code-Beispiel 5](#_Toc192236094)

[3. Cross-Site Request Forgery (CSRF) 6](#_Toc192236095)

[3.1 Information 6](#_Toc192236096)

[3.2 Grafische Darstellung 6](#_Toc192236097)

[3.3 Tools 6](#_Toc192236098)

[3.4 Code-Beispiel 6](#_Toc192236099)

[4. File Inclusion 7](#_Toc192236100)

[4.1 Information 7](#_Toc192236101)

[4.2 Grafische Darstellung 7](#_Toc192236102)

[4.3 Tools 7](#_Toc192236103)

[4.4 Code-Beispiel 7](#_Toc192236104)

[5. File Upload 8](#_Toc192236105)

[5.1 Information 8](#_Toc192236106)

[5.2 Grafische Darstellung 8](#_Toc192236107)

[5.3 Tools 8](#_Toc192236108)

[5.4 Code-Beispiel 8](#_Toc192236109)

# 1 Bruteforce

## Information

Ein Brute-Force-Angriff besteht darin, dass ein Angreifer systematisch vordefinierte Werte ausprobiert, um Zugang zu einem System zu erhalten. Dabei werden Anfragen an einen Server gesendet, und die Antworten werden analysiert. Dies kann entweder durch einen dictionary attack (mit oder ohne Variationen) oder durch einen klassischen Brute-Force-Angriff erfolgen, bei dem Zeichenklassen wie Buchstaben, Zahlen oder Sonderzeichen ausprobiert werden. Die Effizienz des Angriffs hängt von der Methode, der Anzahl der Versuche und der Stärke des Zielsystems ab. Mit diesen Faktoren kann der Angreifer abschätzen, wie lange der Angriff dauern wird.

(Quelle: [Brute Force Attack | OWASP Foundation](https://owasp.org/www-community/attacks/Brute_force_attack))

## What is a Brute Force Attack? Definition, Types & How It WorksGrafische Darstellung

(Quelle: [What is a Brute Force Attack? Definition, Types & How It Works](https://gridinsoft.com/brute-force))

## Tools

* Gobuster
* BruteX
* Dirsearch
* Klassisch Code: Java, Javascript

(Quelle: [11 Brute-Force-Angriffswerkzeuge für Penetrationstests](https://geekflare.com/de/best-brute-force-attack-tools/))

### Gobuster

Gobuster ist ein leistungsstarkes und vielseitiges Tool für Penetrationstester und Sicherheitsforscher, das verwendet wird, um verschiedene Arten von Verzeichnissen, Dateien, DNS-Einträgen oder virtuellen Hostnamen auf einem Server zu finden.

### 1.3.2 BruteX

BruteX ist ein automatisiertes Penetrationstesting-Tool, das für Brute-Force-Angriffe auf verschiedene Dienste wie SSH, FTP, und HTTP verwendet wird. Es kombiniert mehrere Tools und Ansätze, um systematisch Benutzernamen und Passwörter zu testen und unautorisierte Zugriffe zu identifizieren.

### 1.3.3 Dirsearch

Dirsearch ist ein leistungsfähiges und flexibles Befehlszeilen-Tool, das verwendet wird, um versteckte Verzeichnisse und Dateien auf Webservern zu finden. Es basiert auf Brute-Force-Techniken und nutzt Wortlisten, um systematisch Anfragen an einen Server zu senden und dessen Ressourcen zu entdecken.

## Code

1. <?php
2. if( isset( $\_GET[ 'Login' ] ) ) {
3. // Get username
4. $user = $\_GET[ 'username' ];
5. // Get password
6. $pass = $\_GET[ 'password' ];
7. $pass = md5( $pass );
8. // Check the database
9. $query = "SELECT \* FROM `users` WHERE user = '$user' AND password = '$pass';";
10. $result = mysqli\_query($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>' );
11. if( $result && mysqli\_num\_rows( $result ) == 1 ) {
12. // Get users details
13. $row = mysqli\_fetch\_assoc( $result );
14. $avatar = $row["avatar"];
15. // Login successful
16. echo "<p>Welcome to the password protected area {$user}</p>";
17. echo "<img src=\"{$avatar}\" />";
18. }
19. else {
20. // Login failed
21. echo "<pre><br />Username and/or password incorrect.</pre>";
22. }
23. ((is\_null($\_\_\_mysqli\_res = mysqli\_close($GLOBALS["\_\_\_mysqli\_ston"]))) ? false : $\_\_\_mysqli\_res);
24. }
25. ?>

Kein Schutz vor wiederholten Login Versuche, man kann es so oft probieren, wie man möchte.

Gefixter code:

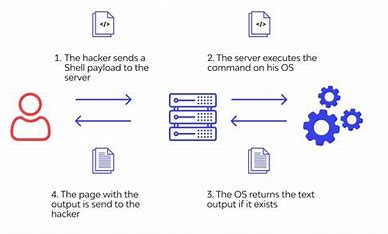
1. <?php
2. session\_start();
3. function track\_failed\_login($ip) {
4. if (!isset($\_SESSION['failed\_logins'])) {
5. $\_SESSION['failed\_logins'] = [];
6. }
7. if (!isset($\_SESSION['failed\_logins'][$ip])) {
8. $\_SESSION['failed\_logins'][$ip] = ['count' => 0, 'last\_attempt' => time()];
9. }
10. $\_SESSION['failed\_logins'][$ip]['count']++;
11. $\_SESSION['failed\_logins'][$ip]['last\_attempt'] = time();
12. }
13. function is\_ip\_blocked($ip) {
14. $max\_attempts = 5;
15. $block\_time = 300;
16. if (isset($\_SESSION['failed\_logins'][$ip])) {
17. $failed\_login = $\_SESSION['failed\_logins'][$ip];
18. if ($failed\_login['count'] >= $max\_attempts && (time() - $failed\_login['last\_attempt']) < $block\_time) {
19. return true;
20. }
21. if ((time() - $failed\_login['last\_attempt']) >= $block\_time) {
22. $\_SESSION['failed\_logins'][$ip]['count'] = 0;
23. }
24. }
25. return false;
26. }
27. if (isset($\_GET['Login'])) {
28. $ip = $\_SERVER['REMOTE\_ADDR'];
29. if (is\_ip\_blocked($ip)) {
30. die("<pre>Too many failed attempts. Try again later.</pre>");
31. }
32. $user = $\_GET['username'];
33. $pass = $\_GET['password'];
34. $pass = md5($pass);
35. $query = "SELECT \* FROM `users` WHERE user = '$user' AND password = '$pass';";
36. $result = mysqli\_query($GLOBALS["\_\_\_mysqli\_ston"], $query) or die('<pre>' . ((is\_object($GLOBALS["\_\_\_mysqli\_ston"])) ? mysqli\_error($GLOBALS["\_\_\_mysqli\_ston"]) : (($\_\_\_mysqli\_res = mysqli\_connect\_error()) ? $\_\_\_mysqli\_res : false)) . '</pre>');
37. if ($result && mysqli\_num\_rows($result) == 1) {
38. $row = mysqli\_fetch\_assoc($result);
39. $avatar = $row["avatar"];
40. echo "<p>Welcome to the password protected area {$user}</p>";
41. echo "<img src=\"{$avatar}\" />";
42. } else {
43. track\_failed\_login($ip);
44. echo "<pre><br />Username and/or password incorrect.</pre>";
45. }
46. ((is\_null($\_\_\_mysqli\_res = mysqli\_close($GLOBALS["\_\_\_mysqli\_ston"]))) ? false : $\_\_\_mysqli\_res);
47. }
48. ?>

# 2. Command Injection

## 2.1 Information

Command Injection ist eine Sicherheitslücke, bei der ein Angreifer beliebige Befehle auf einem Server ausführen kann, indem er schädliche Eingaben in ein System einschleust. Dies geschieht oft durch unsichere Verwendung von Funktionen wie exec(), system() oder shell\_exec() in Skripten. Ein erfolgreicher Angriff kann zur vollständigen Übernahme des Servers führen.

## 2.2 Grafische Darstellung



https://www.bing.com/images/search?view=detailV2&ccid=%2bz6aW1LT&id=DE00E9C11F6FA0C5B29569FA8C6342151D5B8E9C&thid=OIP.-z6aW1LTTW32wU0FDGxzbQAAAA&mediaurl=https%3a%2f%2fassets.website-files.com%2f5ff66329429d880392f6cba2%2f61fb84e37bcd6e3df5f3d410\_command+injection+how+it+work.jpg&cdnurl=https%3a%2f%2fth.bing.com%2fth%2fid%2fR.fb3e9a5b52d34d6df6c14d050c6c736d%3frik%3dnI5bHRVCY4z6aQ%26pid%3dImgRaw%26r%3d0&exph=284&expw=474&q=Command+injection&simid=608035385773071157&FORM=IRPRST&ck=69CFA2356D91CF1A9729A0B2AE3E9217&selectedIndex=8&itb=0

## 2.3 Tools

* **Burp Suite**: Zum Testen von Eingabefeldern auf Command Injection.
* **Commix**: Ein automatisiertes Tool zur Erkennung und Ausnutzung von Command Injection.
* **Netcat**: Zum Erstellen von Reverse Shells nach erfolgreicher Injection.

## 2.4 Code-Beispiel

**Unsicherer Code:**

php

Copy

<?php

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

system("ping -c 4 " . $ip);

?>

**Probleme:**

* Direkte Einbindung von Benutzereingaben in Systembefehle.
* Ein Angreifer kann Befehle wie ; rm -rf / einschleusen.

**Verbesserter Code:**

php

Copy

<?php

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

if (filter\_var($ip, FILTER\_VALIDATE\_IP)) {

system("ping -c 4 " . escapeshellarg($ip));

} else {

echo "Invalid IP address.";

}

?>

**Verbesserungen:**

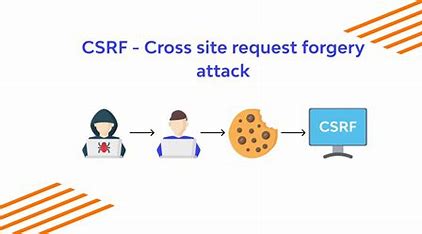
* Validierung der Eingabe mit filter\_var().
* Verwendung von escapeshellarg(), um Benutzereingaben zu sichern.

# 3. Cross-Site Request Forgery (CSRF)

## 3.1 Information

CSRF ist ein Angriff, bei dem ein Angreifer einen Benutzer dazu bringt, unbeabsichtigte Anfragen an eine Webanwendung zu senden, in der er authentifiziert ist. Dies kann zur Änderung von Kontoeinstellungen, Überweisungen oder anderen unerwünschten Aktionen führen.

## 3.2 Grafische Darstellung



https://www.bing.com/images/search?view=detailV2&ccid=y5h%2fgWWe&id=9EE4029AE804B7730EF4BBA30A981323E54D1349&thid=OIP.y5h\_gWWec4l1cb6q-w-bIQHaD4&mediaurl=https%3a%2f%2fassets.website-files.com%2f5ff66329429d880392f6cba2%2f61f251b52a583fa4149b98c0\_CSRF+Preview.jpg&cdnurl=https%3a%2f%2fth.bing.com%2fth%2fid%2fR.cb987f81659e73897571beaafb0f9b21%3frik%3dSRNN5SMTmAqjuw%26pid%3dImgRaw%26r%3d0&exph=628&expw=1200&q=csrf&simid=608002026818793981&FORM=IRPRST&ck=F6E99011160208B8726D91B3F7D6E402&selectedIndex=1&itb=0

## 3.3 Tools

* **Burp Suite**: Zum Testen auf CSRF-Schwachstellen.
* **CSRF PoC Generator**: Zum Erstellen von Proof-of-Concept-CSRF-Angriffen.

## 3.4 Code-Beispiel

**Unsicherer Code:**

php

Copy

<?php

if ($\_SERVER['REQUEST\_METHOD'] === 'POST') {

$new\_password = $\_POST['new\_password'];

// Passwort ändern

}

?>

**Probleme:**

* Keine CSRF-Token-Überprüfung.
* Ein Angreifer kann ein Formular auf einer bösartigen Seite erstellen, um das Passwort zu ändern.

**Verbesserter Code:**

php

Copy

<?php

session\_start();

if ($\_SERVER['REQUEST\_METHOD'] === 'POST') {

if (!isset($\_POST['csrf\_token']) || $\_POST['csrf\_token'] !== $\_SESSION['csrf\_token']) {

die("CSRF validation failed.");

}

$new\_password = $\_POST['new\_password'];

// Passwort ändern

}

$csrf\_token = bin2hex(random\_bytes(32));

$\_SESSION['csrf\_token'] = $csrf\_token;

?>

<form method="POST">

<input type="hidden" name="csrf\_token" value="<?php echo $csrf\_token; ?>">

<input type="password" name="new\_password">

<button type="submit">Change Password</button>

</form>

**Verbesserungen:**

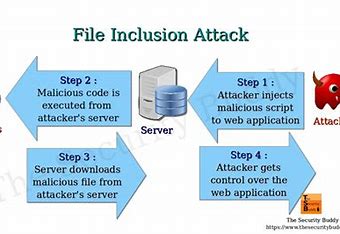
* Verwendung von CSRF-Token zur Validierung der Anfrage.

# 4. File Inclusion

## 4.1 Information

File Inclusion-Schwachstellen treten auf, wenn eine Anwendung Benutzereingaben verwendet, um Dateien einzubinden, ohne diese ausreichend zu validieren. Dies kann zu Local File Inclusion (LFI) oder Remote File Inclusion (RFI) führen, bei denen ein Angreifer schädliche Dateien ausführen kann.

## 4.2 Grafische Darstellung



https://www.bing.com/images/search?view=detailV2&ccid=Ew4KdkZT&id=F51616EA2CA085C87A315EAFC096B10D87DAE1F2&thid=OIP.Ew4KdkZTSdLVeMH8ZGqo8wHaEK&mediaurl=https%3a%2f%2fwww.thesecuritybuddy.com%2fwordpress%2fbdr%2fuploads%2f2020%2f02%2fFileInclusionAttack\_20.jpg&cdnurl=https%3a%2f%2fth.bing.com%2fth%2fid%2fR.130e0a76465349d2d578c1fc646aa8f3%3frik%3d8uHahw2xlsCvXg%26pid%3dImgRaw%26r%3d0&exph=558&expw=992&q=File+inclusion&simid=608021835190248889&FORM=IRPRST&ck=360A2EFEBC6E315D44FC57387CEE8BC3&selectedIndex=2&itb=0

## 4.3 Tools

* **Burp Suite**: Zum Testen auf File Inclusion-Schwachstellen.
* **Wfuzz**: Zum Fuzzing von Dateipfaden.

## 4.4 Code-Beispiel

**Unsicherer Code:**

php

Copy

<?php

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

include($page . '.php');

?>

**Probleme:**

* Ein Angreifer kann Pfade wie ../../etc/passwd oder externe URLs einschleusen.

**Verbesserter Code:**

php

Copy

<?php

$allowed\_pages = ['home', 'about', 'contact'];

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

if (in\_array($page, $allowed\_pages)) {

include($page . '.php');

} else {

die("Invalid page.");

}

?>

**Verbesserungen:**

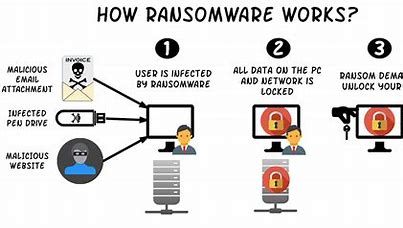
* Whitelisting erlaubter Dateien.

# 5. File Upload

## 5.1 Information

File Upload-Schwachstellen entstehen, wenn eine Anwendung das Hochladen von Dateien ohne ausreichende Validierung erlaubt. Ein Angreifer kann schädliche Dateien (z. B. PHP-Shells) hochladen und ausführen.

## 5.2 Grafische Darstellung



https://www.bing.com/images/search?view=detailV2&ccid=hmG%2bwYy6&id=A51A029D000BB8C1F612490B6AF1D8A5B106ACC9&thid=OIP.hmG-wYy6OlieSLxnZ6sB5AHaDu&mediaurl=https%3a%2f%2fmiro.medium.com%2fmax%2f1200%2f1\*eeUhOlHKynIjgrpUtGm8Xg.png&cdnurl=https%3a%2f%2fth.bing.com%2fth%2fid%2fR.8661bec18cba3a589e48bc6767ab01e4%3frik%3dyawGsaXY8WoLSQ%26pid%3dImgRaw%26r%3d0&exph=605&expw=1200&q=File+upload+cyberattack&simid=607996318793992835&FORM=IRPRST&ck=EC96745BF1267AF21F7A14059F3D180A&selectedIndex=11&itb=0

## 5.3 Tools

* **Burp Suite**: Zum Testen auf File Upload-Schwachstellen.
* **Web Shells**: Zum Hochladen und Ausführen von schädlichem Code.

## 5.4 Code-Beispiel

**Unsicherer Code:**

php

Copy

<?php

if ($\_FILES['file']) {

$target\_dir = "uploads/";

$target\_file = $target\_dir . basename($\_FILES['file']['name']);

move\_uploaded\_file($\_FILES['file']['tmp\_name'], $target\_file);

}

?>

**Probleme:**

* Keine Validierung des Dateityps.
* Ein Angreifer kann eine PHP-Shell hochladen.

**Verbesserter Code:**

php

Copy

<?php

$allowed\_types = ['image/jpeg', 'image/png'];

if ($\_FILES['file']) {

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

if (in\_array($file\_type, $allowed\_types)) {

$target\_dir = "uploads/";

$target\_file = $target\_dir . basename($\_FILES['file']['name']);

move\_uploaded\_file($\_FILES['file']['tmp\_name'], $target\_file);

} else {

die("Invalid file type.");

}

}

?>

**Verbesserungen:**

* Whitelisting erlaubter Dateitypen.
* Deaktivierung der Ausführung von Skripten im Upload-Verzeichnis.