

DVWA-Test system

Daniel Karacic 4aWI



17. Januar 2025

Inhalt

[1 Bruteforce 2](#_Toc188004495)

[1.1 Information 2](#_Toc188004496)

[1.2 Grafische Darstellung 2](#_Toc188004497)

[1.3 Tools 2](#_Toc188004498)

[1.3.1 Gobuster 2](#_Toc188004499)

[1.3.2 BruteX 2](#_Toc188004500)

[1.3.3 Dirsearch 3](#_Toc188004501)

[1.4 Code 3](#_Toc188004502)

# 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. ?>